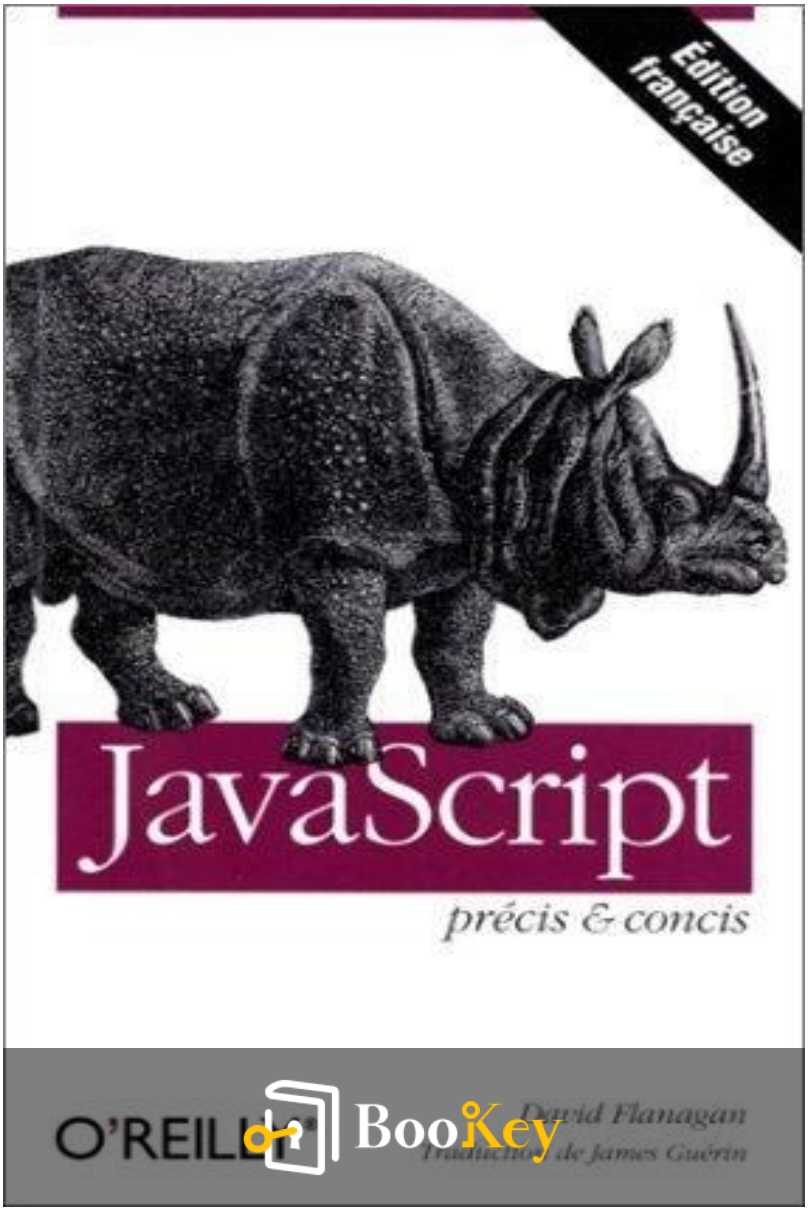
JavaScript PDF

David Flanagan



Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

JavaScript

Maîtrisez JavaScript grâce au Guide de Référence Complet Ultime.

Écrit par Bookey

En savoir plus sur le résumé de JavaScript

Écouter JavaScript Livre audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

À propos du livre

JavaScript, maintenant dans sa septième édition, est le guide de référence ultime pour les développeurs web et les programmeurs désireux de tirer le meilleur parti de JavaScript. Avec des mises à jour importantes incluant un nouveau contenu sur ECMAScript 2017 (ES8) et des chapitres supplémentaires explorant les fonctionnalités clés du langage, ce guide à succès est parfait pour les développeurs expérimentés souhaitant approfondir leur compréhension du langage de programmation du web, ainsi que pour ceux qui cherchent à améliorer leurs compétences existantes en JavaScript.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

À propos de l'auteur

David Flanagan est un auteur très respecté et un expert dans le domaine de la programmation informatique, surtout connu pour ses œuvres influentes sur JavaScript et l'écosystème plus vaste du développement web. Avec une formation en ingénierie logicielle et une passion pour l'enseignement, Flanagan a écrit de nombreux livres qui constituent des ressources essentielles tant pour les programmeurs débutants qu'avancés. Son style d'écriture clair et ses explications approfondies rendent des sujets complexes plus accessibles, lui valant une reconnaissance en tant que voix de confiance dans la communauté technologique. En tant que défenseur des logiciels open source et des standards, Flanagan continue de contribuer à l'avancement des pratiques de programmation, consolidant ainsi sa réputation en tant que figure clé dans le développement des technologies web modernes.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Ad

Scanner pour télécharger

Essayez l'appli Bookey pour lire plus de 1000 résumés des meilleurs livres du monde

Débloquez 1000+ titres, 80+ sujets

Nouveaux titres ajoutés chaque semaine

Brand | Leadership & collaboration | Gestion du temps | Relations & communication | Know

tégie d'entreprise | Créativité | Mémoires | Argent & investissements | Positive Psychology

Entrepreneuriat | Histoire du monde | Communication parent-enfant | Soins Personnels

Aperçus des meilleurs livres du monde

Essai gratuit avec Bookey

Liste du contenu du résumé

Chapitre 1 : Section 1.1. Syntaxe

Chapitre 2 : Section 1.3. Types de Données

Chapitre 3 : Section 1.4. Expressions et Opérateurs

Chapitre 4 : Section 1.5. Instructions

Chapitre 5 : Section 1.6. JavaScript orienté objet

Chapitre 6 : Section 1.7. Expressions Régulières

Chapitre 7 : Section 1.8. Versions de JavaScript

Chapitre 8 : Section 2.1. JavaScript dans HTML

Chapitre 9 : Section 2.2. L'objet Window

Chapitre 10 : Section 2.3. L'objet Document

Chapitre 11 : Section 2.4. Le DOM Héritage

Chapitre 12 : Section 2.5. Le W3C DOM

Chapitre 13 : Section 2.6. DOM IE 4

Chapitre 14 : Section 2.7. DHTML : Scriptage des styles CSS

Chapitre 15 : Section 2.8. Événements et Gestion des Événements

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Chapitre 16 : Section 2.9. Restrictions de sécurité en JavaScript

Chapitre 17 : Tableau

Chapitre 18 : Date

Chapitre 19 : Document

Chapitre 20 : Élément

Chapitre 21 : Événement

Chapitre 22 : Global

Chapitre 23 : Input

Chapitre 24 : Calque

Chapitre 25 : Lien

Chapitre 26 : Math

Chapitre 27 : Navigateur

Chapitre 28 : Node

Chapitre 29 : Nombre

Chapitre 30 : Objet

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Chapitre 31 : RegExp

Chapitre 32 : Sélection

Chapitre 33 : Chaîne

Chapitre 34 : Style

Chapitre 35 : Window

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Chapitre 1 Résumé : Section 1.1. Syntaxe

Section	Description
1. Syntaxe	La syntaxe de JavaScript est similaire à celle de Java, ce qui permet aux programmeurs familiers avec C, C++ ou Java de l'apprendre facilement.
1.1 Sensibilité à la casse	JavaScript est sensible à la casse, nécessitant une capitalisation cohérente pour les mots-clés, les variables et les noms de fonctions.
1.2 Espaces blancs	Les caractères d'espacement sont ignorés, permettant un formatage flexible et une indentation pour une meilleure lisibilité.
1.3 Points-virgules	Les instructions se terminent généralement par des points-virgules, mais ils peuvent être omis s'ils sont suivis d'un saut de ligne, avec des règles spécifiques concernant les retours à la ligne.
1.4 Commentaires	Supporte les commentaires de type C/C++ : commentaires sur une ligne avec `//` et commentaires sur plusieurs lignes avec `/* ... */`.
1.5 Identifiants	Les identifiants peuvent inclure des lettres, des chiffres, des traits de soulignement (_) et des signes dollar ($), mais ne peuvent pas commencer par un chiffre.
1.6 Mots-clés	Les mots-clés réservés incluent ceux ayant des significations spéciales, comme break, if, return, etc. Les mots-clés à usage futur sont également listés.

1. Syntaxe

La syntaxe de JavaScript ressemble beaucoup à celle de Java, qui est dérivée de C et C++. Cette familiarité facilite

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

l'apprentissage de JavaScript pour les programmeurs ayant une expérience en C, C++ ou Java.

1.1 Sensibilité à la casse

JavaScript est sensible à la casse. Tous les mots-clés sont en minuscules, et il est essentiel de maintenir une capitalisation cohérente pour les variables, les noms de fonctions et les identifiants.

1.2 Espaces blancs

Les caractères d'espacement comme les espaces, les tabulations et les sauts de ligne sont ignorés par JavaScript, ce qui permet une mise en forme flexible et une indentation du code pour améliorer sa lisibilité.

1.3 Points-virgules

Les instructions en JavaScript se terminent généralement par des points-virgules. Cependant, lorsqu'elles sont suivies d'un saut de ligne, les points-virgules peuvent être omis, avec des restrictions sur les endroits où les instructions peuvent être divisées sur plusieurs lignes.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

1.4 Commentaires

JavaScript permet les commentaires de style C et C++ :
- Les commentaires sur une seule ligne commencent par `//`.
- Les commentaires multi-lignes sont encadrés par `/*` et `*/`.

1.5 Identifiants

Les identifiants en JavaScript peuvent inclure des lettres, des chiffres, des traits de soulignement (_) et des signes dollar ($). Le premier caractère ne peut pas être un chiffre. Des exemples d'identifiants valides incluent :
- i
- nom_variable
- v13
- $str

1.6 Mots-clés

JavaScript a un ensemble de mots-clés réservés qui ont une signification spéciale et ne peuvent pas être utilisés comme identifiants :

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

- Mots-clés courants : break, do, if, switch, typeof, case, else, in, this, var, catch, false, instanceof, throw, void, continue, finally, new, true, while, default, for, null, try, with, delete, function, return.
- Mots réservés supplémentaires pour une utilisation future incluent : abstract, enum, int, short, boolean, export, interface, static, byte, extends, long, super, char, final, native, synchronized, class, float, package, throws, const, goto, private, transient, debugger, implements, protected, volatile, double, import, public.

Les développeurs doivent également éviter de nommer les variables d'après des propriétés et des fonctions globales et ne doivent pas utiliser `arguments` comme nom de variable locale.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Exemple

Point clé:Sensibilité à la casse en JavaScript

Exemple:En JavaScript, des noms de variable comme 'myVariable' et 'myvariable' sont complètement différents ; gardez cela à l'esprit pour éviter les erreurs.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Pensée critique

Point clé:La familiarité de la syntaxe de JavaScript avec celle de C, C++ et Java est une simplification excessive.

Interprétation critique:Bien qu'il soit vrai que la syntaxe de JavaScript partage des similarités avec ces langages, affirmer que cela rend JavaScript plus facile pour tous les programmeurs néglige les différences considérables en matière de paradigmes, comme l'héritage prototypal en JavaScript par rapport aux modèles basés sur les classes en Java. Cette considération est cruciale car elle peut induire en erreur les néophytes en leur faisant sous-estimer la singularité et la complexité de JavaScript, notamment dans la programmation asynchrone et les architectures orientées événements. Des auteurs comme Douglas Crockford dans 'JavaScript: The Good Parts' et Kyle Simpson dans 'You Don’t Know JS' approfondissent ces distinctions complexes, fournissant un contexte plus large qui peut contrebalancer la perspective de Flanagan.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Chapitre 2 Résumé : Section 1.3. Types de Données

Section	Contenu
1.3 Types de données	JavaScript prend en charge trois types de données primitifs (nombres, booléens, chaînes) et deux types composés (objets, tableaux), y compris des types spécialisés comme les fonctions et les dates.
1.3.1 Nombres	Format à virgule flottante 64 bits. Les littéraux numériques comprennent des valeurs décimales, hexadécimales et des valeurs spéciales (infinity, NaN). Fonctions : isNaN(), constantes numériques, méthodes Math.
1.3.2 Booléens	Deux valeurs : vrai et faux, représentant la vérité et le mensonge.
1.3.3 Chaînes	Séquences de caractères, représentées entre guillemets. Des séquences d'échappement spéciales modifient les caractères. Les chaînes ont des méthodes et une propriété de longueur, elles sont immuables.
1.3.4 Objets	Types de données composés avec des propriétés nommées accédées via l'opérateur . ou la notation entre crochets. Créés à l'aide de l'opérateur new ou de littéraux d'objet.
1.3.5 Tableaux	Objets numérotés accessibles par des indices, pouvant contenir des types mixtes. Créés à l'aide du constructeur Array ou de la syntaxe de littéral de tableau.
1.3.6 Fonctions et Méthodes	Blocs de code réutilisables. Les méthodes sont des fonctions sur des objets (mot-clé this). Les arguments sont accessibles via le tableau d'arguments.
1.3.7 Null et Undefined	Null indique l'absence de valeur ; undefined représente des variables non initialisées/propriétés manquantes. Semblable avec == mais différent avec ===.

1.3 Types de Données en JavaScript

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

JavaScript prend en charge trois types de données primitifs : les nombres, les booléens et les chaînes de caractères ; ainsi que deux types de données composés : les objets et les tableaux. Il définit également des types d'objets spécialisés tels que les fonctions, les expressions régulières et les dates.

1.3.1 Nombres

- Les nombres sont représentés en format flottant sur 64 bits sans distinction entre entiers et nombres à virgule flottante.
- Les littéraux numériques peuvent être en décimal, hexadécimal (par ex., 0xFF pour 255), et peuvent représenter des valeurs spéciales : l'infini positif/négatif et NaN (not-a-number).
- La fonction `isNaN()` teste les valeurs NaN.
- L'objet `Number` fournit des constantes, tandis que l'objet `Math` offre diverses fonctions mathématiques.

1.3.2 Booléens

- Le type booléen a deux valeurs : `true` et `false`, indiquant la vérité et le faux.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

1.3.3 Chaînes de caractères

- Les chaînes de caractères sont des séquences de caractères, représentées avec des guillemets (simples ou doubles).
- Des séquences d'échappement spéciales modifient les caractères au sein des chaînes (par ex., `\n` pour un saut de ligne, `\'` pour une apostrophe).
- Les chaînes ont des méthodes pour des opérations, la propriété `length`, et sont immuables, ce qui signifie que les modifications renvoient de nouvelles chaînes plutôt que de modifier l'original.

1.3.4 Objets

- Les objets sont des types de données composés avec des propriétés nommées.
- Accédez aux propriétés avec l'opérateur `.` (par ex., `o.x = 1`) ou la notation par crochets (par ex., `o["x"] = 1`).
- Les objets peuvent être créés en utilisant l'opérateur `new` ou des littéraux d'objet (par ex., `var o = {x:1, y:2}`).

1.3.5 Tableaux

- Les tableaux sont des objets numérotés accessibles par des

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

indices (par ex., `a[0]` pour le premier élément).
- Les tableaux peuvent contenir des types mixtes et ont une propriété de longueur.
- Utilisez le constructeur `Array` ou la syntaxe littérale des tableaux (par ex., `var a = [1,2,3]`) pour créer des tableaux.

1.3.6 Fonctions et Méthodes

- Les fonctions sont des blocs de code réutilisables définis une fois et appelés plusieurs fois (par ex., `function sum(x, y) { return x + y; }`).
- Les méthodes sont des fonctions attribuées aux propriétés des objets, avec `this` référant à l'objet contenant.
- Utilisez le tableau `arguments` pour accéder à tous les arguments passés à une fonction.

1.3.7 Null et Indéfini

- `null` indique "aucune valeur" et signifie une absence de données valides.
- `undefined` représente des variables non initialisées ou des propriétés d'objet manquantes.
- `null` et `undefined` sont traités de manière similaire avec l'opérateur `==` mais diffèrent lorsqu'on utilise `===`.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Exemple

Point clé:Comprendre les types de données primitifs et composés est essentiel pour coder efficacement en JavaScript.

Exemple:Imaginez que vous programmez une simple application web. Vous devez stocker l'âge d'un utilisateur — un nombre, par exemple, comme 28. Maintenant, vous voulez également vérifier s'il est adulte. Ici, un booléen (vrai ou faux) vous sera utile. Que faire si vous devez représenter le nom de l'utilisateur ? Vous utiliserez des chaînes de caractères. En ajoutant plusieurs utilisateurs, vous regroupez leurs données dans un tableau, ce qui facilite la gestion d'une collection. Reconnaître comment ces types de données fonctionnent et interagissent est primordial pour construire des applications JavaScript dynamiques et réactives.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Pensée critique

Point clé:Types de données primitifs vs. composés

Interprétation critique:Un point clé de ce chapitre est la distinction entre les types de données primitifs et composés en JavaScript, que l'auteur présente comme une connaissance fondamentale pour comprendre le langage. Bien que Flanagan mette en avant la simplicité des types de données primitifs de JavaScript—nombres, booléens et chaînes de caractères—comparés aux structures plus complexes des objets et des tableaux, il est essentiel de critiquer cette simplification. Par exemple, la gestion des nombres par un système à virgule flottante peut conduire à des erreurs de précision inattendues, comme le souligne la norme IEEE sur les nombres à virgule flottante ([IEEE 754](https://en.wikipedia.org/wiki/IEEE_754)). Cela soulève la question de savoir si les débutants pourraient surestimer la clarté et la fiabilité de ces types sans une compréhension plus profonde de leurs implications. Les critiques pourraient soutenir que la présentation de l'auteur ne prépare pas adéquatement les apprenants aux défis qu'ils pourraient rencontrer lorsque ces types de données interagissent dans des scénarios pratiques.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 3 Résumé : Section 1.4. Expressions et Opérateurs

Catégorie	Description	P	A	Opérateur	Opération
Expressions et Opérateurs	Les expressions JavaScript combinent des valeurs à l'aide d'opérateurs, avec des parenthèses pour le regroupement.				
Priorité des Opérateurs	Détermine l'ordre des opérations.				
Associativité des Opérateurs	Décrit comment les opérateurs de même priorité sont regroupés (de gauche à droite ou de droite à gauche).				
Vue d'ensemble des Opérateurs	P A Opérateur Opération 15L.Accéder à une propriété d'objet 15L[ ]Accéder à un élément de tableau 15L( )Appeler une fonction 14RnewCréer un nouvel objet 14R++Incrémenter (unaire) 14R--Décrémenter (unaire) 14R-Moins unaire 14R+Plus unaire 14R~Complément à un bit 14R!Complément logique 14RdeleteDéfinir une propriété comme indéfinie 14RtypeofRetourner le type de données 14RvoidRetourner indéfini 13L*, /, %Multiplication, division, reste 12L+, -Ajouter, soustraire ; Concaténer des chaînes 11L<<, >>, >>>Décalages à bits 10L<, <=, >, >=Opérateurs relationnels 9L==, !=, ===, !==Vérifications d'égalité et d'identité 8L&ET à bits 7L^XOR à bits 6L|OU à bits 5L&&ET logique 4L||OU logique 3R?:Opérateur conditionnel 2R=Affectation 2R*=, +=, -=, etc.Affectation avec	15	L	.	Accéder à une propriété d'objet

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Catégorie	Description	P	A	Opérateur	Opération
	opération 1L,Évaluation multiple				
15	L	.	Accéder à une propriété d'objet		
15	L	[ ]	Accéder à un élément de tableau		
15	L	( )	Appeler une fonction		
14	R	new	Créer un nouvel objet		
14	R	++	Incrémenter (unaire)		
14	R	--	Décrémenter (unaire)		
14	R	-	Moins unaire		
14	R	+	Plus unaire		
14	R	~	Complément à un bit		
14	R	!	Complément logique		
14	R	delete	Définir une propriété comme indéfinie		
14	R	typeof	Retourner le type de données		
14	R	void	Retourner indéfini		
13	L	*, /, %	Multiplication, division, reste		
12	L	+, -	Ajouter, soustraire ; Concaténer des chaînes		
11	L	<<, >>, >>>	Décalages à bits		
10	L	<, <=, >, >=	Opérateurs relationnels		
9	L	==, !=, ===, !==	Vérifications d'égalité et d'identité		

Catégorie	Description	P	A	Opérateur	Opération
8	L	&	ET à bits		
7	L	^	XOR à bits		
6	L	|	OU à bits		
5	L	&&	ET logique		
4	L	||	OU logique		
3	R	?:	Opérateur conditionnel		
2	R	=	Affectation		
2	R	*=, +=, -=, etc.	Affectation avec opération		
1	L	,	Évaluation multiple		

Expressions et Opérateurs en JavaScript

Les expressions JavaScript combinent diverses valeurs à l'aide d'opérateurs, avec des parenthèses utilisées pour le regroupement et l'altération de l'ordre d'évaluation.

Priorité des Opérateurs et Associativité

JavaScript dispose d'une large gamme d'opérateurs similaires à ceux de C, C++ et Java. Les opérateurs sont classés par priorité et associativité :

-

Priorité (P)

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Détermine l'ordre des opérations.

-

Associativité (A)

: Décrit comment les opérateurs de même priorité sont regroupés (soit de gauche à droite, soit de droite à gauche).

Vue d'ensemble des Opérateurs

P	A	Opérateur	Opération effectuée
15	L	.	Accéder à une propriété d'objet
15	L	[ ]	Accéder à un élément de tableau
15	L	( )	Appeler une fonction
14	R	new	Créer un nouvel objet
14	R	++	Incrémentation pré- ou post-fixe (unaire)
14	R	--	Décrémentation pré- ou post-fixe (unaire)

Installer l'application Bookey pour débloquer le texte complet et l'audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

Scanner pour télécharger

Pourquoi Bookey est une application incontournable pour les amateurs de livres

Contenu de 30min
Plus notre interprétation est profonde et claire, mieux vous saisissez chaque titre.

Format texte et audio
Absorbez des connaissances même dans un temps fragmenté.

Quiz
Vérifiez si vous avez maîtrisé ce que vous venez d'apprendre.

Et plus
Plusieurs voix & polices, Carte mentale, Citations, Clips d'idées...

Essai gratuit avec Bookey

Chapitre 4 Résumé : Section 1.5. Instructions

Section	Description	Syntaxe
Programme JavaScript	Composé d'une séquence d'instructions semblable à C, C++ et Java	N/A
Instructions d'expression	Chaque expression peut se suffire à elle-même comme instruction	Exemples : s = "hello world";, x = Math.sqrt(4);, x++;
Instructions composées	Un regroupement d'instructions dans des accolades	N/A
Instructions vides	Une instruction qui ne fait rien	;
Instructions étiquetées	Instructions étiquetées pour utilisation avec break et continue	label: statement;
break	Sort de la boucle la plus proche	break; ou break label;
case	Étiquette une condition dans des instructions switch	N/A
continue	Passe à l'itération suivante d'une boucle	continue; ou continue label;
default	Joue le rôle de solution de repli dans les instructions switch	N/A
do/while	Exécute un bloc tant qu'une condition est vraie	do statement while (expression);
for	Une boucle combinant initialisation, condition et mise à jour	for (initialize; test; update) statement;
for/in	Parcourt les propriétés d'un objet	for (variable in object) statement;
function	Déclare une fonction	function funcname(args) { statements }
if/else	Exécution conditionnelle des instructions	if (expression) statement;else statement2;
return	Sort d'une fonction et retourne éventuellement une valeur	return; ou return expression;

switch	Ramification multipath basée sur une expression	switch(expression) {
		case constant-expression: statements
		...
		default: statements
		}

throw	Signale une erreur, transférant le contrôle au gestionnaire d'exceptions	throw expression;
try/catch/finally	Mécanisme de gestion des exceptions	try { statements } catch (argument) { statements }

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Section	Description	Syntaxe
		finally { statements }
var	Déclare et initialise des variables	var name [= value] [, name2 [= value2] ...];
while	Une boucle qui continue tant qu'une condition est vraie	while (expression) statement;
with	Ajoute un objet à la chaîne de portée; son utilisation est déconseillée	with (object) statement;

Aperçu des instructions JavaScript

Un programme JavaScript se compose d'une séquence d'instructions, dont la syntaxe ressemble largement à celle de C, C++ et Java.

Instructions d'expression

- Chaque expression en JavaScript peut être une instruction autonome.
- Les exemples incluent des affectations (par ex., `s = "hello world"`), des appels de méthode (par ex., `x = Math.sqrt(4)`), et des opérations d'incrémentation/décrémentation (par ex., `x++`).

Instructions composées

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

- Un regroupement d'instructions entre accolades forme une instruction composée.
- Utile pour plusieurs instructions dans des constructions comme des boucles ou des conditionnelles.

Instructions vides

- Une instruction vide est uniquement représentée par un point-virgule (`;`) et ne fait rien.
- Elle est utile pour des boucles nécessitant un corps vide.

Instructions étiquetées

- Les instructions peuvent être étiquetées pour être utilisées avec break et continue.
- Syntaxe : `label: statement`.

Instructions JavaScript courantes

-
break
: Sort de la boucle la plus proche.
- Syntaxe : `break;` ou `break label;`

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
case
: Utilisé dans des instructions switch ; étiquette une condition.

-
continue
: Passe à l'itération suivante d'une boucle.
- Syntaxe : `continue;` ou `continue label;`

-
default
: Sert de solution de secours dans les instructions switch.
-
do/while
: Exécute un bloc tant qu'une condition est vraie.
- Syntaxe : `do statement while (expression);`

-
for
: Une boucle combinant initialisation, condition et mise à jour en une seule ligne.
- Syntaxe : `for (initialize; test; update) statement;`
-

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

for/in
: Boucle sur les propriétés d'un objet.
- Syntaxe : `for (variable in object) statement;`

-
function
: Déclare une fonction.
- Syntaxe : `function funcname(args) { statements }`

-
if/else
: Exécution conditionnelle des instructions.
- Syntaxe : `if (expression) statement;` et `else statement2;`

-
return
: Sort d'une fonction et retourne éventuellement une valeur.
- Syntaxe : `return;` ou `return expression;`

-
switch
: Ramification multiple basée sur une expression.
- Syntaxe :
  ```
  switch(expression) {
    case constant-expression: statements
    ...
    default: statements
```

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

```
}
```

- **throw** : Signale une erreur, transférant le contrôle au gestionnaire d'exception le plus proche.
  - Syntaxe : `throw expression;`
- **try/catch/finally** : Mécanisme de gestion des exceptions.
  - Syntaxe :

```
try { statements }
catch (argument) { statements }
finally { statements }
```

- **var** : Déclare et initialise des variables.
  - Syntaxe : `var name [= value] [, name2 [= value2] ...];`
- **while** : Une boucle qui se poursuit tant qu'une condition est vraie.
  - Syntaxe : `while (expression) statement;`

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-

with

: Ajoute un objet à la chaîne de portée ; son utilisation est déconseillée en raison de la complexité.

- Syntaxe : `with (object) statement;`

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Exemple

Point clé:Comprendre les instructions JavaScript est crucial pour ordonner votre logique efficacement.

Exemple:Lorsque vous écrivez un programme JavaScript, pensez à vos instructions comme à des ingrédients dans une recette ; chacune a un rôle spécifique à jouer. Par exemple, lorsque vous réalisez que `let result = Math.pow(2, 3);` n'est pas simplement une expression mais une instruction autonome qui apporte de la clarté à votre code, vous comprenez comment chaque élément de la syntaxe s'assemble pour créer une logique fonctionnelle. Si vous oubliez de placer un point-virgule, votre dessert pourrait se retrouver gâté ! Chaque instruction peut affecter le contrôle de flux, comme l'utilisation d'une boucle `for` pour itérer sur des données vous permet de manipuler des collections sans effort, renforçant ainsi l'idée que chaque instruction est essentielle pour une exécution correcte.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Pensée critique

Point clé:La similarité de la syntaxe avec C, C++ et Java

Interprétation critique:Bien que Flanagan mette en avant la ressemblance de la syntaxe de JavaScript avec celles de C, C++ et Java, il est essentiel d'aborder cette perspective de manière critique. L'auteur suggère que ces similarités facilitent la courbe d'apprentissage pour les programmeurs familiers avec ces langages. Cependant, ce point de vue peut simplifier à l'excès les différences fondamentales en matière de paradigmes et d'exécution : JavaScript est principalement axé sur les événements et asynchrone, ce qui diverge significativement de la nature synchrone de C/C++. Comprendre ces distinctions est crucial pour maîtriser JavaScript, car les langages de programmation ne peuvent pas être compris uniquement à travers la syntaxe. Ceux qui s'intéressent à des explorations plus profondes des paradigmes linguistiques devraient consulter des sources telles que "Programming JavaScript Applications" d'Eric Elliott, qui souligne les aspects uniques de JavaScript au-delà de la simple syntaxe.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Chapitre 5 Résumé : Section 1.6. JavaScript orienté objet

Sujet	Résumé
JavaScript orienté objet	JavaScript prend en charge la programmation orientée objet à travers les objets, l'héritage et les classes.
Objets JavaScript	Les objets sont des tableaux associatifs reliant des valeurs à des propriétés nommées et prennent en charge l'héritage.
Définir une classe	Une classe est définie à l'aide d'une fonction constructeur, initialisée avec l'opérateur `new` et le mot-clé `this`.
Propriété Prototype	La propriété `prototype` d'un constructeur permet de partager des méthodes et des propriétés entre les instances de classe.
Exemples de méthodes	Des méthodes peuvent être définies sur le prototype pour effectuer des opérations comme le calcul de distances et la conversion en chaînes de caractères.
Méthodes et propriétés statiques	Les méthodes/propriétés statiques sont directement attachées au constructeur, par exemple en définissant un point constant.
Créer et utiliser des objets	Pour créer un objet, utilisez le constructeur et appelez ses méthodes pour effectuer des opérations.

JavaScript orienté objet

Objets JavaScript

Les objets JavaScript sont des tableaux associatifs qui lient des valeurs à des propriétés nommées. Ils supportent un mécanisme d'héritage simple et permettent la création de nouvelles classes d'objets.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Définir une classe

Pour définir une nouvelle classe, une fonction constructeur est créée. Le constructeur est invoqué avec l'opérateur `new` et utilise le mot-clé `this` pour initialiser l'objet nouvellement créé. Par exemple, un constructeur `Point` initialise les coordonnées x et y :

```javascript
function Point(x, y) {
    this.x = x;
    this.y = y;
}
```

Propriété Prototype

Chaque fonction constructeur possède une propriété `prototype` pointant vers un objet prototype spécial pour la classe associée. Les propriétés définies sur ce prototype sont héritées par tous les objets créés avec le constructeur. Cela est généralement utilisé pour définir des méthodes pour les instances de classe.

Exemples de méthodes

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Définir des méthodes sur le prototype permet aux instances d'effectuer des opérations, telles que calculer la distance entre deux points et les convertir en chaînes de caractères :

```javascript
Point.prototype.distanceTo = function(that) {
    var dx = this.x - that.x;
    var dy = this.y - that.y;
    return Math.sqrt(dx * dx + dy * dy);
}
Point.prototype.toString = function() {
    return '(' + this.x + ',' + this.y + ')';
}
```

Méthodes et propriétés statiques

Les méthodes ou propriétés statiques sont assignées directement à la fonction constructeur. Par exemple, une constante fréquemment utilisée peut être définie comme suit :

```javascript
Point.ORIGIN = new Point(0, 0);
```

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Créer et utiliser des objets

Pour utiliser la classe `Point`, un nouvel objet peut être créé et des méthodes appelées comme suit :

```javascript
var p = new Point(3, 4);
var d = p.distanceTo(Point.ORIGIN);
var msg = "Distance vers " + p + " est " + d;
```

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Exemple

Point clé:Comprendre la propriété prototype améliore votre programmation orientée objet en JavaScript.

Exemple:En travaillant avec JavaScript, imaginez créer une application robuste où vous définissez une classe 'Personne' pour gérer les utilisateurs. En définissant des méthodes comme 'calculerÂge' sur le prototype, vous permettez à chaque objet personne de calculer facilement son âge sans dupliquer le code, illustrant ainsi le pouvoir des prototypes pour fournir une fonctionnalité partagée à toutes les instances.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Pensée critique

Point clé:L'explication de l'auteur sur les principes de la programmation orientée objet en JavaScript met l'accent sur l'encapsulation et l'héritage par chaîne de prototypes.

Interprétation critique:Bien que Flanagan présente ces concepts comme fondamentaux dans la conception de JavaScript, il est essentiel de noter que tous les développeurs ne s'accordent pas sur la supériorité ou la simplicité de cette approche. Certains soutiennent que l'héritage basé sur les prototypes de JavaScript peut mener à des complications, en particulier pour ceux qui passent de langages basés sur les classes. Cette perspective est explorée dans des ressources telles que 'You Don’t Know JS' de Kyle Simpson qui souligne les pièges potentiels de la dépendance exclusive à l'héritage par prototype, suggérant qu'une compréhension plus nuancée de la flexibilité et des particularités de JavaScript est nécessaire pour une programmation efficace.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Chapitre 6 Résumé : Section 1.7. Expressions Régulières

Section	Description
1.7 Expressions Régulières	JavaScript prend en charge les expressions régulières pour la correspondance de motifs, similaire à la syntaxe Perl.
1.7.1 Caractères Littéraux	La plupart des caractères se représentent eux-mêmes, avec des significations spéciales pour certaines ponctuations et séquences d'échappement.
Séquences d'Échappement	Représentent des caractères littéraux (par exemple, `\n`, `\r`, `\t`) et de la ponctuation (par exemple, `\\`, `\/`).
Caractères Hexadécimaux et Unicode	Représentés sous la forme `\xnn` et `\uxxxx`.
1.7.2 Classes de Caractères	Définies à l'aide de crochets, permettant des correspondances dans des ensembles de caractères spécifiques.
Échappements de Classes de Caractères Courants	`[...]`, `[^...]`, `.`, `\w`, `\W`, `\s`, `\S`, `\d`, `\D` pour différentes correspondances de caractères.
1.7.3 Répétition	Contrôle combien de fois un motif peut correspondre à une chaîne.
Opérateurs de Répétition	`?`, `+`, `*`, `{n}`, `{n,}`, `{n,m}` et correspondance non gourmande avec `?`.
1.7.4 Regroupement et Alternance	Utilise des parenthèses pour regrouper des sous-expressions et séparer des alternatives.
Correspondances de Groupes	Se réfère aux sous-expressions avec `$n` pour le texte correspondant.
1.7.5 Ancrage de la Position de Correspondance	Définit des positions pour les correspondances dans une chaîne.
Synthaxe d'Ancrage	`^`, `$`, `\b`, `\B`, `(?=p)`, `(?!p)` pour correspondre à diverses positions de chaîne et affirmations.

1.7 Expressions Régulières

JavaScript utilise des expressions régulières pour la correspondance de motifs, employant une syntaxe similaire à Perl. JavaScript 1.2 prend en charge les expressions Perl 4,

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

tandis que JavaScript 1.5 les améliore avec certaines fonctionnalités de Perl 5. Les expressions régulières peuvent être exprimées littéralement à l'aide de délimiteurs slash (/) ou sous forme d'objets RegExp créés avec le constructeur RegExp().

1.7.1 Caractères Littéraux

La plupart des caractères dans une expression régulière se représentent eux-mêmes. Certains signes de ponctuation et séquences d'échappement ont des significations spéciales, y compris :

- **Séquences d'échappement** : Représentent des caractères littéraux (par ex. `\n`, `\r`, `\t`) et de la ponctuation (par ex. `\\`, `\/`).
- **Caractères Hexadécimaux et Unicode**

Installer l'application Bookey pour débloquer le texte complet et l'audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

App Store
Coup de cœur

22k avis 5 étoiles

Scanner pour télécharger

Retour Positif

Fabienne Moreau

ue résumé de livre ne testent
ion, mais rendent également
musant et engageant.
té la lecture pour moi.

Fantastique!

Giselle Dubois

Je suis émerveillé par la variété de livres et de langues que Bookey supporte. Ce n'est pas juste une application, c'est une porte d'accès au savoir mondial. De plus, gagner des points pour la charité est un grand plus !

Fi

Le
liv
co
pr

é Blanchet

de lecture
eption de
es,
cous.

J'adore !

Isoline Mercier

Bookey m'offre le temps de parcourir les parties importantes d'un livre. Cela me donne aussi une idée suffisante pour savoir si je devrais acheter ou non la version complète du livre ! C'est facile à utiliser !"

Gain de temps !

Bookey est mon applicat
intellectuelle. Les résum
magnifiquement organis
monde de connaissance

ppli géniale !

Joachim Lefevre

adore les livres audio mais je n'ai pas toujours le temps
'écouter le livre entier ! Bookey me permet d'obtenir
n résumé des points forts du livre qui m'intéresse !!!
Quel super concept !!! Hautement recommandé !

Appli magnifique

Cette application est une bouée de sauve
amateurs de livres avec des emplois du te
Les résumés sont précis, et les cartes mer
renforcer ce que j'ai appris. Hautement re

Essai gratuit avec Bookey

Chapitre 7 Résumé : Section 1.8. Versions de JavaScript

1.8 Versions de JavaScript

Netscape a développé plusieurs versions de JavaScript, avec Microsoft introduisant des versions compatibles appelées JScript. Le comité de normalisation ECMA a publié trois éditions de la norme ECMAScript. Voici un aperçu des différentes versions et de leur relation :

-

JavaScript 1.0 :
La version originale pleine de bogues, implémentée dans Netscape 2 et désormais obsolète.

-

JavaScript 1.1 :
A introduit de véritables objets Array et résolu de nombreux bogues ; implémentée par Netscape 3.

-

JavaScript 1.2 :
A ajouté l'instruction switch, les expressions régulières et d'autres fonctionnalités ; presque conforme à ECMA v1.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-

JavaScript 1.3 :

A corrigé les incompatibilités et s'est conformée à ECMA v1 ; implémentée dans Netscape 4.5.

-

JavaScript 1.4 :

Utilisée uniquement dans les produits serveur de Netscape.

-

JavaScript 1.5 :

A introduit la gestion des exceptions et s'est conformée à ECMA v3 ; implémentée par Mozilla et Netscape 6.

Versions JScript :

-

JScript 1.0 :

Équivalent à JavaScript 1.0 pour les premières versions d'IE 3.

-

JScript 2.0 :

Équivalent à JavaScript 1.1 pour les versions ultérieures d'IE 3.

-

JScript 3.0 :

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Comparable à JavaScript 1.3 et conforme à ECMA v1 ; implémenté par IE 4.

-

JScript 5.0 :

A supporté la gestion des exceptions ; partiellement conforme à ECMA v3 ; utilisé dans IE 5.

-

JScript 5.5 :

Équivalent à JavaScript 1.5 et entièrement conforme à ECMA v3 ; vu dans IE 5.5 et IE 6.

Versions ECMA :

-

ECMA v1 :

Normalise les fonctionnalités de JavaScript 1.1, mais exclut l'instruction switch et le support des expressions régulières ; les implémentations conformes sont JavaScript 1.3 et JScript 3.0.

-

ECMA v2 :

Une version de maintenance avec des clarifications et aucune nouvelle fonctionnalité.

-

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

ECMA v3 :
Normalise les instructions switch, les expressions régulières et la gestion des exceptions ; conformité atteinte par JavaScript 1.5 et JScript 5.5.

Chapitre 1 : Le langage JavaScript

JavaScript est un langage de script léger optimisé pour être intégré dans des pages HTML, couvrant les aspects fondamentaux et côté client de JavaScript.

2.1 JavaScript dans HTML

JavaScript peut être inclus dans HTML via des scripts, des gestionnaires d'événements et des URL :
-
<script> Tag :
La plupart du code JavaScript est inséré entre les balises <script>. Avec JavaScript 1.1 et plus, des scripts externes peuvent être inclus à l'aide de l'attribut `src`.
-
Gestionnaires d'événements :
JS peut apparaître dans les attributs de gestionnaires d'événements des balises HTML, activés lors d'événements

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

spécifiques (par exemple, `onclick`).

-

URLs JavaScript :

Appliqué à l'aide du pseudo-protocole `javascript:` pour exécuter du code JS directement.

2.2 L'objet Window

L'objet Window représente une fenêtre de navigateur, agissant comme un espace global pour les propriétés et méthodes. Les caractéristiques clés incluent :

-

Dialogues simples :

Méthodes comme `alert()`, `confirm()`, et `prompt()` pour les interactions utilisateur.

-

Gestion de la ligne d'état :

Définition du texte dans la zone d'état via la propriété `status`.

-

Minuteurs :

Utilisation de `setTimeout()` et `setInterval()` pour planifier l'exécution de fonctions.

-

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Informations système :
Accéder aux propriétés `navigator` et `screen` pour les détails du navigateur et du système.
-
Naviguer dans le navigateur :
Utilisation des propriétés `location` et `history` pour manipuler le flux de navigation et naviguer.
-
Contrôle de la fenêtre :
Méthodes pour déplacer, redimensionner et ouvrir/fermer des fenêtres de navigateur.
-
Multiples fenêtres et cadres :
Considérer chaque cadre comme un objet Window séparé et y accéder en conséquence.

2.3 L'objet Document

L'objet Document est essentiel, représentant le document HTML chargé.

DOM hérité :

Fournit l'accès au contenu et aux propriétés du document

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

mais est limité en capacité.

2.5 DOM W3C :

Améliore le DOM hérité, permettant l'accès à n'importe quel élément de document avec des méthodes comme `getElementById()` et des méthodes pour manipuler les attributs et la structure des nœuds.

2.6 DOM IE 4 :

Introduit par Internet Explorer 4, il prend en charge des fonctionnalités similaires à W3C mais manque de certaines standardisations.

2.7 DHTML : Scripting des styles CSS

Le HTML dynamique combine HTML, CSS et JavaScript pour des pages web visuellement dynamiques. La propriété `style` permet de manipuler directement les attributs CSS.

2.8 Événements et gestion des événements

JavaScript peut gérer les événements en utilisant des attributs

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

standard préfixés par "on". Les événements peuvent être associés aux balises HTML, les problèmes de compatibilité entre différents navigateurs affectant la gestion avancée des événements.

2.9 Restrictions de sécurité de JavaScript

JavaScript impose plusieurs restrictions pour des raisons de sécurité, notamment :
- Limitations de la politique de cross-origin.
- Prevention de la gestion de fichiers sur les machines locales.
- Confirmation de l'utilisateur requise pour des actions sensibles.

Ce chapitre fournit une base sur l'évolution, la structure et la fonctionnalité de JavaScript, mettant en lumière à la fois sa puissance et les limitations imposées pour des raisons de sécurité.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Pensée critique

Point clé:L'évolution de JavaScript à travers ses différentes versions illustre les complexités et les incohérences des normes en développement web.

Interprétation critique:Alors que l'évolution de JavaScript et ses différentes versions sont exposées, les lecteurs pourraient s'interroger sur la fiabilité de cette transformation rapide, qui pourrait suggérer une instabilité dans une technologie web fondamentale. Le fait que plusieurs itérations et normes de compatibilité variées (comme JScript par rapport à ECMAScript) ont vu le jour pour résoudre des bugs antérieurs indique non seulement un progrès, mais aussi une potentielle confusion pour les développeurs qui s'adaptent aux changements. Cela soulève la question de la confiance exclusive dans l'interprétation d'un seul auteur, telle que présentée dans le résumé, qui pourrait négliger des perspectives et critiques plus larges concernant les problèmes de normalisation dans les langages de programmation. Pour des insights plus approfondis, des références comme 'JavaScript the Good Parts' de Douglas Crockford et la série 'You Don’t Know JS' de Kyle Simpson offrent des points de vue contrastés sur

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

l'évolution et les défis de JavaScript.

Chapitre 8 Résumé : Section 2.1. JavaScript dans HTML

Section	Contenu
JavaScript dans le HTML	Intégration via des scripts, des gestionnaires d'événements et des URLs JavaScript.
La balise <script>	Intégrée entre les balises <script> et </script>. Fichiers externes référencés à l'aide de l'attribut src : <script src="library.js">. Les fichiers ont normalement l'extension .js ; </script> est obligatoire même avec src. L'attribut language est obsolète ; utilisez l'attribut type défini sur text/javascript.
Gestionnaires d'Événements	Exécutés via les attributs des balises HTML commençant par 'on'. Exemple : <input type="button" value="Appuyez sur moi" onclick="alert('Bonjour le monde !');"> déclenche une alerte au clic.
URLs JavaScript	Exécutés en utilisant le pseudo-protocole javascript:. Utilisez l'opérateur void pour exécuter JavaScript sans changer le document : <form action="javascript:void validate()">.

JavaScript dans HTML

JavaScript peut être intégré dans des fichiers HTML par le biais de scripts, de gestionnaires d'événements et d'URL JavaScript.

La balise <script>

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

- JavaScript est généralement inclus entre les balises `<script>` et `</script>`.
- Vous pouvez référencer des fichiers JavaScript externes en utilisant l'attribut `src`: `<script src="library.js"></script>`.
- Les fichiers ont généralement une extension `.js`, et la balise `</script>` reste nécessaire même avec l'attribut `src`.
- L'attribut `language` spécifie le langage de script mais est remplacé par l'attribut `type`, qui doit être défini sur `text/javascript`.

Gestionnaires d'événements

- JavaScript peut également être exécuté via des attributs de gestionnaire d'événements dans les balises HTML, qui commencent par 'on'.
- Par exemple: `<input type="button" value="Appuie sur moi" onclick="alert('Bonjour le monde !');">` déclenche une alerte JavaScript lors du clic sur le bouton.

URLs JavaScript

- Le code JavaScript peut également être exécuté via des

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

URL utilisant le pseudo-protocole `javascript:`.
- Pour exécuter JavaScript sans modifier le document actuel, utilisez l'opérateur `void`: `<form action="javascript:void validate()">`.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Pensée critique

Point clé:Intégration de JavaScript dans HTML

Interprétation critique:Bien que David Flanagan souligne l'intégration simple de JavaScript dans HTML, on pourrait faire valoir que cette dépendance à l'incorporation de scripts directement au sein de HTML peut entraîner des problèmes potentiels en matière de maintenabilité et de séparation des préoccupations. En associant étroitement les scripts avec le balisage, les développeurs peuvent créer involontairement des complexités lors du débogage et des mises à jour, ce qui va à l'encontre des pratiques modernes soutenues par des frameworks comme React ou Angular, qui promeuvent une séparation claire entre la structure (HTML) et le comportement (JavaScript). Il est important de considérer que les réflexions de Flanagan sont précieuses, mais elles doivent être envisagées au regard des meilleures pratiques qui évoluent dans le développement web, comme le soutient des sources telles que 'Clean Code' de Robert C. Martin. Cela met en lumière l'importance d'évaluer les points de vue de l'auteur dans le contexte des méthodologies émergentes.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Chapitre 9 Résumé : Section 2.2. L'objet Window

Section	Détails
L'objet Window	L'objet Window représente une fenêtre de navigateur web et sert d'objet global dans JavaScript côté client.
Propriétés et Méthodes Globales	Définit les propriétés/méthodes globales de haut niveau, accessibles sans préfixes. Propriété notable : document.
Boîtes de Dialogue Simples	Méthodes : alert() pour les messages. confirm() pour les questions oui/non. prompt() pour la saisie de texte par l'utilisateur.
La Ligne d'État	Affiche l'URL et d'autres informations. Définir le texte d'état avec status et le défaut avec defaultStatus.
Temporisateurs	Exécute du code après des intervalles : setTimeout() pour des exécutions uniques. setInterval() pour des exécutions répétées. Identifiants d'annulation retournés par clearTimeout() et clearInterval().
Informations Système	Les propriétés navigator et screen fournissent des détails système, aidant dans le code spécifique au navigateur.
Navigation dans le Navigateur	Gère l'URL avec location (lecture pour l'URL actuelle, définition pour naviguer). La propriété history permet de naviguer dans l'historique.
Contrôle de la Fenêtre	Les méthodes pour la manipulation de fenêtres incluent open() pour de nouvelles fenêtres et close() pour fermer les fenêtres créées par le script.
Multiple Fenêtres et Cadres	Chaque fenêtre/cadre a un objet Window séparé. Les interactions nécessitent des références explicites. Les cadres HTML sont des objets Window indépendants.

L'objet Window

L'objet Window représente une fenêtre de navigateur web et sert d'objet global dans JavaScript côté client.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Propriétés et Méthodes Globales

- L'objet Window définit des propriétés et des méthodes globales de haut niveau, accessibles sans préfixes.
- Une propriété importante est `document`, qui fait référence à l'objet Document affiché dans le navigateur.

Boîtes de Dialogue Simples

- Trois méthodes pour afficher des boîtes de dialogue :
  - `alert()` pour les messages.
  - `confirm()` pour les questions oui/non.
  - `prompt()` pour la saisie de texte par l'utilisateur.

La Ligne de Statut

- La ligne de statut affiche les URL et d'autres informations.

Installer l'application Bookey pour débloquer le texte complet et l'audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

Scanner pour télécharger

Lire, Partager, Autonomiser

Terminez votre défi de lecture, faites don de livres aux enfants africains.

Le Concept

BOOKS FOR AFRICA × ×

Cette activité de don de livres se déroule en partenariat avec Books For Africa. Nous lançons ce projet car nous partageons la même conviction que BFA : Pour de nombreux enfants en Afrique, le don de livres est véritablement un don d'espoir.

La Règle

Gagnez 100 points → **Échangez un livre** → **Faites un don à l'Afrique**

Votre apprentissage ne vous apporte pas seulement des connaissances mais vous permet également de gagner des points pour des causes caritatives ! Pour chaque 100 points gagnés, un livre sera donné à l'Afrique.

Essai gratuit avec Bookey

Chapitre 10 Résumé : Section 2.3. L'objet Document

L'objet Document

Chaque objet Window possède une propriété document qui renvoie à un objet Document, essentiel car il représente le document HTML affiché dans la fenêtre du navigateur.

Le Modèle d'Objet Document (DOM)

Le DOM décrit comment le contenu d'un document est accessible et modifiable, et il existe plusieurs types de DOM :

DOM Héritage

Le DOM héritage original a évolué avec les premières versions de JavaScript, permettant d'accéder aux parties clés du document telles que les formulaires et les images. Il est bien supporté sur tous les navigateurs.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

DOM W3C

Ce modèle offre un accès complet à tout le contenu du document et est normalisé par le World Wide Web Consortium (W3C). Supporté par les navigateurs modernes, il diffère du DOM IE 4 mais standardise de nombreuses caractéristiques héritées. Le livre se concentre sur les fonctionnalités de base pertinentes pour les programmeurs JavaScript.

DOM IE 4

Développé par Microsoft, ce DOM a étendu le DOM héritage, offrant des fonctionnalités puissantes pour la manipulation du contenu du document. Ces fonctionnalités n'étaient pas normalisées mais ont trouvé un certain soutien dans des navigateurs non-Microsoft.

Conclusion

Les sections suivantes détailleront chaque type de DOM et leurs fonctionnalités pour accéder et modifier le contenu du document.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Exemple

Point clé:Comprendre l'objet Document est essentiel pour programmer efficacement en JavaScript.

Exemple:Imaginez que vous construisez une page web qui met à jour dynamiquement le contenu en fonction des interactions de l'utilisateur. L'objet Document, représentant votre document HTML, vous permet d'accéder et de manipuler des éléments comme des paragraphes et des boutons. Par exemple, si vous voulez changer le texte d'un paragraphe lorsqu'un utilisateur clique sur un bouton, vous utiliseriez JavaScript pour interagir avec l'objet Document. En comprenant comment naviguer dans le DOM, vous pouvez créer des applications web réactives et interactives qui réagissent aux entrées de l'utilisateur, enrichissant ainsi considérablement l'expérience utilisateur.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Pensée critique

Point clé:L'importance du Document Object Model (DOM) dans le développement web.

Interprétation critique:L'auteur souligne le rôle crucial du DOM en tant qu'interface entre les documents web et JavaScript, qui permet la manipulation dynamique du contenu. Cependant, bien que le DOM W3C soit largement accepté comme standard, les développeurs doivent rester vigilants concernant la compatibilité entre navigateurs et les modèles alternatifs, tels que le DOM IE 4. L'hypothèse selon laquelle tous les navigateurs prendront uniformément en charge le modèle W3C peut conduire à des négligences dans les applications pratiques, comme en témoignent de nombreux développeurs qui rencontrent des comportements incohérents sur différentes plateformes. Des ressources comme Can I use (caniuse.com) peuvent fournir des aperçus sur la compatibilité des navigateurs et les problèmes potentiels.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Chapitre 11 Résumé : Section 2.4. Le DOM Héritage

Le DOM Héritage

Le DOM Héritage fait référence au modèle d'objet document JavaScript côté client original (DOM), qui permet d'accéder à diverses propriétés et contenus du document.

Propriétés de l'objet Document

- L'objet Document fournit plusieurs propriétés en lecture seule, notamment :
  - **titre :** Titre du document
  - **URL :** URL du document
  - **dernierModifié :** Date de la dernière modification du document

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

- Il inclut également des tableaux qui référencent des types de contenu spécifiques du document :

- **forms[] :**
  Tableau d'objets Form
- **images[] :**
  Tableau d'objets Image
- **applets[] :**
  Tableau d'objets applet Java incorporés
- **links[] :**
  Tableau d'objets Link
- **anchors[] :**
  Tableau d'objets Anchor

Références Nommées

Des références nommées peuvent être créées en utilisant l'attribut name HTML, ce qui permet aux scripts d'accéder aux éléments du document par leur nom, en plus de la notation de tableau.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Objets Form

L'objet Form contient :
-
elements[] :
Un tableau d'éléments d'entrée au sein du formulaire qui peuvent être accessibles dans l'ordre ou par leur nom.

Limitations du DOM Héritage

Le DOM Héritage est limité à des types de contenu spécifiques (forms, images, etc.) et ne dispose pas de méthodes pour interagir directement avec d'autres balises ou obtenir le texte du document.

Documents Générés Dynamiquement

- La méthode `write()` de l'objet Document permet la génération de contenu dynamique pendant la phase de chargement du document.
- Attention : L'utilisation de `document.write()` après le chargement du document écrasera le document actuel.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Formulaires Dynamiques

- Les éléments d'entrée peuvent être mis à jour dynamiquement en utilisant leurs propriétés, telles que `value`.
- Exemple de définition de l'heure locale actuelle dans un champ de formulaire dynamiquement.

Validation de Formulaire

- La balise `<form>` prend en charge un gestionnaire d'événements `onsubmit` pour valider les soumissions de formulaire en vérifiant les champs de saisie.

Effets d'Image

- Les effets d'image peuvent être réalisés en utilisant le tableau d'images et en changeant la propriété `src` lors d'événements de souris (par exemple, `onmouseover`, `onmouseout`).

Gestion des Cookies

- La propriété cookie permet de définir et d'interroger les

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

cookies associés au document.
- Les cookies doivent suivre un format spécifique pour les paires nom/valeur, avec des dates d'expiration et des spécifications de chemin optionnelles.
- Une fonction exemple est fournie pour récupérer la valeur d'un cookie spécifique.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 12 Résumé : Section 2.5. Le W3C DOM

Le W3C DOM

Le W3C DOM standardise les fonctionnalités DOM héritées et en introduit de nouvelles, permettant aux scripts d'accéder et de manipuler divers éléments du document au-delà des formulaires et des images.

Recherche d'Éléments par ID

- Utilisez un attribut `id` unique pour des éléments spéciaux.
- Accédez aux éléments avec la méthode `getElementById()`.

Exemple :
```javascript
var t = document.getElementById("title");
```

Recherche d'Éléments par Nom de Tag

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

- Accédez aux éléments via leur nom de tag en utilisant `getElementsByTagName()`, qui renvoie un tableau d'éléments correspondants.

Exemple :
```javascript
var lists = document.getElementsByTagName("ul");
var item = lists[1].getElementsByTagName("li")[2];
```

Parcours d'un Arbre de Document

- Le W3C DOM représente un document comme un arbre de nœuds (tags HTML, texte, commentaires).
- Utilisez des propriétés comme `parentNode`, `firstChild`, `nextSibling` et `lastChild` pour naviguer dans l'arbre.

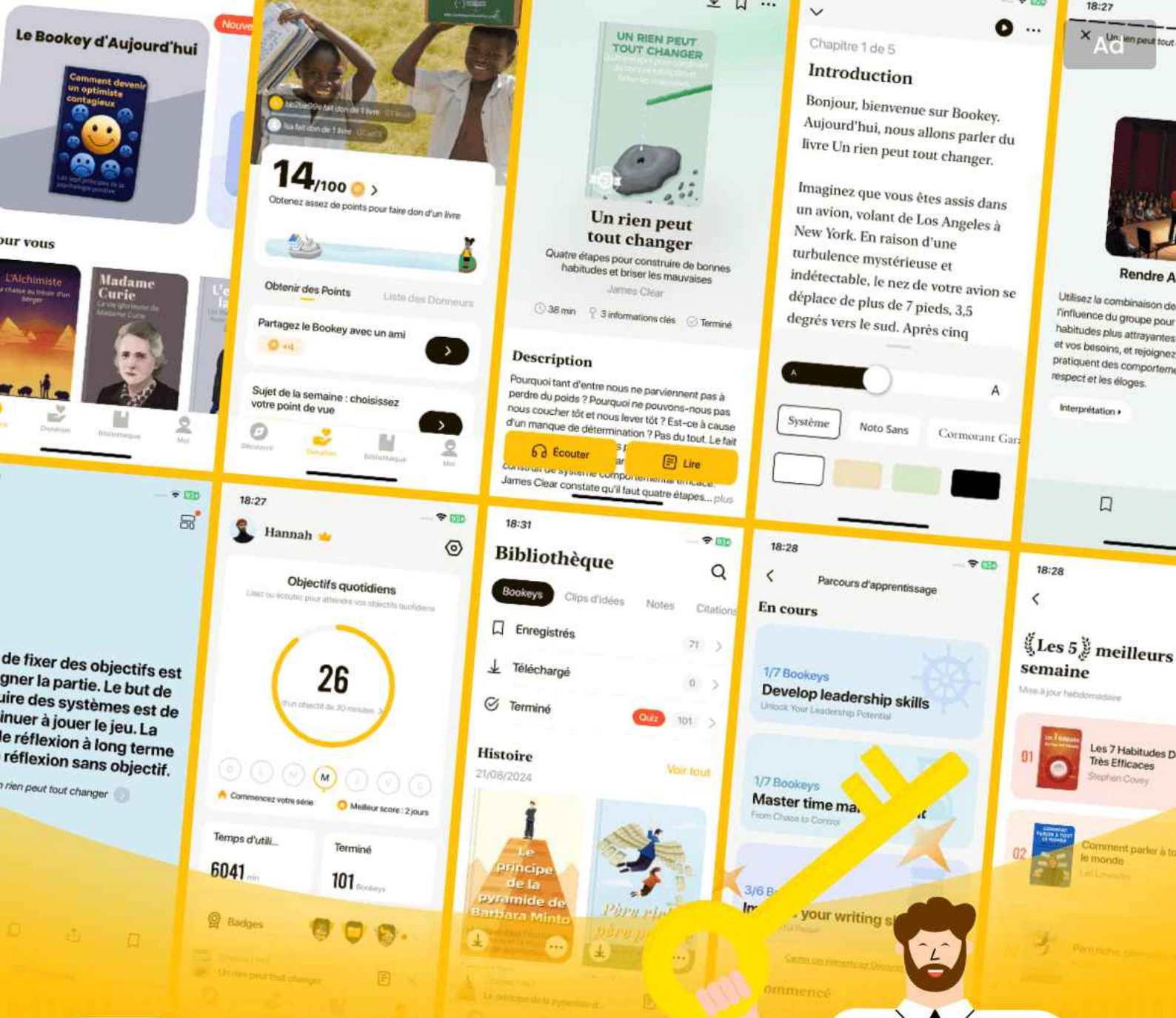
Exemple :
```javascript

**Installer l'application Bookey pour débloquer le texte complet et l'audio**

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé
```



Les meilleures idées du monde débloquent votre potentiel

Essai gratuit avec Bookey

Télécharger dans App Store

DISPONIBLE SUR Google Play

Scanner pour télécharger

Chapitre 13 Résumé : Section 2.6. DOM IE 4

Vue d'ensemble du DOM IE 4

Le DOM IE 4, introduit avec Internet Explorer 4, est un modèle d'objet documentaire puissant mais non standard. Bien que les versions IE 5 et ultérieures prennent en charge la plupart des fonctionnalités du DOM W3C, ce modèle plus ancien reste pertinent en raison de son utilisation.

Accéder aux éléments du document

Contrairement au DOM W3C, le DOM IE 4 n'utilise pas la méthode getElementById(). Au lieu de cela, les éléments peuvent être accessibles via le tableau all[] de l'objet document, soit en utilisant un identifiant sous forme de chaîne, soit en y faisant référence directement. La méthode tags() sur le tableau all[] est utilisée pour récupérer les éléments par nom de balise, nécessitant des noms de balises en majuscules.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Parcourir l'arbre du document

Le parcours dans le DOM IE 4 ressemble à celui du DOM W3C mais utilise des noms de propriété différents : `children[]` remplace `childNodes[]`, et `parentElement` remplace `parentNode`. Notamment, les commentaires et le texte du document sont exclus de l'arbre, le texte étant accessible via les propriétés innerHTML et innerText.

Modifier le contenu et la structure du document

Les objets d'éléments dans le DOM IE 4 peuvent voir leurs propriétés définies et interrogées de manière similaire aux éléments du DOM W3C. La propriété innerText modifie le contenu textuel d'un élément, tandis que la propriété innerHTML permet de remplacer le contenu d'un élément par une chaîne HTML, bien que de manière moins efficace. La propriété outerHTML et les méthodes insertAdjacentHTML() sont d'autres moyens, bien que moins courants, de manipuler les éléments.

Compatibilité du DOM

Pour garantir la compatibilité entre les différentes

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

implémentations du DOM, les scripts peuvent vérifier l'existence de méthodes ou de propriétés. Cela permet à la fonctionnalité conçue pour le DOM W3C de revenir au DOM IE 4 si nécessaire, offrant une solution de script robuste pour divers navigateurs.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 14 Résumé : Section 2.7. DHTML : Scriptage des styles CSS

DHTML : Scriptage des styles CSS

Aperçu du DHTML

Le DHTML (HTML dynamique) intègre HTML, CSS et JavaScript, permettant la modification dynamique des styles, y compris la position et la visibilité des éléments du document.

Propriété de style

- Chaque élément du document possède une propriété de style liée à ses styles en ligne.
- La propriété de style fait référence à un objet Style dont les propriétés correspondent aux attributs CSS.
- Par exemple, pour accéder à l'attribut de couleur : `e.style.color` et pour définir la couleur de fond : `e.style.backgroundColor`.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

- Les attributs CSS avec des tirets utilisent des noms en camel case en JavaScript (par exemple, `background-color` devient `backgroundColor`).

Cas particulier

- L'attribut `float` est un mot réservé en JavaScript ; il est appelé `cssFloat`.

Propriétés CSS et scriptage dynamique

- La norme CSS comprend des propriétés pour améliorer l'apparence visuelle, en particulier le positionnement et la visibilité :
  - **Position** : Valeurs - absolue, relative, fixe, statique.
  - **Gauche/Haut** : Coordonnées X/Y.
  - **Largeur/Hauteur** : Dimensions de l'élément.
  -

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Z-Index
: Ordre de superposition ; les valeurs plus élevées sont au-dessus.
-
Affichage
: Valeurs - bloc, en ligne, aucun.
-
Visibilité
: Peut être définie comme cachée ou visible ; de l'espace est alloué pour les éléments cachés.
-
Débordement
: Comment gérer le contenu dépassant la taille de l'élément (visible, caché, défilement, automatique).
-
Clip
: Portion du contenu affichée (syntaxe : `rect(haut droite bas gauche)`).

Exemple d'animation DHTML

- Démontre une animation simple de mouvement d'un élément.
- Code pour animer un élément en le déplaçant de 10 pixels

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

vers la droite toutes les 50 millisecondes jusqu'à ce qu'il ait bougé 20 fois, moment auquel il devient invisible.

```html
<h1 id='title'>Animation DHTML</h1>
<script>
var e = document.getElementById("title");
e.style.position = "absolute";
var frame = 0;
function nextFrame() {
  if (frame++ < 20) {
    e.style.left = (10 * frame) + "px";
    setTimeout("nextFrame()", 50);
  } else e.style.visibility = "hidden";
}
nextFrame();
</script>
```

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 15 Résumé : Section 2.8. Événements et Gestion des Événements

2.8 Événements et Gestion des Événements

Cette section aborde la manière d'implémenter JavaScript côté client dans les documents HTML à travers des attributs de gestion d'événements.

Attributs Standards de Gestion des Événements

Un tableau résume les gestionnaires d'événements standards commençant par "on" et leurs balises HTML correspondantes. Les gestionnaires clés incluent :

- **onabort** : Déclenché lorsqu'un chargement d'image est abandonné.
- **onclick** : S'exécute lors de l'appui et du relâchement de la souris.
- **onfocus**

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Activé lorsqu'un élément acquiert le focus clavier.

-

onload

: Se déclenche une fois qu'un document ou une image est entièrement chargé.

-

onsubmit

: Lié à la soumission de formulaire, permettant de prévenir la soumission par une instruction return false.

Les actions par défaut de certains événements peuvent être empêchées en retournant false, sauf dans le cas de onmouseover, où retourner true empêche l'URL d'être affichée dans la barre d'état.

2.8.1 Gestionnaires d'Événements en Tant que Fonctions JavaScript

Les gestionnaires d'événements en HTML correspondent à

Installer l'application Bookey pour débloquer le texte complet et l'audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

Scanner pour télécharger

Essayez l'appli Bookey pour lire plus de 1000 résumés des meilleurs livres du monde

Débloquez 1000+ titres, 80+ sujets

Nouveaux titres ajoutés chaque semaine

Brand | Leadership & collaboration | Gestion du temps | Relations & communication | Know

tégie d'entreprise | Créativité | Mémoires | Argent & investissements | Positive Psychology

Entrepreneuriat | Histoire du monde | Communication parent-enfant | Soins Personnels

Aperçus des meilleurs livres du monde

Essai gratuit avec Bookey

Chapitre 16 Résumé : Section 2.9. Restrictions de sécurité en JavaScript

Restrictions de sécurité en JavaScript

Présentation

Les implémentations de JavaScript côté client imposent diverses restrictions de sécurité pour empêcher des actions dangereuses, protéger la vie privée des utilisateurs et améliorer l'expérience globale. Ces restrictions ne sont pas standardisées, mais beaucoup sont communes à différents navigateurs.

Restrictions clés

- **Politique de même origine** : Les scripts ne peuvent accéder qu'aux propriétés des fenêtres et des documents du même serveur web, limitant le scripting entre fenêtres.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
Restrictions sur le téléchargement de fichiers
: Les scripts ne peuvent pas manipuler la propriété value de l'élément de formulaire FileUpload.
-
Soumission d'e-mails et de nouvelles
: Les scripts sont empêchés de soumettre des formulaires à des URLs mailto: ou news: sans confirmation de l'utilisateur.
-
Gestion des fenêtres
: Les scripts ne peuvent fermer que les fenêtres qu'ils ont créées, sauf si une confirmation de l'utilisateur est obtenue.
-
Accès au cache
: Les scripts ne peuvent pas charger d'URLs de type about: comme about:cache.
-
Création de fenêtres
: Les scripts ne peuvent pas créer de fenêtres hors écran ou petites, ou de fenêtres sans barres de titre.

Restrictions évolutives

En raison de l'utilisation abusive de JavaScript pour des

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

actions indésirables, les navigateurs plus récents permettent aux utilisateurs de configurer les paramètres de sécurité pour bloquer les publicités pop-up et manipuler la taille ou la position des fenêtres.

JavaScript côté client

Définition et fonctionnalité

JavaScript côté client fait référence au code intégré dans les fichiers HTML, exécuté par les navigateurs web. Il donne accès aux objets de navigateur et de document, et est souvent basé sur des événements, réagissant aux interactions des utilisateurs.

Cadre et implications de sécurité

Les capacités puissantes de JavaScript côté client peuvent introduire des vulnérabilités de sécurité significatives, poussant les navigateurs à appliquer des restrictions sur les actions des scripts. Cette section se concentre sur l'intégration de JavaScript dans HTML, introduit les objets JavaScript, la gestion des événements et décrit les mesures de sécurité.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Pensée critique

Point clé:L'absence de standardisation des restrictions de sécurité en JavaScript soulève des inquiétudes concernant la confidentialité et la sécurité des utilisateurs.

Interprétation critique:Bien que l'auteur souligne l'importance de diverses restrictions de sécurité en JavaScript côté client, il est essentiel d'évaluer de manière critique leur efficacité et leur cohérence à travers les navigateurs. L'absence d'un standard universel signifie que les mesures de sécurité peuvent varier considérablement, laissant potentiellement les utilisateurs vulnérables selon le navigateur qu'ils utilisent. Cette incohérence mine la fiabilité de JavaScript en tant que technologie et suggère que le point de vue de l'auteur pourrait surestimer la nature protectrice de ces mesures. En fait, des recherches menées par des experts en sécurité comme Michal Zalewski ont tiré la sonnette d'alarme sur la sécurité des navigateurs, soulignant la nécessité d'une approche plus normalisée et robuste en matière de sécurité JavaScript. Les lecteurs sont encouragés à consulter des sources externes pour mieux comprendre les complexités et les

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

lacunes des pratiques de sécurité en JavaScript.

Chapitre 17 Résumé : Tableau

Création et manipulation de tableaux en JavaScript

Méthodes de constructeur

-
new Array()
: Crée un tableau vide.
-
new Array(n)
: Crée un tableau avec `n` éléments non définis.
-
new Array(e0, e1,...)
: Initialise un tableau avec des éléments spécifiés.

Syntaxe littérale

- Les tableaux peuvent être créés en utilisant des crochets avec des valeurs séparées par des virgules, par exemple, `var a = [1, true, 'abc'];`.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Propriétés des tableaux

-
length
: Un entier en lecture/écriture qui indique le nombre d'éléments dans le tableau. Il peut être modifié pour tronquer ou étendre le tableau.

Méthodes des tableaux

-
concat(value, ...)
: Combine des tableaux et des valeurs dans un nouveau tableau.
-
join(separator)
: Convertit chaque élément en chaîne et les concatène avec un séparateur spécifié.
-
pop()
: Supprime et retourne le dernier élément, modifiant la longueur du tableau.
-
push(value, ...)

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Ajoute des valeurs à la fin et retourne la nouvelle longueur.

-

reverse()

: Inverse l'ordre du tableau.

-

shift()

: Supprime et retourne le premier élément, ajustant la longueur.

-

slice(start, end)

: Crée un nouveau tableau avec des éléments allant de l'indice de départ jusqu'à, mais sans inclure, l'indice de fin.

-

sort(orderfunc)

: Trie les éléments sur place selon une fonction de tri optionnelle.

-

splice(start, deleteCount, value,...)

: Supprime les éléments spécifiés et insère de nouveaux éléments à l'indice défini, retournant les éléments supprimés.

-

toLocaleString()

: Retourne une représentation de chaîne localisée du tableau.

-

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

toString()
: Convertit le tableau en chaîne.
-
unshift(value, ...)
: Ajoute de nouveaux éléments au début, déplaçant les éléments existants et retournant la nouvelle longueur.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 18 Résumé : Date

Objet Date en JavaScript

L'objet Date en JavaScript est utilisé pour travailler avec les dates et les heures, et il peut être créé en utilisant différents formats.

Variantes du constructeur

- `new Date();` - Crée un objet Date pour la date et l'heure actuelles.
- `new Date(milliseconds);` - Initialise à partir d'un horodatage (millisecondes).
- `new Date(datestring);` - Analyse une chaîne de date fournie.
- `new Date(year, month, day, hours, minutes, seconds, ms);` - Définit les champs de date et d'heure individuels, avec les deux premiers arguments (année et mois) étant obligatoires.

Méthodes de l'objet Date

L'objet Date fonctionne sans propriétés et s'appuie sur

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

différentes méthodes :

-

Méthodes Get :

- `get[UTC]Date()`: Renvoie le jour du mois (1-31).
- `get[UTC]Day()`: Renvoie le jour de la semaine (0-6).
- `get[UTC]FullYear()`: Renvoie l'année complète.
- `get[UTC]Hours()`: Renvoie les heures (0-23).
- `get[UTC]Milliseconds()`: Renvoie les millisecondes.
- `get[UTC]Minutes()`: Renvoie les minutes (0-59).
- `get[UTC]Month()`: Renvoie le mois (0-11).
- `get[UTC]Seconds()`: Renvoie les secondes (0-59).
- `getTime()`: Renvoie les millisecondes depuis le 1er janvier 1970.
- `getTimezoneOffset()`: Renvoie la différence en minutes par rapport à l'UTC.
- `getYear()`: Renvoie l'année moins 1900 (déprécié).

-

Installer l'application Bookey pour débloquer le texte complet et l'audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

Scanner pour télécharger

Pourquoi Bookey est une application incontournable pour les amateurs de livres

Contenu de 30min
Plus notre interprétation est profonde et claire, mieux vous saisissez chaque titre.

Format texte et audio
Absorbez des connaissances même dans un temps fragmenté.

Quiz
Vérifiez si vous avez maîtrisé ce que vous venez d'apprendre.

Et plus
Plusieurs voix & polices, Carte mentale, Citations, Clips d'idées...

Essai gratuit avec Bookey

Chapitre 19 Résumé : Document

Objet Document dans JavaScript côté client

Présentation

L'objet Document représente un document HTML et est un élément clé dans JavaScript côté client, introduit dans JavaScript 1.0, avec des améliorations supplémentaires dans JavaScript 1.1. À la fois Netscape et Internet Explorer ont ajouté des fonctionnalités non standard à l'objet Document, tandis que la norme DOM du W3C fournit des spécifications supplémentaires.

Propriétés communes

- **alinkColor**
: Couleur des liens activés (obsolète).
- **anchors[]**
: Tableau d'objets Anchor dans le document.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
applets[]
: Tableau d'objets Applet dans le document.
-
bgColor
: Couleur de fond (obsolète).
-
cookie
: Gère les cookies associés au document.
-
domain
: Spécifie le domaine Internet du document pour la sécurité.
-
embeds[]
: Objets représentant des données intégrées via la balise `<embed>`.
-
fgColor
: Couleur du texte par défaut (obsolète).
-
forms[]
: Tableau d'objets Form dans le document.
-
images[]

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Tableau d'objets Image dans le document.

-

lastModified

: Date de la dernière modification du document.

-

linkColor

: Couleur des liens non visités (obsolète).

-

links[]

: Tableau d'objets Link dans le document.

-

location

: URL du document (obsolète).

-

plugins[]

: Synonyme pour embeds[].

-

referrer

: URL du document qui a lié au document actuel.

-

title

: Contenu textuel de la balise `<title>`.

-

URL

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: URL du document.

-

vlinkColor

: Couleur des liens visités (obsolète).

Propriétés W3C DOM

-

body

: Référence à la balise `<body>` du document.

-

defaultView

: La fenêtre dans laquelle le document apparaît.

-

documentElement

: Référence à la balise `<html>` du document.

-

implementation

: L'implémentation qui a créé le document.

Propriétés Internet Explorer 4

-

activeElement

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Fait référence à l'élément d'entrée actuellement actif.

-

all[]

: Tableau de tous les objets Element dans le document.

-

charset

: Jeu de caractères du document.

-

children[]

: Enfants directs du document sous forme de tableau.

-

defaultCharset

: Jeu de caractères par défaut du document.

-

expando

: Empêche l'expansion des objets côté client.

-

parentWindow

: La fenêtre contenant le document.

-

readyState

: États de chargement du document (non initialisé, en chargement, interactif, complet).

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Propriétés Netscape 4

-
height
: Hauteur du document en pixels.
-
layers[]
: Tableau d'objets Layer dans le document (obsolète).
-
width
: Largeur du document en pixels.

Méthodes communes

-
clear()
: Efface le contenu du document (obsolète).
-
close()
: Ferme un flux de document.
-
open()
: Ouvre un flux pour un nouveau contenu de document.
-

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

write(value...)
: Insère des chaînes dans le document.
-
writeln(value...)
: Insère des chaînes avec un retour à la ligne dans le document.

Méthodes W3C DOM

-
createAttribute(name)
: Crée un nœud d'attribut.
-
createComment(text)
: Retourne un nouveau nœud Comment.
-
createDocumentFragment()
: Retourne un nœud DocumentFragment vide.
-
createElement(tagName)
: Crée un nouveau nœud Element.
-
createTextNode(text)
: Crée un nœud Text avec le texte spécifié.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
getElementById(id)
: Récupère un Element par son id.
-
getElementsByName(name)
: Récupère des éléments par leur nom.
-
getElementsByTagName(tagname)
: Récupère des éléments par leur nom de balise.
-
importNode(importedNode, deep)
: Copie un nœud d'un autre document.

Méthodes Netscape 4

-
getSelection()
: Retourne le texte sélectionné du document sans balises HTML.

Méthodes Internet Explorer 4

-
elementFromPoint(x,y)

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Récupère l'Element situé aux coordonnées spécifiées.

Gestionnaires d'événements

L'objet Document supporte des gestionnaires d'événements génériques, similaires à l'objet Element. Les gestionnaires d'événements comme onload et onunload sont des propriétés de l'objet Window.

Sujets associés

Anchor, Applet, Element, Form, Image, Layer, Link, Window.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Exemple

Point clé:Comprendre l'importance de l'objet Document en JavaScript

Exemple:L'objet Document constitue votre interface principale pour manipuler et interagir avec les pages web via JavaScript, vous permettant de modifier dynamiquement les éléments HTML en fonction des actions ou événements des utilisateurs.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Chapitre 20 Résumé : Élément

Objet Element dans le DOM

Aperçu

L'objet Element représente un élément ou une balise HTML dans un document, héritant des propriétés de l'objet Node tel que défini dans le niveau 1 du DOM. Différents navigateurs, y compris IE 4 et ultérieurs, prennent en charge l'accès aux éléments du document, mais avec des différences dans leurs implémentations du DOM.

Propriétés DOM W3C

-
Noms des Propriétés
: Correspondent aux attributs HTML ; les mots réservés nécessitent une nomination spéciale (par exemple, className pour class).
-
Propriétés Courantes

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

:

-

className

: Chaîne de l'attribut de classe.

-

style

: Objet représentant les styles CSS en ligne.

-

tagName

: Nom de balise en lecture seule de l'élément.

Propriétés DOM IE

- Définies à partir de IE 4, distinctes des normes W3C.

-

Propriétés Clés

:

-

all[]

: Tableau des objets Element descendants.

-

children[]

: Tableau des objets Element enfants directs.

-

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

innerHTML
: Texte HTML à l'intérieur de l'élément.
- **offsetHeight/Width**
: Dimensions de l'élément.
- **outerHTML**
: Représentation HTML complète incluant les balises.

Méthodes DOM W3C

- Incluent des méthodes pour manipuler les attributs et les éléments:
- **getAttribute(name)**
: Récupère la valeur d'un attribut.
- **setAttribute(name, value)**
: Définit un attribut nommé.
- **removeAttribute(name)**
: Supprime un attribut nommé.

Méthodes DOM IE

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

- Méthodes non standard présentes dans IE:
  - **contains(target)**
  : Vérifie si l'élément contient un autre élément.
  - **insertAdjacentHTML(where, text)**
  : Insère du HTML à une position spécifiée.
  - **scrollIntoView(top)**
  : Fait défiler la vue pour rendre l'élément visible.

Gestionnaires d'Événements

- Les éléments réagissent à diverses actions de l'utilisateur par le biais de gestionnaires d'événements:
  - **onclick**
  : Sur clic de l'utilisateur.
  - **ondblclick**
  : Sur double clic.
  - **onkeydown/onkeyup**

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Sur activité du clavier.

- **onmousemove/onmouseout**
: Sur mouvement de la souris sur/sortant de l'élément.

Concepts Connus

Références à d'autres objets qui partagent une interaction avec l'objet Element incluent Formulaire, Entrée, Node, Sélection, et Textarea.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 21 Résumé : Événement

Vue d'ensemble de l'objet Événement

L'objet Événement fournit des détails sur un événement et contrôle sa propagation à travers différents environnements de navigateurs. Différents standards et implémentations existent à travers divers navigateurs tels que DOM Level 2, Internet Explorer (IE) et Netscape.

Objet Événement DOM Level 2

-
Types d'événements
: L'objet Événement est utilisé dans les modèles d'événements DOM et Netscape, en le passant comme argument aux gestionnaires d'événements. En revanche, le modèle IE accède à l'événement via la propriété événement de l'objet Window.
-
Constantes de phase d'événement
:
-

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Event.CAPTURING_PHASE
= 1
-
Event.AT_TARGET
= 2
-
Event.BUBBLING_PHASE
= 3

Propriétés DOM

Toutes les propriétés sont en lecture seule :
-
altKey
: Indique si la touche Alt était enfoncée pendant l'événement.
-
bubbles
: Indique si l'événement peut remonter.

Installer l'application Bookey pour débloquer le texte complet et l'audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

App Store
Coup de cœur

22k avis 5 étoiles

Scanner pour télécharger

Retour Positif

Fabienne Moreau

ue résumé de livre ne testent
ion, mais rendent également
musant et engageant.
té la lecture pour moi.

Fantastique!

Giselle Dubois

Je suis émerveillé par la variété de livres et de langues que Bookey supporte. Ce n'est pas juste une application, c'est une porte d'accès au savoir mondial. De plus, gagner des points pour la charité est un grand plus !

Fi

Le
liv
co
pr

é Blanchet

de lecture
eption de
es,
cous.

J'adore !

Isoline Mercier

Bookey m'offre le temps de parcourir les parties importantes d'un livre. Cela me donne aussi une idée suffisante pour savoir si je devrais acheter ou non la version complète du livre ! C'est facile à utiliser !"

Gain de temps !

Bookey est mon applicat
intellectuelle. Les résum
magnifiquement organis
monde de connaissance

ppli géniale !

Joachim Lefevre

adore les livres audio mais je n'ai pas toujours le temps
'écouter le livre entier ! Bookey me permet d'obtenir
n résumé des points forts du livre qui m'intéresse !!!
Quel super concept !!! Hautement recommandé !

Appli magnifique

Cette application est une bouée de sauve
amateurs de livres avec des emplois du te
Les résumés sont précis, et les cartes mer
renforcer ce que j'ai appris. Hautement re

Essai gratuit avec Bookey

Chapitre 22 Résumé : Global

Aperçu global de JavaScript

L'objet Global dans JavaScript contient un ensemble de propriétés et de fonctions globales qui peuvent être accessibles directement sans référence à un autre objet. Les variables et les fonctions créées au niveau supérieur de votre code sont considérées comme des propriétés de l'objet Global. Dans le JavaScript côté client, cet objet Global est représenté par l'objet Window, qui inclut également des méthodes et des propriétés supplémentaires.

Propriétés Globales

- **Infinity** : Représente l'infini positif.
- **NaN** : Représente une valeur non numérique.

Fonctions Globales

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
decodeURI(uri)
: Renvoie une version décodée de l'URI fourni, remplaçant les séquences d'échappement hexadécimales par leurs caractères correspondants.
-
decodeURIComponent(s)
: Semblable à decodeURI mais fonctionne spécifiquement sur les parties composantes d'une URI.
-
encodeURI(uri)
: Encode une URI, remplaçant certains caractères par des séquences d'échappement hexadécimales, tout en laissant les séparateurs comme #, ?, et @ non encodés.
-
encodeURIComponent(s)
: Semblable à encodeURI mais encode tous les caractères de ponctuation.
-
escape(s)
: Renvoie une copie encodée de la chaîne avec certains caractères remplacés (déprécié au profit de encodeURI et encodeURIComponent).

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
eval(code)
: Évalue une chaîne contenant du code JavaScript et renvoie le résultat.
-
isFinite(n)
: Vérifie si n est un nombre fini, renvoyant vrai ou faux.
-
isNaN(x)
: Vérifie si x n'est pas un nombre, renvoyant vrai ou faux.
-
parseFloat(s)
: Convertit une chaîne en nombre et le renvoie.
-
parseInt(s, radix)
: Convertit une chaîne en entier avec un argument radix facultatif spécifiant la base (de 2 à 36).
-
unescape(s)
: Décode une chaîne précédemment encodée avec escape (déprécié au profit de decodeURI et decodeURIComponent).

Voir Également

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
Window
: L'objet qui représente la fenêtre du navigateur et sert de contexte global pour le JavaScript côté client.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 23 Résumé : Input

JavaScript Côté Client 1.0

Présentation de l'Objet Input

L'objet Input représente les éléments HTML `<input>` et fournit des propriétés pour leurs attributs.

Propriétés

- **checked**
: Indique si une case à cocher ou un bouton radio est sélectionné.
- **defaultChecked**
: Indique si la case à cocher ou le bouton radio était sélectionné initialement.
- **defaultValue**
: Texte initial pour les entrées de type texte/mot de passe ;

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

non applicable aux entrées de type fichier.

-

form

: Référence en lecture seule à l'objet Form contenant.

-

name

: L'attribut name de l'élément input.

-

type

: Spécifie le type d'entrée sous forme de chaîne (par exemple, "texte," "case à cocher," etc.).

Valeurs Légales pour le Type

- Comprend des valeurs comme "bouton," "case à cocher," "fichier," "caché," "image," "mot de passe," "radio," "réinitialiser," "texte," et "soumettre."

-

value

: Représente la valeur de chaîne soumise avec le formulaire, variant selon le type d'entrée.

Méthodes

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
blur()
: Retire le focus de l'élément d'entrée.
-
click()
: Simule un clic de souris sur l'entrée.
-
focus()
: Définit le focus clavier sur l'entrée.
-
select()
: Sélectionne le texte dans l'entrée (applicable pour "texte," "mot de passe," et "fichier").

Gestionnaires d'Événements

-
onblur
: Déclenché lorsque l'élément perd le focus.
-
onchange
: Activé lorsque le texte est changé et le focus est déplacé.
-

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

onclick
: Activé lors des clics sur les boutons ; peut empêcher la soumission du formulaire.
-
onfocus
: Déclenché lorsque l'élément obtient le focus.

Références aux Objets Associés

Inclut des associations avec les objets Form, Option, Select et Textarea.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 24 Résumé : Calque

Calque Client-side uniquement Netscape 4

Résumé

- Fournit un aperçu de l'objet Calque, qui est une fonctionnalité non standard exclusive à Netscape 4 pour la gestion des objets positionnés dynamiquement.

Construction de l'objet Calque

- `document.layers[i]`, `document.layers[nom_du_calque]`, et `new Layer(width, parent_layer)` peuvent être utilisés pour référencer ou créer des calques.

Description

- Les calques sont représentés par des éléments HTML avec un positionnement CSS absolu et peuvent être créés en utilisant soit la balise `<layer>` soit le constructeur Layer. Cette fonctionnalité a été abandonnée dans Netscape 6.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Propriétés

-
above
: Référence en lecture seule au calque au-dessus.
-
background
: Image de fond du calque.
-
below
: Référence en lecture seule au calque en dessous.
-
bgColor
: Couleur de fond du calque.
-
clip
: Propriétés définissant les limites et dimensions de la zone

Installer l'application Bookey pour débloquer le texte complet et l'audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

Scanner pour télécharger

Lire, Partager, Autonomiser

Terminez votre défi de lecture, faites don de livres aux enfants africains.

Le Concept

BOOKS FOR AFRICA × ×

Cette activité de don de livres se déroule en partenariat avec Books For Africa. Nous lançons ce projet car nous partageons la même conviction que BFA : Pour de nombreux enfants en Afrique, le don de livres est véritablement un don d'espoir.

La Règle

Gagnez 100 points → **Échangez un livre** → **Faites un don à l'Afrique**

Votre apprentissage ne vous apporte pas seulement des connaissances mais vous permet également de gagner des points pour des causes caritatives ! Pour chaque 100 points gagnés, un livre sera donné à l'Afrique.

Essai gratuit avec Bookey

Chapitre 25 Résumé : Lien

Lien JavaScript côté client 1.0

Héritage et aperçu

- L'objet Lien hérite de la classe Élément. Il représente les éléments `<a>` ou `<area>` dans HTML.

Propriétés de l'objet Lien

- L'objet Lien comprend plusieurs propriétés qui représentent différentes parties d'une URL. Voici les descriptions et exemples basés sur une URL fictive : `http://www.oreilly.com:1234/catalog/search.html?q=JavaScript&m=10#results`.

- **hash** : Une chaîne représentant la partie ancre de l'URL (par exemple, `#result`).

- **host**

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Une chaîne représentant le nom d'hôte et le port (par exemple, `www.oreilly.com:1234`).

- **hostname** : Une chaîne pour le nom d'hôte uniquement (par exemple, `www.oreilly.com`).

- **href** : L'URL complète sous forme de chaîne.

- **pathname** : La partie chemin de l'URL (par exemple, `/catalog/search.html`).

- **port** : Une chaîne représentant le port (par exemple, `1234`).

- **protocol** : Une chaîne pour le protocole de l'URL, incluant le

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

deux-points (par exemple, `http:`).

- **search** : Une chaîne pour la partie requête de l'URL, incluant le point d'interrogation (par exemple, `?q=JavaScript&m=10`).

- **target** : Une chaîne spécifiant le nom de l'objet Fenêtre où le document lié s'ouvrira. Cela inclut les attributs cibles HTML standard (_blank, _top, _parent, et _self).

Gestionnaires d'événements pour l'objet Lien

- **onclick** : Déclenché lorsque le lien est cliqué. Peut retourner false en JavaScript 1.1 pour empêcher le suivi du lien.

- **onmouseout** : Déclenché lorsque la souris quitte le lien, disponible en JavaScript 1.1.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
onmouseover
: Déclenché lorsque la souris survole le lien. Peut modifier la propriété status de la fenêtre et retourner true pour supprimer l'affichage de l'URL dans la ligne de statut.

Concepts connexes

- Inclut des références aux objets Ancre et Localisation pour une meilleure compréhension du comportement et des propriétés des liens.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 26 Résumé : Math

Objet Math en JavaScript

L'objet Math est une caractéristique fondamentale en JavaScript qui organise les constantes et fonctions mathématiques. Contrairement à d'autres objets comme Date et String, il n'a pas de constructeur et ses fonctions sont indépendantes, pas des méthodes d'un objet.

Constantes

- **Math.E**
: Base du logarithme naturel.
- **Math.LN10**
: Logarithme naturel de 10.
- **Math.LN2**
: Logarithme naturel de 2.
- **Math.LOG10E**

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Logarithme en base 10 de e.

-

Math.LOG2E

: Logarithme en base 2 de e.

-

Math.PI

: Constante À.

-

Math.SQRT1_2

: 1 divisé par la racine carrée de 2.

-

Math.SQRT2

: Racine carrée de 2.

Fonctions

-

Math.abs(x)

: Valeur absolue de x.

-

Math.acos(x)

: Arc cosinus de x (0 à À radians).

-

Math.asin(x)

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Arc sinus de x (-À/2 à À/2 radians).

-

Math.atan(x)

: Arc tangente de x (-À/2 à À/2 radians).

-

Math.atan2(y, x)

: Angle antihoraire entre l'axe X positif et le point (x, y).

-

Math.ceil(x)

: Entier le plus proche supérieur ou égal à x.

-

Math.cos(x)

: Cosinus de x.

-

Math.exp(x)

: e élevé à la puissance de x.

-

Math.floor(x)

: Entier le plus proche inférieur ou égal à x.

-

Math.log(x)

: Logarithme naturel de x.

-

Math.max(args...)

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

: Le plus grand des arguments spécifiés ; retourne -Infinity s'il n'y a pas d'arguments ou NaN pour des entrées non numériques.

-

Math.min(args...)

: Le plus petit des arguments spécifiés ; retourne Infinity s'il n'y a pas d'arguments ou NaN pour des entrées non numériques.

-

Math.pow(x, y)

: x élevé à la puissance de y.

-

Math.random()

: Nombre pseudo-aléatoire entre 0.0 et 1.0.

-

Math.round(x)

: Entier le plus proche de x.

-

Math.sin(x)

: Sinus de x.

-

Math.sqrt(x)

: Racine carrée de x (retourne NaN si x < 0).

-

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Math.tan(x)
: Tangente de x.

Sujet connexe

-
Nombre
: Pour plus de détails sur les opérations numériques et les comportements en JavaScript.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 27 Résumé : Navigateur

Informations sur le Navigateur et JavaScript côté client 1.0

Synopsis

Cette section fournit les propriétés et méthodes essentielles de l'objet `navigator` utilisé en JavaScript pour obtenir des informations sur le navigateur utilisé.

Propriétés

- **appCodeName** : Une chaîne indiquant le surnom du navigateur, généralement "Mozilla" pour les navigateurs Netscape et Microsoft.
- **appName** : Une chaîne spécifiant le nom du navigateur, "Netscape" pour les navigateurs Netscape et "Microsoft Internet

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Explorer" pour IE.

-

appVersion

: Une chaîne indiquant la version du navigateur et la plateforme. Le numéro de version peut être extrait en utilisant `parseInt()` ou `parseFloat()`.

-

cookieEnabled

: Une valeur booléenne qui est vraie si les cookies sont activés dans le navigateur.

-

language

: Une chaîne qui spécifie la langue par défaut du navigateur en utilisant des codes à deux lettres standard.

-

platform

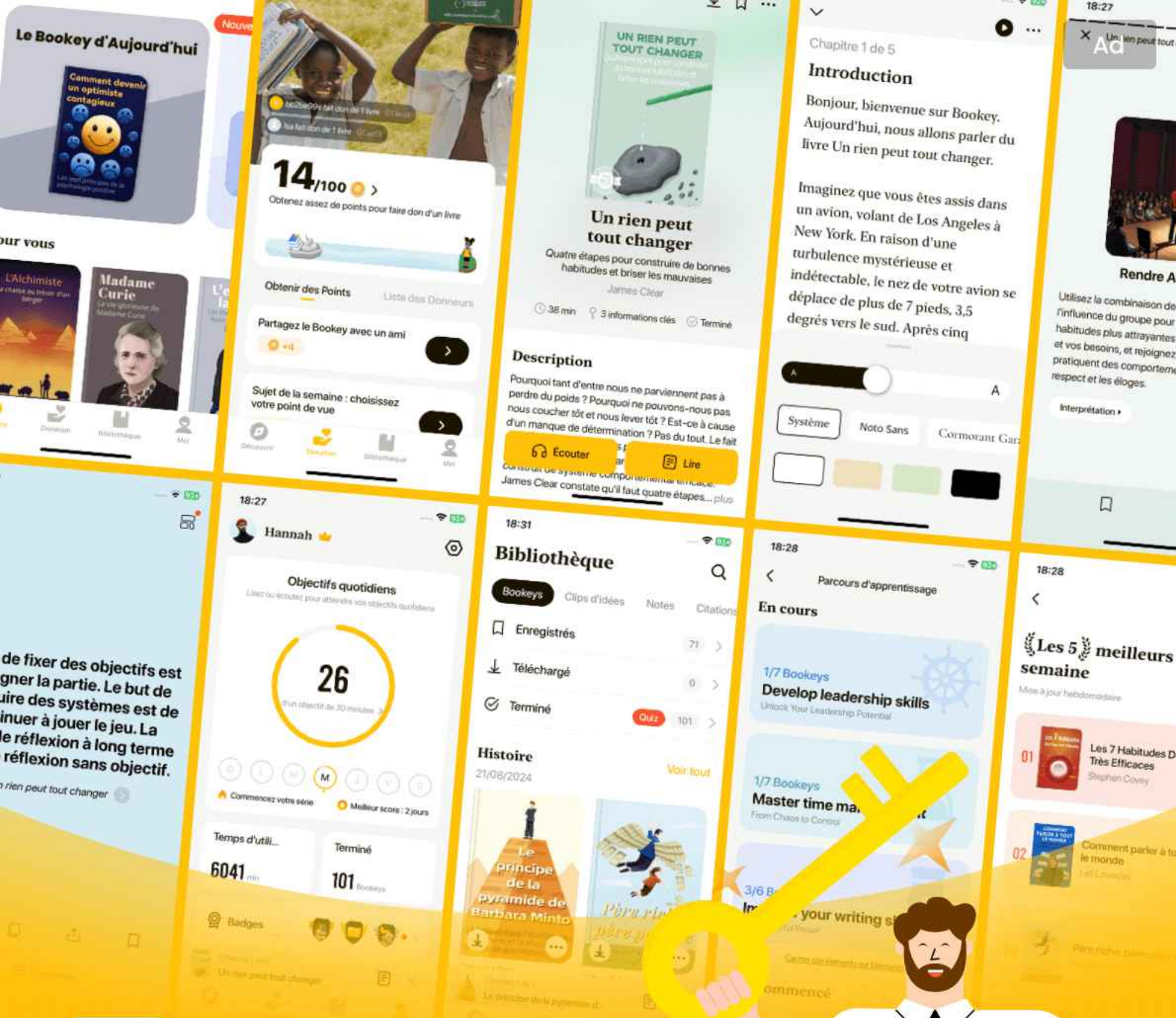
: Une chaîne indiquant le système d'exploitation ou la plateforme matérielle, comme "Win32" ou "Linux i586".

Installer l'application Bookey pour débloquer le texte complet et l'audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé



Les meilleures idées du monde débloquent votre potentiel

Essai gratuit avec Bookey

Télécharger dans App Store

DISPONIBLE SUR Google Play

Scanner pour télécharger

Chapitre 28 Résumé : Node

Niveau 1 du DOM Node

Définition

Un Node dans un arbre de document est une instance de l'une des plusieurs sous-classes, y compris Attr, Comment, Document, DocumentFragment, Element et Text. Chaque objet Node a une propriété `nodeType` qui indique sa sous-classe spécifique.

Constantes de Type de Node

Les constantes suivantes définissent les valeurs légales pour `nodeType` :
- `Node.ELEMENT_NODE = 1` (Élément)
- `Node.ATTRIBUTE_NODE = 2` (Attr)
- `Node.TEXT_NODE = 3` (Texte)
- `Node.COMMENT_NODE = 8` (Commentaire)
- `Node.DOCUMENT_NODE = 9` (Document)
- `Node.DOCUMENT_FRAGMENT_NODE = 11`

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

(DocumentFragment)

Propriétés de Node

- `attributes[]`: Tableau en lecture seule d'objets Attr pour les nœuds élément.
- `childNodes[]`: Tableau en lecture seule d'objets Node enfants.
- `firstChild`: Le premier nœud enfant ou null s'il n'en existe pas.
- `lastChild`: Le dernier nœud enfant ou null s'il n'en existe pas.
- `nextSibling`: Le nœud frère suivant ou null s'il n'en existe pas.
- `nodeName`: Le nom du nœud, varie selon le type de nœud.
- `nodeType`: Spécifie le type du nœud.
- `nodeValue`: Contient la valeur sous forme de chaîne, lecture/écriture pour les nœuds Texte et Commentaire.
- `ownerDocument`: L'objet Document contenant ce Node ou null pour les nœuds Document.
- `parentNode`: Le nœud conteneur ou null pour les nœuds sans parent.
- `previousSibling`: Le nœud frère précédent ou null s'il n'en existe pas.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Méthodes de Node

- `addEventListener(type, listener, useCapture)`: Enregistre un écouteur d'événements.
- `appendChild(newChild)`: Ajoute un nouveau nœud enfant à l'arbre de document.
- `cloneNode(deep)`: Retourne une copie du nœud, incluant éventuellement les descendants.
- `hasAttributes()`: Vérifie la présence d'attributs dans un nœud élément.
- `hasChildNodes()`: Vérifie si le nœud a des enfants.
- `insertBefore(newChild, refChild)`: Insère un nouvel enfant avant un enfant de référence spécifié.
- `isSupported(feature, version)`: Vérifie la prise en charge d'une version de fonctionnalité.
- `normalize()`: Consolidation des nœuds de Texte adjacents et suppression des nœuds de Texte vides.
- `removeChild(oldChild)`: Supprime un nœud enfant spécifié.
- `removeEventListener(type, listener, useCapture)`: Supprime un écouteur d'événements.
- `replaceChild(newChild, oldChild)`: Remplace un ancien nœud enfant par un nouveau nœud enfant.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Sous-classes Associées

Inclut Attr, Comment, Document, DocumentFragment, Element et Text.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 29 Résumé : Nombre

Résumé du Chapitre 29 : Nombres en JavaScript

Constructeur Number

- Le constructeur `Number` peut être utilisé avec l'opérateur `new` pour créer un nouvel objet `Number`. Sans `new`, il fonctionne comme une fonction de conversion, renvoyant une valeur numérique.

Constantes

- **Number.MAX_VALUE** : Le plus grand nombre représentable (~1.79E+308).
- **Number.MIN_VALUE** : Le plus petit nombre représentable (~5E-324).
- **Number.NaN** : Représente 'Pas un nombre'.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

- **Number.NEGATIVE_INFINITY** : Valeur infinie négative.
- **Number.POSITIVE_INFINITY** : Valeur infinie positive.

Méthodes

- **toExponential(digits)** : Renvoie une chaîne en notation exponentielle avec un nombre spécifié de chiffres après la virgule (0-20).
- **toFixed(digits)** : Renvoie une chaîne avec exactement le nombre spécifié de chiffres décimaux (0-20) sans notation exponentielle.
- **toLocaleString()** : Renvoie une représentation de chaîne dépendante de la locale du nombre.
- **toPrecision(precision)** : Renvoie une représentation de chaîne contenant un nombre

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

spécifié de chiffres significatifs (1-21).

- **toString(radix)** : Convertit un nombre en chaîne dans une base spécifiée (2-36) ; par défaut, c'est la base 10 si omise.

Référence Associée

- Des informations supplémentaires peuvent être trouvées dans la documentation de l'objet Math.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 30 Résumé : Objet

Objet en JavaScript

Constructeur

- Le constructeur `new Object();` crée un objet vide auquel il est possible d'ajouter des propriétés arbitraires.

Propriétés des objets JavaScript

- **constructor**
: Une référence à la fonction qui a créé l'objet (JS 1.1; JScript 2.0; ECMA v1).

Méthodes des objets JavaScript

1. **hasOwnProperty(propname)**
:
   - Renvoie `true` si l'objet a une propriété non héritée avec

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

le nom spécifié, sinon renvoie `false` (JS 1.5; JScript 5.5; ECMA v3).

2.
isPrototypeOf(o)
:
- Renvoie `true` si l'objet est le prototype de `o`, sinon renvoie `false` (JS 1.5; JScript 5.5; ECMA v3).

3.
propertyIsEnumerable(propname)
:
- Renvoie `true` si l'objet a une propriété énumérable non héritée avec le nom spécifié (JS 1.5; JScript 5.5; ECMA v3).

4.
toLocaleString()
:

Installer l'application Bookey pour débloquer le texte complet et l'audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

Scanner pour télécharger

Essayez l'appli Bookey pour lire plus de 1000 résumés des meilleurs livres du monde

Débloquez 1000+ titres, 80+ sujets

Nouveaux titres ajoutés chaque semaine

Brand | Leadership & collaboration | Gestion du temps | Relations & communication | Know

tégie d'entreprise | Créativité | Mémoires | Argent & investissements | Positive Psychology

Entrepreneuriat | Histoire du monde | Communication parent-enfant | Soins Personnels

Aperçus des meilleurs livres du monde

Essai gratuit avec Bookey

Chapitre 31 Résumé : RegExp

RegExp en JavaScript

Vue d'ensemble

Les expressions régulières (RegExp) sont utilisées en JavaScript pour la correspondance de motifs. Elles peuvent être définies en utilisant la syntaxe littérale ou en employant un constructeur.

Syntaxe

- **Syntaxe Littérale** : `/motif/attributs`
- **Constructeur** : `new RegExp(motif, attributs)`

Propriétés d'Instance

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
global
: Indique si la RegExp a l'attribut 'g' pour une correspondance globale (booléen en lecture seule).
-
ignoreCase
: Indique si la RegExp a l'attribut 'i' pour une correspondance insensible à la casse (booléen en lecture seule).
-
lastIndex
: Pour les objets globaux, cette propriété (lecture/écriture) montre la position après la dernière correspondance pour la prochaine recherche.
-
multiline
: Indique si la RegExp a l'attribut 'm' pour une correspondance sur plusieurs lignes (booléen en lecture seule).
-
source
: Une chaîne en lecture seule contenant le motif de l'expression régulière sans barres obliques ni attributs.

Méthodes

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

-
exec(chaine)
: Effectue une correspondance d'une chaîne avec la RegExp et retourne un tableau des résultats de correspondance ou null si aucune correspondance n'est trouvée. Le tableau inclut le texte correspondant et toute correspondance de sous-expression, ainsi qu'une propriété d'index indiquant la position de début de la correspondance.
-
test(chaine)
: Retourne true si la chaîne contient une correspondance pour la RegExp ; sinon, retourne false.

Méthodes Associées

- String.match()
- String.replace()
- String.search()

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 32 Résumé : Sélection

JavaScript côté client 1.0 : Aperçu de l’objet Select

Héritage

- Hérite de l’objet Element.

Synopsis

- Accédé via :
  - `form.elements[i]`
  - `form.elements[element_name]`
  - `form.element_name`

Propriétés

- **form** : L’objet Form contenant (lecture seule).
- **length**

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

: Le nombre d’éléments dans le tableau options (lecture seule).

-

options[]

: Tableau d’objets Option ; peut être manipulé en modifiant `options.length` ou en mettant des éléments individuels à null.

-

selectedIndex

: Entier en lecture/écriture pour l’index de l’option sélectionnée ; -1 si aucune sélectionnée. Désélectionne toutes si mis à -1.

-

type

: Chaîne en lecture seule indiquant le type de sélection ("select-one" pour unique, "select-multiple" pour multiple).

Méthodes

-

add(new, old)

: Insère un nouvel objet Option avant l'ancien dans le tableau options ; l’ajoute à la fin si old est null.

-

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

blur()
: Libère le focus du clavier.
-
focus()
: Obtient le focus du clavier.
-
remove(n)
: Supprime le nième élément du tableau options.

Gestionnaires d’événements

-
onblur
: Déclenché lorsque le focus est perdu.
-
onchange
: Déclenché lorsqu’un élément est sélectionné/désélectionné.
-
onfocus
: Déclenché lorsqu’on obtient le focus.

Concepts associés

- Référez-vous aux objets Form, Input et Option pour plus de contexte.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 33 Résumé : Chaîne

Chaîne en JavaScript

Présentation

- La manipulation des chaînes est une fonctionnalité clé de JavaScript, introduite dans JavaScript 1.0, JScript 1.0, et normalisée dans l'ECMA v1.
- Le type `String` hérite de l'objet de base `Object`.

Constructeur

- `String(s)` - Sans l'opérateur `new`, il convertit son argument en chaîne. Avec `new`, il crée un objet String.

Propriétés

- `length` - Une propriété en lecture seule représentant le nombre de caractères dans la chaîne.

Méthodes

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

1.
Accès aux Caractères

- `charAt(n)` - Renvoie le caractère à la position `n`.
- `charCodeAt(n)` - Renvoie l'encodage Unicode du caractère à la position `n`.

2.
Manipulation de Chaînes

- `concat(value...)` - Concatène des chaînes.
- `indexOf(substring, start)` - Renvoie la position de la première occurrence de `substring` après `start` ou -1 si non trouvé.
- `lastIndexOf(substring, start)` - Renvoie la position de la dernière occurrence avant `start` ou -1 si non trouvé.

3.
Correspondance de Modèles

Installer l'application Bookey pour débloquer le texte complet et l'audio

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

Scanner pour télécharger

Pourquoi Bookey est une application incontournable pour les amateurs de livres

Contenu de 30min
Plus notre interprétation est profonde et claire, mieux vous saisissez chaque titre.

Format texte et audio
Absorbez des connaissances même dans un temps fragmenté.

Quiz
Vérifiez si vous avez maîtrisé ce que vous venez d'apprendre.

Et plus
Plusieurs voix & polices, Carte mentale, Citations, Clips d'idées...

Essai gratuit avec Bookey

Chapitre 34 Résumé : Style

Style DOM Niveau 2 : Propriétés CSS en ligne dans IE 4

Synopsis

L'objet Style en JavaScript est utilisé pour manipuler les propriétés CSS en ligne d'un élément. Il comprend des propriétés correspondant aux attributs CSS définis par la spécification CSS2. La nomenclature de ces propriétés imite de près leurs homologues CSS, avec des modifications pour garantir la compatibilité avec JavaScript, comme la suppression des traits d'union et la capitalisation des mots suivants.

Propriétés Clés

- Les attributs multi-mots comme `font-family` sont réécrits en `fontFamily`.
- La propriété `float` est renommée `cssFloat` en raison de conflits avec des mots réservés en JavaScript.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Notes Importantes

- Toutes les propriétés sont représentées sous forme de chaînes. Pour les propriétés avec des valeurs numériques, il est essentiel d'utiliser `parseFloat()` pour convertir les chaînes en nombres lors des interrogations, et de convertir les nombres de nouveau en chaînes lors de la définition des propriétés (typiquement en ajoutant des unités comme "px").
- Toutes les propriétés ne sont pas mises en œuvre par les navigateurs actuels, il est donc conseillé de se référer aux références CSS pour les significations et les valeurs valides.

Liste des Propriétés CSS Visuelles

- background, backgroundColor, backgroundImage, backgroundPosition, backgroundRepeat
- border (divers sous-propriétés), borderCollapse, borderColor, borderStyle, borderWidth
- color, cursor, display, direction
- font (divers sous-propriétés), fontSize, fontWeight, fontStyle
- height, left, right, top, width

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

- margin (divers sous-propriétés), padding (divers sous-propriétés)
- overflow, position, visibility, zIndex

Ces propriétés permettent une manipulation détaillée de la présentation d'un élément dans le développement web. Pour des conseils complets, consultez des références CSS spécifiques.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 35 Résumé : Window

Résumé du Chapitre 35 : Objet Window et JavaScript côté client

Synopsis

Ce chapitre aborde l'objet Window dans JavaScript côté client, détaillant ses propriétés, méthodes, gestion des événements et leur impact sur le scripting dans les navigateurs web.

Propriétés de l'Objet Window

-
Propriétés de Base :

- `closed` : Booléen indiquant si la fenêtre est fermée.
- `defaultStatus` : Message d'état affiché lorsqu'aucun autre message n'est présenté.
- `document` : Référence à l'objet Document dans la fenêtre.
- `frames[]` : Tableau d'objets Window représentant chaque

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

cadre au sein de la fenêtre.
- `location` : Objet location représentant l'URL actuelle de la fenêtre.
- `navigator` : Référence à l'objet Navigator fournissant les informations sur le navigateur.

-
Propriétés Spécifiques à Netscape 4 :

- `innerHeight`, `innerWidth` : Dimensions de la zone d'affichage du document.
- `outerHeight`, `outerWidth` : Dimensions totales de la fenêtre.
- `pageXOffset`, `pageYOffset` : Positions de défilement.

-
Propriétés Spécifiques à Internet Explorer 4 :

- `clientInformation` : Alias pour navigator.
- `event` : Référence à l'objet d'événement le plus récent.

Méthodes de l'Objet Window

-
Méthodes Courantes :

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

- `alert()` : Affiche un message dans une boîte de dialogue.
- `confirm()` : Affiche une boîte de dialogue oui/non et renvoie le choix de l'utilisateur.
- `prompt()` : Demande une saisie utilisateur avec une valeur par défaut.
- `setTimeout()` et `setInterval()` : Planifient des actions après des intervalles de temps spécifiés.
- `open()` : Ouvre une nouvelle fenêtre avec une URL et des fonctionnalités spécifiées.

Gestionnaires d'Événements

C'est défini à l'aide d'attributs au sein des balises HTML, tels que `onload`, `onresize` et `onclick`, qui déclenchent des fonctions JavaScript lors d'événements spécifiques.

Restrictions de Sécurité JavaScript

Les mesures de sécurité limitent les capacités des scripts côté client, comme la prévention d'accès à différents domaines et le contrôle des fonctionnalités de téléchargement de fichiers.

Conclusion

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

L'objet Window constitue la base de JavaScript côté client, permettant d'interagir avec les pages web grâce à des méthodes et des propriétés qui influencent le comportement du navigateur et la présentation du contenu.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Exemple

Point clé:Comprendre l'objet Window est essentiel pour un développement efficace de JavaScript côté client.

Exemple:En explorant JavaScript côté client, imaginez que vous créez une application web dynamique. Vous pourriez utiliser la méthode `alert()` pour informer les utilisateurs des mises à jour importantes, les incitant à prêter attention. En manipulant la propriété `location`, imaginez que vous redirigez les utilisateurs vers une expérience plus personnalisée sur votre site, améliorant leur parcours. En définissant des gestionnaires d'événements avec des attributs comme `onclick`, pensez à créer un bouton qui réagit vraiment aux clics des utilisateurs, offrant un retour immédiat ou des actions. Chaque interaction dépend de votre compréhension de l'objet Window, car il régit toutes ces fonctions essentielles, rendant vos scripts non seulement interactifs mais aussi intuitifs et conviviaux.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

App Store
Coup de cœur

22k avis 5 étoiles

Scanner pour télécharger

Retour Positif

Fabienne Moreau

ue résumé de livre ne testent
ion, mais rendent également
musant et engageant.
té la lecture pour moi.

Fantastique!

Giselle Dubois

Je suis émerveillé par la variété de livres et de langues que Bookey supporte. Ce n'est pas juste une application, c'est une porte d'accès au savoir mondial. De plus, gagner des points pour la charité est un grand plus !

Fi

Le
liv
co
pr

é Blanchet

de lecture
eption de
es,
cous.

J'adore !

Isoline Mercier

Bookey m'offre le temps de parcourir les parties importantes d'un livre. Cela me donne aussi une idée suffisante pour savoir si je devrais acheter ou non la version complète du livre ! C'est facile à utiliser !"

Gain de temps !

Bookey est mon applicat
intellectuelle. Les résum
magnifiquement organis
monde de connaissance

ppli géniale !

Joachim Lefevre

adore les livres audio mais je n'ai pas toujours le temps
'écouter le livre entier ! Bookey me permet d'obtenir
n résumé des points forts du livre qui m'intéresse !!!
Quel super concept !!! Hautement recommandé !

Appli magnifique

Cette application est une bouée de sauve
amateurs de livres avec des emplois du te
Les résumés sont précis, et les cartes mer
renforcer ce que j'ai appris. Hautement re

Essai gratuit avec Bookey

Meilleures phrases du JavaScript par David Flanagan avec numéros de page

Voir sur le site de Bookey et générer de belles images de citation

Chapitre 1 | Phrases des pages 14-17

1. JavaScript est un langage sensible à la casse.
2. JavaScript ignore les espaces blancs entre les tokens.
3. Les instructions JavaScript se terminent par des points-virgules.
4. Les noms de variables, de fonctions et d'étiquettes sont des identifiants JavaScript.
5. Les mots-clés suivants font partie du langage JavaScript et ont une signification spéciale pour l'interpréteur JavaScript.

Chapitre 2 | Phrases des pages 20-27

1. Les nombres en JavaScript sont représentés en format flottant 64 bits.
2. Une chaîne JavaScript est une séquence de lettres, de chiffres et d'autres caractères issus de l'ensemble de caractères Unicode 16 bits.
3. Un objet est un type de données composé qui contient un

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

nombre quelconque de propriétés.

4. Le premier élément d'un tableau JavaScript est l'élément 0.
5. Une fonction est un morceau de code JavaScript qui est défini une fois et peut être exécuté plusieurs fois par un programme.

Chapitre 3 | Phrases des pages 28-32

1. Les expressions JavaScript se forment en combinant des valeurs à l'aide d'opérateurs JavaScript.
2. Les parenthèses peuvent être utilisées dans une expression pour regrouper des sous-expressions et modifier l'ordre d'évaluation par défaut.
3. JavaScript définit un ensemble complet d'opérateurs, dont la plupart devraient être familiers aux programmeurs C, C++ et Java.
4. L'opérateur d'identité, ===, est plus strict : il n'évalue à vrai que si ses opérandes sont identiques.
5. Les opérateurs relationnels <, <=, > et >= comparent les chaînes en fonction de l'ordre alphabétique.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

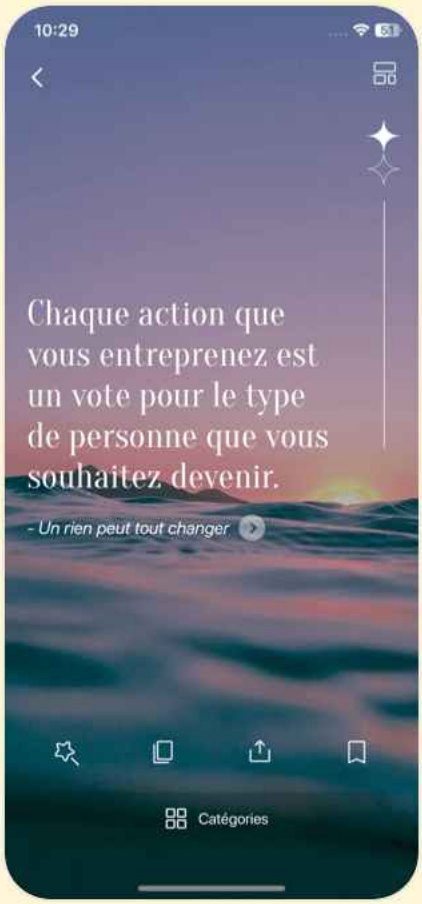
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 4 | Phrases des pages 33-40

1. Lorsqu'une séquence d'instructions JavaScript est encadrée par des accolades, cela compte comme une seule instruction composée.
2. L'instruction return provoque l'arrêt de l'exécution de la fonction en cours et retourne à son appelant.
3. L'instruction try/catch/finally est le mécanisme de gestion des exceptions en JavaScript.
4. L'instruction for est une boucle facile à utiliser qui combine les expressions d'initialisation et d'incrément avec l'expression de condition de boucle.
5. Toute instruction peut être étiquetée avec un nom.

Chapitre 5 | Phrases des pages 41-43

1. Les objets JavaScript sont des tableaux associatifs qui associent des valeurs à des propriétés nommées.
2. Pour définir une nouvelle classe, commencez par écrire une fonction constructeur.
3. Chaque fonction JavaScript utilisée comme constructeur a

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

une propriété nommée prototype.

4. Définir une méthode nommée toString permet aux instances de votre classe d'être converties en chaînes de caractères.
5. Si vous souhaitez définir des méthodes ou des propriétés statiques (ou de classe), vous pouvez les assigner directement à la fonction constructeur.
6. Les fragments de code précédents définissent une simple classe Point que nous pouvons utiliser avec du code comme celui-ci

Chapitre 6 | Phrases des pages 44-48

1. JavaScript prend en charge les expressions régulières pour la correspondance de motifs avec la même syntaxe que le langage de programmation Perl.
2. Une expression régulière est spécifiée littéralement dans un programme JavaScript comme une séquence de caractères entourée de caractères slash (/).
3. Dans JavaScript 1.5, n'importe quel caractère de répétition

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

peut être suivi d'un point d'interrogation pour les rendre non-greedy, ce qui signifie qu'ils correspondent à aussi peu de répétitions que possible tout en permettant toujours à l'ensemble du motif de correspondre.

4. Les expressions régulières utilisent des parenthèses pour grouper des sous-expressions, tout comme le font les expressions mathématiques.
5. Un ancre dans une expression régulière correspond à une position dans une chaîne sans correspondre à aucun des caractères d'une chaîne.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

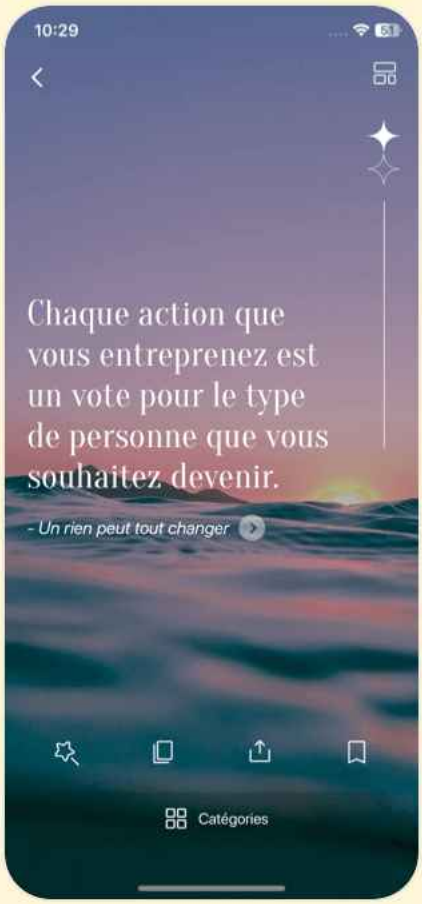
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 7 | Phrases des pages 49-90

1. JavaScript est un langage de script léger basé sur des objets qui peut être intégré dans des pages HTML.
2. Le code JavaScript peut également apparaître comme la valeur d'un attribut de gestionnaire d'événements d'une balise HTML.
3. Cependant, lorsque vous faites cela, vous devez écrire le contenu complet du nouveau document, et n'oubliez pas d'appeler la méthode document.close() lorsque vous avez terminé.
4. L'objet Window représente une fenêtre de navigateur web.
5. JavaScript considère chaque cadre comme un objet Window séparé, et les scripts dans différents cadres s'exécutent indépendamment les uns des autres.
6. DHTML, ou HTML dynamique, est le résultat de la combinaison de HTML, CSS et JavaScript : il utilise des scripts pour modifier dynamiquement le style qui peut inclure la position et la visibilité des éléments du

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

document.

7. Pour des raisons de sécurité, les implémentations de JavaScript côté client imposent généralement des restrictions sur les tâches que les scripts peuvent effectuer.

Chapitre 8 | Phrases des pages 93-96

1. Dans JavaScript 1.1 et versions ultérieures, vous pouvez utiliser l'attribut src de la balise <script> pour spécifier l'URL d'un script externe à charger et à exécuter.
2. HTML permet d'écrire des scripts dans des langages autres que JavaScript, et certains navigateurs, comme Internet Explorer, supportent des langages tels que VBScript.
3. Pour JavaScript, définissez cet attribut au type MIME "text/javascript" : <script src="functions.js" language="JavaScript1.5" type="text/javascript"></script>
4. Les noms des attributs de gestion des événements commencent toujours par "on". Le code spécifié par l'un de ces attributs est exécuté lorsque l'événement nommé se produit.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 9 | Phrases des pages 97-105

1. L'objet Window représente une fenêtre de navigateur web.
2. La propriété la plus importante de l'objet Window est document.
3. alert( ) permet d'afficher un message à l'utilisateur, confirm( ) permet de poser à l'utilisateur une question avec un oui ou un non, et prompt( ) permet de demander à l'utilisateur d'entrer une seule ligne de texte.
4. Vous pouvez également utiliser des minuteurs pour spécifier du code à exécuter lorsqu'un certain nombre de millisecondes s'est écoulé.
5. La propriété history de l'objet Window fait référence à l'objet History pour la fenêtre du navigateur.
6. Plus important que ces méthodes qui manipulent une fenêtre existante, il y a la méthode open( ) qui crée une nouvelle fenêtre de navigateur et la méthode close( ) qui ferme une fenêtre créée par le script.
7. JavaScript traite chaque cadre comme un objet Window

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

séparé, et les scripts dans différents cadres s'exécutent indépendamment les uns des autres.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

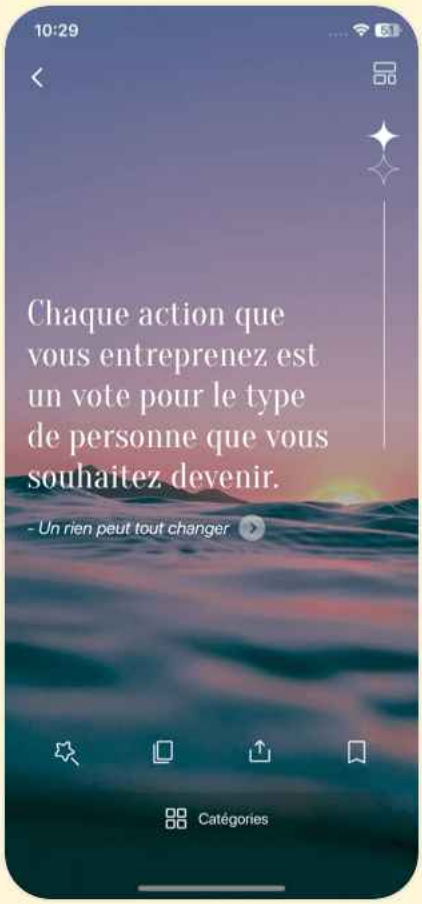
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 10 | Phrases des pages 106-108

1. L'objet Document est sans doute plus important que l'objet Window lui-même : tandis que Window représente la fenêtre du navigateur, l'objet Document représente le document HTML affiché dans cette fenêtre.
2. La manière dont le contenu du document est accédé et modifié s'appelle le modèle d'objet document, ou DOM.
3. Le DOM W3C n'est pas étroitement compatible avec le DOM IE 4, mais il standardise de nombreuses fonctionnalités héritées de l'ancien DOM.

Chapitre 11 | Phrases des pages 109-117

1. Le DOM hérité ne fournit aucun moyen de référencer le contenu du document en dehors des formulaires, des éléments de formulaire, des images, des applets, des liens et des ancres.
2. La méthode write( ) ne peut être utilisée à partir d'une balise <script> que tant qu'un document est encore en cours de chargement.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

3. Si le gestionnaire d'événements onsubmit renvoie faux, le formulaire n'est pas soumis.
4. Pour changer l'image affichée, il suffit de définir cette propriété sur une nouvelle URL.
5. Un cookie avec le nom et la valeur spécifiés pour ce document.

Chapitre 12 | Phrases des pages 118-124

1. Le W3C DOM standardise la plupart des fonctionnalités de l'ancien DOM, mais y ajoute également des éléments nouveaux importants.
2. Le W3C DOM représente chaque document sous forme d'arbre.
3. Une manière simple de manipuler les documents HTML avec le W3C DOM est tout simplement de définir les propriétés qui correspondent aux attributs HTML.
4. Le W3C DOM vous permet de modifier la structure en arbre du document lui-même.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

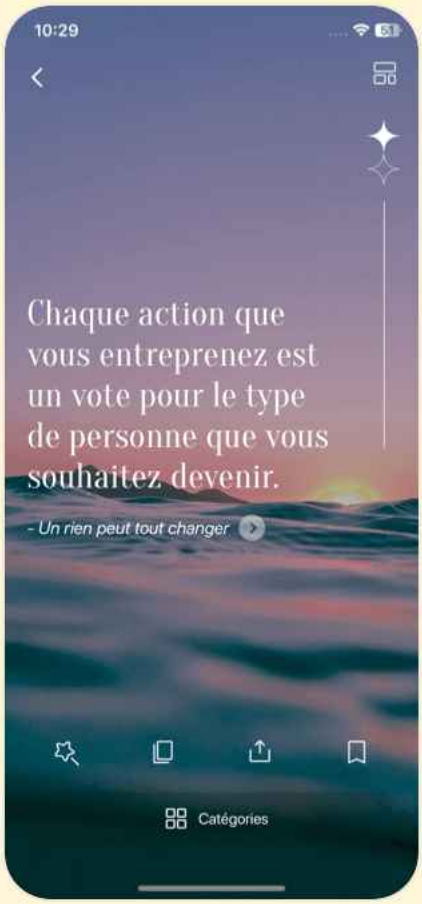
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 13 | Phrases des pages 125-129

1. Le DOM d'IE 4 ne supporte pas la méthode getElementById().
2. Vous pouvez parcourir un arbre de document IE 4... au lieu de childNodes[], IE 4 utilise children[], et au lieu de parentNode, IE 4 utilise parentElement.
3. Le DOM d'IE 4 n'a pas de méthodes pour créer, insérer, supprimer ou remplacer explicitement des nœuds de l'arbre du document.
4. Cependant, il supporte la propriété très importante innerHTML, qui vous permet de remplacer le contenu de n'importe quel élément de document par une chaîne arbitraire de HTML.

Chapitre 14 | Phrases des pages 130-134

1. DHTML, ou HTML dynamique, est le résultat de la combinaison de HTML, CSS et JavaScript : il utilise des scripts pour modifier dynamiquement le style...
2. Les propriétés de positionnement et de visibilité sont

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

particulièrement pertinentes pour le scripting dynamique.

3. Notez que les valeurs de toutes les propriétés de style sont toujours des chaînes de caractères, même pour des propriétés comme left et width qui représentent des nombres.
4. Chaque fois qu'elle est appelée, la fonction nextFrame( ) déplace un élément de 10 pixels vers la droite...
5. Après 20 invocations, la fonction utilise la propriété de visibilité pour cacher l'élément et cesse de s'appeler elle-même.

Chapitre 15 | Phrases des pages 135-141

1. En général, si un gestionnaire d'événements renvoie false, l'action par défaut (comme suivre un lien ou soumettre un formulaire) n'est pas effectuée.
2. Il en va de même pour les gestionnaires d'événements. Si votre document HTML inclut une seule balise <form> avec un attribut de gestionnaire d'événements onsubmit, ce gestionnaire est accessible sous :

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

document.forms[0].onsubmit

3. Les gestionnaires d'événements ne sont déclenchés que pour l'élément du document sur lequel l'événement s'est produit.
4. Dans les modèles avancés, les événements peuvent se propager vers le haut et/ou vers le bas dans la hiérarchie des éléments et être gérés par un ou plusieurs gestionnaires d'événements.
5. Cela permet aux applications avancées d'enregistrer plus d'un gestionnaire pour le même type d'événement.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

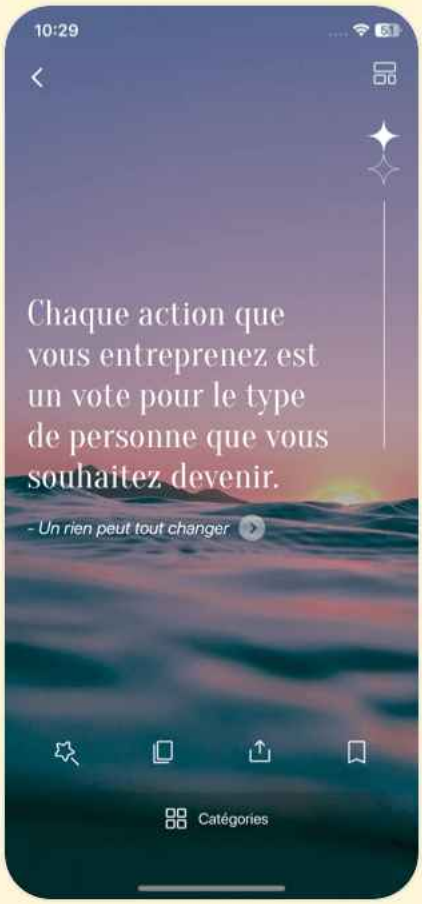
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 16 | Phrases des pages 142-145

1. 'Les scripts ne peuvent lire que les propriétés des fenêtres et des documents chargés depuis le même serveur web.'
2. 'Les scripts ne peuvent pas définir la propriété de valeur de l'élément de formulaire FileUpload.'
3. 'Un script ne peut fermer que les fenêtres de navigateur qu'il a lui-même créées, à moins d'obtenir la confirmation de l'utilisateur.'
4. 'Avec l'essor de JavaScript, les publicitaires et les personnages peu scrupuleux ont commencé à en faire des choses agaçantes.'
5. 'Les navigateurs plus récents, comme Mozilla 1.0, permettent des restrictions de sécurité configurables par l'utilisateur qui peuvent empêcher les scripts d'ouvrir de nouvelles fenêtres.'

Chapitre 17 | Phrases des pages 156-159

1. Par exemple : var a = [1, true, 'abc'];
2. Changer la valeur de cette propriété tronque ou étend le

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

tableau.

3. Renvoie la chaîne qui résulte de la conversion de chaque élément d'un tableau en chaîne, puis de la concaténation des chaînes ensemble, avec la chaîne de séparateur entre les éléments.
4. Supprime et renvoie le dernier élément du tableau, en diminuant la longueur du tableau.
5. Trie les éléments d'un tableau et renvoie une référence au tableau.
6. Insère l'argument ou les arguments en tant que nouveaux éléments au début d'un tableau, déplaçant les éléments du tableau existants vers le haut pour faire de la place.

Chapitre 18 | Phrases des pages 171-177

1. Sans arguments, le constructeur Date( ) crée un objet Date réglé sur la date et l'heure actuelles.
2. Lorsqu'il est appelé comme une fonction sans l'opérateur new, Date( ) ignore tous les arguments qui lui sont passés et renvoie une représentation sous forme de chaîne de caractères de la date et de l'heure actuelles.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

3. Renvoie le nombre de millisecondes entre minuit (UTC) du 1er janvier 1970 et la date et l'heure représentées par l'objet Date.
4. Renvoie une chaîne qui représente la partie date de la date, exprimée dans le fuseau horaire local.
5. Définit la représentation interne de la date en millisecondes. Renvoie l'argument en millisecondes.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

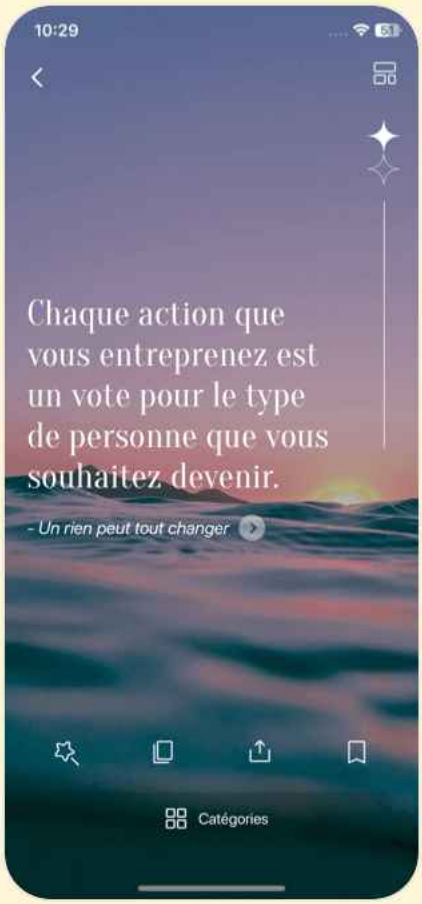
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 19 | Phrases des pages 178-186

1. L'objet Document représente un document HTML et est l'un des objets les plus importants dans JavaScript côté client.
2. Propriétés communes Toutes les implémentations de l'objet Document prennent en charge les propriétés suivantes.
3. L'API dépend du contrôle spécifique.
4. Cette propriété peut être particulièrement utile pour les programmeurs qui passent à JavaScript après s'être habitués à des langages insensibles à la casse.
5. readyState spécifie le statut de chargement d'un document.

Chapitre 20 | Phrases des pages 189-197

1. .... tous les éléments d'un document HTML possèdent des propriétés qui correspondent à leurs attributs HTML...
2. Malheureusement, les méthodes et propriétés définies par le DOM IE 4 ne sont pas les mêmes que celles définies par la norme DOM W3C.
3. Définir cette propriété remplace le contenu de l'élément.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

4. Définir les propriétés de cet objet change le style d'affichage de l'élément.
5. Texte HTML contenu dans l'élément, sans inclure les balises d'ouverture et de fermeture de l'élément lui-même.

Chapitre 21 | Phrases des pages 200-208

1. L'objet Event sert à fournir à la fois des détails sur un événement et un contrôle sur la propagation de cet événement.
2. DOM Level 2 définit un objet Event standard, mais Internet Explorer 4, 5 et 6 utilisent un objet propriétaire à la place.
3. Un objet Event est passé comme argument au gestionnaire d'événements.
4. La phase actuelle de la propagation d'événements est représentée par la propriété eventPhase.
5. Indique au navigateur web de ne pas effectuer l'action par défaut associée à cet événement.
6. Arrête l'événement de se propager davantage à travers les phases de capture, de cible ou de bulle de propagation des événements.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

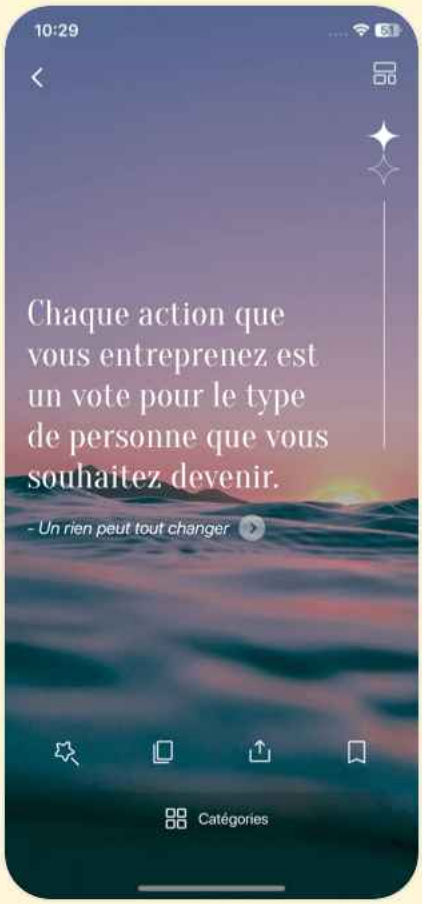
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 22 | Phrases des pages 215-218

1. L'objet global n'a pas de nom, mais vous pouvez y faire référence dans le code de haut niveau (c'est-à-dire en dehors des méthodes) avec le mot-clé this.
2. L'infini : une valeur numérique qui représente l'infini positif.
3. NaN : la valeur non numérique.
4. eval(code) Évalue une chaîne de code JavaScript et retourne le résultat.
5. isFinite(n) : retourne vrai si n est (ou peut être converti en) un nombre fini.
6. parseInt(s, radix) : convertit la chaîne s (ou un préfixe de s) en entier et retourne cet entier.

Chapitre 23 | Phrases des pages 224-228

1. L'objet Input définit des propriétés pour chacun des attributs de la balise HTML <input>.
2. checked Un booléen en lecture/écriture qui spécifie si un élément d'entrée de type 'checkbox' ou 'radio' est coché

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

(vrai) ou non (faux).

3. defaultValue Une chaîne de caractères qui spécifie le texte qui apparaît dans un élément d'entrée de type 'text' ou 'password' lorsqu'il est créé pour la première fois ou lorsqu'il est réinitialisé à son état initial.
4. value La valeur en chaîne qui est envoyée lorsque le formulaire est soumis.
5. click( ) Simule un clic de souris sur l'élément de formulaire et ne renvoie rien.
6. onchange Pour les éléments de saisie de texte de type 'text', 'password' et 'file', ce gestionnaire d'événements est invoqué lorsque l'utilisateur change le texte affiché puis transfère le focus du clavier hors de l'élément.
7. Gestionnaires d'Événements onblur Invoqué lorsque l'élément perd le focus du clavier.

Chapitre 24 | Phrases des pages 229-233

1. L'objet Layer n'est pris en charge que dans Netscape 4 et a été abandonné dans Netscape 6.
2. Tout élément HTML avec un attribut de position CSS

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

absolue est représenté par un objet Layer en JavaScript.

3. Lorsque deux couches se chevauchent, celle avec le zIndex le plus élevé apparaît en haut et obscurcit celle avec le zIndex le plus faible.
4. Définir cette propriété définit également la valeur de clip.bottom.
5. Déplacer une couche au-dessus d'une autre et ne rien renvoie.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

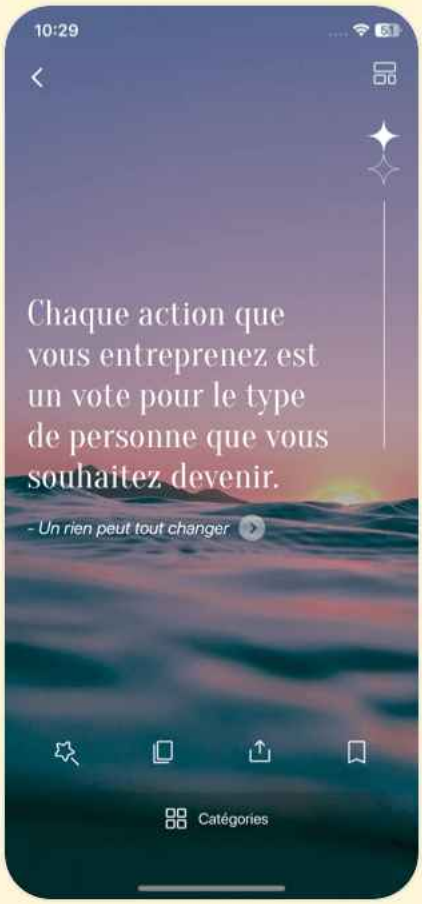
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 25 | Phrases des pages 234-236

1. Une propriété de chaîne en lecture/écriture qui spécifie la partie d’ancre de l'URL du lien, y compris le signe dièse (#) initial.
2. Par exemple : "www.oreilly.com:1234".
3. Les noms spéciaux "_blank", "_top", "_parent" et "_self" sont autorisés.
4. Dans JavaScript, ce gestionnaire d'événements peut empêcher le lien d'être suivi en retournant faux.
5. La propriété status de la fenêtre actuelle peut être définie ici.

Chapitre 26 | Phrases des pages 239-242

1. L'objet Math est un moyen de regrouper les constantes et les fonctions mathématiques.
2. Math.abs(x) Renvoie la valeur absolue de x.
3. Math.max(args...) Renvoie la plus grande des valeurs arguments.
4. Math.random( ) Renvoie un nombre pseudo-aléatoire entre 0.0 et 1.0.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

5.Math.sqrt(x) Renvoie la racine carrée de x.

Chapitre 27 | Phrases des pages 243-246

1. appCodeName Une chaîne en lecture seule qui spécifie un surnom pour le navigateur. Dans tous les navigateurs Netscape, c'est 'Mozilla'.
2. cookieEnabled Un booléen en lecture seule qui est vrai si le navigateur a les cookies activés, et faux s'ils sont désactivés.
3. userAgent Une chaîne en lecture seule qui spécifie la valeur utilisée par le navigateur pour l'en-tête user-agent dans les requêtes HTTP.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

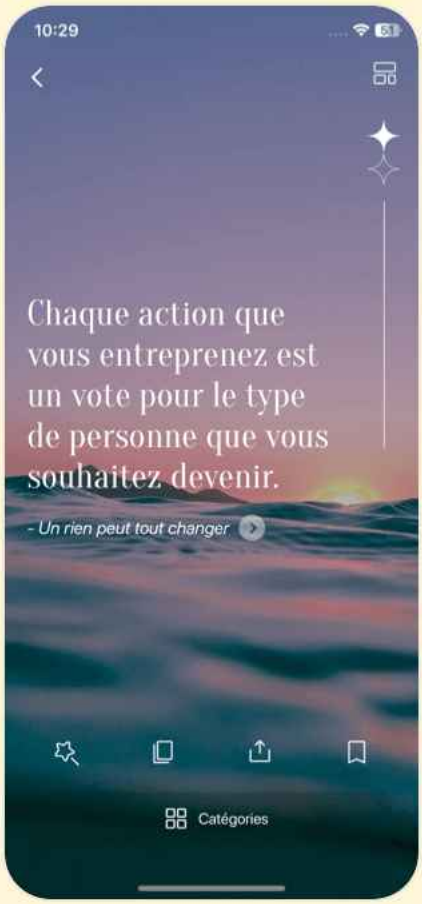
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 28 | Phrases des pages 247-252

1. Tous les nœuds d'un document HTML sont des instances de l'une des sous-classes de Node énumérées ci-dessus.
2. Chaque objet Node a une propriété nodeType qui spécifie de quelle sous-classe il est une instance.
3. appendChild(newChild) Ajoute le nœud newChild à l'arbre de document en l'attachant au tableau childNodes[] de ce nœud.
4. removeChild(oldChild) Supprime le nœud oldChild de l'arbre de document.
5. normalize() Normalise tous les nœuds de texte descendants de ce nœud en supprimant les nœuds de texte vides et en fusionnant les nœuds de texte adjacents.

Chapitre 29 | Phrases des pages 253-255

1. Number.MAX_VALUE La plus grande valeur représentable. Environ 1,79E+308.
2. Number.MIN_VALUE La plus petite valeur représentable. Environ 5E-324.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

3.Number.NaN Valeur non numérique. Identique à la valeur NaN globale.

4.Number.NEGATIVE_INFINITY Valeur infinie négative.

5.Number.POSITIVE_INFINITY Valeur infinie. Identique à l'infini global.

Chapitre 30 | Phrases des pages 256-258

1. Ce constructeur crée un objet vide auquel vous pouvez ajouter des propriétés arbitraires.
2. Retourne vrai si l'objet a une propriété non héritée avec le nom spécifié.
3. Retourne vrai si cet objet est le prototype de o.
4. Retourne vrai si cet objet a une propriété énumérable non héritée avec le nom spécifié.
5. Retourne une représentation sous forme de chaîne localisée de l'objet.
6. L'implémentation de cette méthode fournie par la classe Object est assez générique et n'apporte pas beaucoup d'informations utiles.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

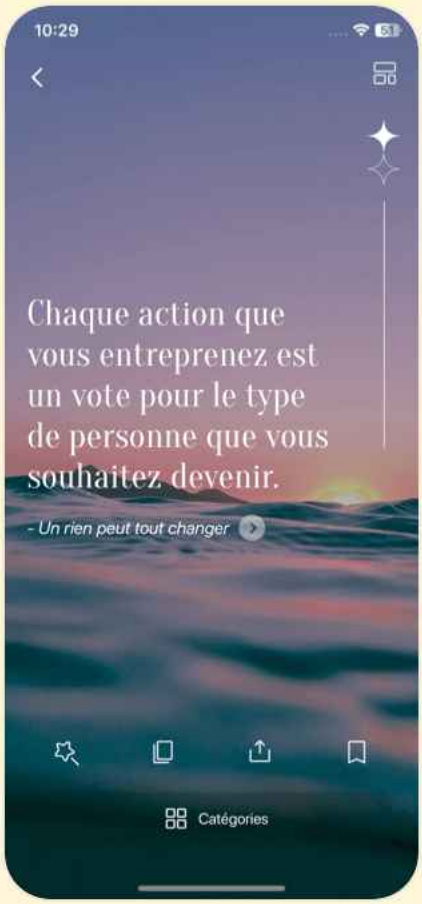
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 31 | Phrases des pages 261-263

1. /pattern/attributes
2. new RegExp(pattern, attributes)
3. global A read-only boolean that specifies whether the RegExp has the g attribute and therefore performs global matching.
4. ignoreCase A read-only boolean that specifies whether the RegExp has the i attribute and therefore performs case-insensitive matching.
5. lastIndex For global RegExp objects, this read/write property specifies the character position immediately following the last match; this is the first character examined for the next match.
6. exec(string) Matches string against this RegExp and returns an array containing the results of the match, or null if no match was found.

Chapitre 32 | Phrases des pages 266-269

1. L'objet Select définit les propriétés pour chacun des attributs de la balise HTML <select>, comme

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

disabled, multiple, name et size.

2. Vous pouvez supprimer des options individuelles en définissant un élément du tableau sur null — cela déplace les éléments au-dessus vers le bas, raccourcissant ainsi le tableau.
3. Définir cette propriété entraîne la désélection de toutes les autres options. La définir sur -1 provoque également la désélection de toutes les options.
4. Si l'objet Select n'autorise qu'une seule sélection (c'est-à-dire que l'attribut multiple n'apparaît pas dans la définition HTML de l'objet), cette propriété est 'select-one'.

Chapitre 33 | Phrases des pages 270-274

1. Sans le mot-clé new, la fonction String( ) convertit son argument en chaîne de caractères.
2. Retourne une nouvelle chaîne résultant de la conversion de chacun des arguments en chaîne et de la concaténation des chaînes résultantes.
3. Retourne la position de la première occurrence de la sous-chaîne dans cette chaîne qui apparaît à partir de la

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

position de départ ou -1 si aucune occurrence n'est trouvée.

4. Retourne une nouvelle chaîne, avec le texte correspondant à l'expression régulière remplacé par une valeur de remplacement.
5. Retourne un tableau de chaînes, créé en divisant la chaîne en sous-chaînes aux limites spécifiées par le délimiteur.
6. Retourne une copie de la chaîne, avec toutes les lettres majuscules converties en leur équivalent minuscule.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

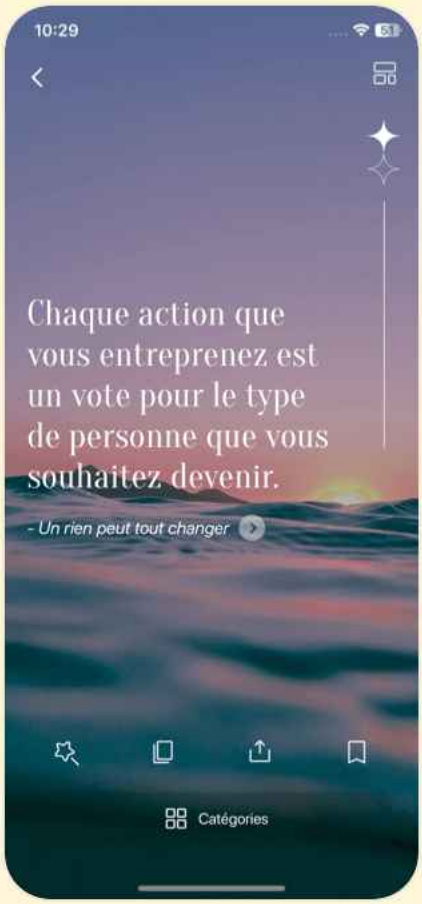
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



Chapitre 34 | Phrases des pages 275-278

1. L'objet Style définit un grand nombre de propriétés : une propriété pour chaque attribut CSS défini par la spécification CSS2.
2. Les attributs multi-mots qui contiennent des tirets, tels que font-family, s'écrivent sans tirets en JavaScript, et chaque mot après le premier est mis en majuscule : fontFamily.
3. Notez que les navigateurs actuels n'implémentent pas toutes ces propriétés.
4. Toutes les propriétés sont des chaînes de caractères, et il faut faire attention lors de l'utilisation de propriétés ayant des valeurs numériques.

Chapitre 35 | Phrases des pages 285-366

1. L'objet Window est l'objet global pour JavaScript côté client.
2. JavaScript est un langage de script léger et orienté objet qui peut être intégré dans des pages HTML.
3. Les expressions JavaScript sont formées en combinant des valeurs à l'aide des opérateurs JavaScript.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

4. JavaScript prend en charge trois types de données primitifs : les nombres, les booléens et les chaînes de caractères ; et deux types de données composés : les objets et les tableaux.
5. Les objets sont créés avec l'opérateur new.
6. L'instruction for/in parcourt les propriétés d'un objet spécifié.
7. JavaScript est un langage sensible à la casse.
8. JavaScript n'a pas de portée de bloc.
9. Le mot clé this fait référence à l'objet pour lequel la fonction est une propriété.
10. La gestion des erreurs en JavaScript se fait via l'instruction try/catch/finally.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

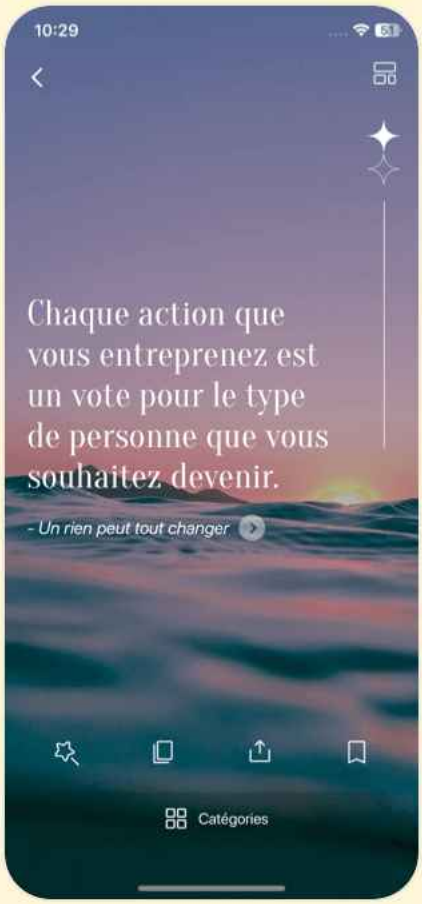
Téléchargez l'appli Bookey pour profiter

Plus d'un million de citations
Plus de 1000 résumés de livres

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play



JavaScript Questions

Voir sur le site de Bookey

Chapitre 1 | Section 1.1. Syntaxe| Questions et réponses

1.Question

Quelle est l'importance de la sensibilité à la casse en JavaScript ?

Réponse:La sensibilité à la casse en JavaScript signifie que le langage considère les identifiants avec des capitalisations différentes comme distincts. Par exemple, 'variable', 'Variable' et 'VARIABLE' seraient tous considérés comme des entités différentes. Cela est important pour éviter les conflits de noms et améliorer la clarté du code, car cela encourage les développeurs à adhérer constamment à leurs conventions de nommage.

2.Question

Comment les espaces blancs affectent-ils le code JavaScript ?

Réponse:Les espaces blancs en JavaScript sont ignorés entre

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

les tokens, ce qui donne aux programmeurs la liberté de formater leur code de manière lisible et organisée. Cette flexibilité permet une meilleure structuration du code, facilitant non seulement la compréhension personnelle mais aussi la collaboration avec les autres, car le code devient plus facile à lire et à maintenir.

3.Question

Pourquoi les points-virgules sont-ils importants et quand peuvent-ils être omis en JavaScript ?

Réponse:Les points-virgules sont utilisés pour terminer les instructions en JavaScript, ce qui aide l'interpréteur à comprendre où une instruction se termine et où une autre commence. Ils peuvent être omis lorsque l'instruction se termine par un retour à la ligne ; cependant, cela pourrait entraîner des erreurs si la première ligne peut se suffire à elle-même comme une instruction valide. Cette fonctionnalité souligne la nécessité d'un formatage soigné pour éviter des conséquences inattendues lors de l'exécution du code.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

4.Question

Quel est le but des commentaires en JavaScript ?

Réponse:Les commentaires en JavaScript sont cruciaux pour la documentation et l'amélioration de la lisibilité du code. Ils permettent aux développeurs de laisser des notes, des explications ou des rappels dans le code, ce qui peut aider d'autres programmeurs (ou eux-mêmes plus tard) à comprendre la logique et le but derrière certaines sections de code. Par exemple, un commentaire pourrait expliquer la raison d'un choix d'algorithme particulier ou noter une fonctionnalité prévue pour de futures mises à jour.

5.Question

Que doivent garder à l'esprit les développeurs lorsqu'ils nomment des identifiants en JavaScript ?

Réponse:Les développeurs doivent suivre des règles spécifiques lors du nommage des identifiants en JavaScript. Les identifiants peuvent inclure des lettres, des chiffres, des soulignés et des signes dollar, mais ne doivent pas commencer par un chiffre. De plus, ils devraient éviter les

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

mots réservés et ne pas utiliser les mêmes noms que les propriétés et méthodes globales pour éviter les conflits et assurer la clarté du code.

6.Question

Quels sont les mots réservés en JavaScript et pourquoi devraient-ils être évités comme identifiants ?

Réponse:Les mots réservés en JavaScript ont des significations et des fonctions prédéfinies au sein du langage, tels que 'if', 'for' et 'return'. Ceux-ci ne peuvent pas être utilisés comme identifiants car cela peut entraîner des complications dans le code et une confusion concernant la structure du langage. Il est essentiel de respecter ces mots réservés pour maintenir la logique et le déroulement des scripts.

7.Question

Comment un style de programmation peut-il affecter la qualité du code JavaScript ?

Réponse:Un style de programmation cohérent, incluant une utilisation adéquate de la casse pour les mots-clés et les noms

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

de fonctions, l'utilisation appropriée des espaces blancs et des commentaires réfléchis, a un impact considérable sur la maintenabilité et la lisibilité du code JavaScript. Un code bien stylé est plus facile à naviguer, à comprendre et à déboguer, surtout dans des environnements d'équipe.

Chapitre 2 | Section 1.3. Types de Données| Questions et réponses

1.Question

Quels sont les principaux types de données primitifs supportés par JavaScript ?

Réponse:JavaScript prend en charge trois types de données primitifs : les nombres, les booléens et les chaînes.

2.Question

Comment les nombres sont-ils représentés en JavaScript ?

Réponse:Les nombres sont représentés en JavaScript en utilisant un format à virgule flottante de 64 bits, sans distinction entre entiers et nombres à virgule flottante.

3.Question

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Quels valeurs spéciales une opération numérique peut-elle renvoyer en cas d'erreur ?

Réponse:Lorsque une opération numérique déborde, elle renvoie l'infini positif ou négatif ; lorsqu'elle est sous-dimensionnée, elle renvoie zéro ; et si une opération entraîne une erreur (comme prendre la racine carrée d'un nombre négatif), elle renvoie NaN (not a number).

4.Question

Peux-tu donner un exemple de comment définir une chaîne en JavaScript ?

Réponse:Une chaîne peut être définie en utilisant des guillemets simples ou doubles, par exemple : 'Bonjour le monde' ou "JavaScript est amusant !".

5.Question

Que signifie le fait que les chaînes en JavaScript soient immuables ?

Réponse:Cela signifie qu'une fois qu'une chaîne est créée, son contenu ne peut pas être modifié. Toute opération semblant modifier la chaîne renverra en réalité une nouvelle

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

chaîne modifiée.

6.Question

Comment créer un nouvel objet en JavaScript ?

Réponse:Vous pouvez créer un nouvel objet en utilisant l'opérateur new, comme ceci : var o = new Object(); ou en utilisant la syntaxe littérale d'objet : var o = {x: 1, y: 2};.

7.Question

Qu'est-ce qu'un tableau en JavaScript ?

Réponse:Un tableau est un type spécialisé d'objet qui contient des valeurs numérotées (éléments) et peut contenir n'importe quel type de valeur, y compris d'autres objets et tableaux.

8.Question

Quel est le but des fonctions en JavaScript ?

Réponse:Les fonctions sont des blocs de code réutilisables qui peuvent être définis une fois et exécutés plusieurs fois, ce qui aide à organiser et encapsuler la logique du code.

9.Question

Peux-tu expliquer la différence entre null et undefined en JavaScript ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:Null est une valeur spéciale qui indique explicitement 'pas de valeur', tandis qu'undefined est un type de variable qui a été déclaré mais pas encore attribué de valeur.

10.Question

Quelle est la signification du mot-clé 'this' dans une méthode en JavaScript ?

Réponse:Dans le contexte d'une méthode, le mot-clé 'this' fait référence à l'objet pour lequel la méthode est une propriété, permettant d'accéder aux propriétés de cet objet.

Chapitre 3 | Section 1.4. Expressions et Opérateurs| Questions et réponses

1.Question

Qu'est-ce que les expressions JavaScript et comment sont-elles construites ?

Réponse:Les expressions JavaScript sont créées en combinant des valeurs à l'aide d'opérateurs, qui peuvent inclure des littéraux, des variables, des propriétés d'objet, des éléments de tableau et des invocations de fonction. Des parenthèses peuvent

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

être utilisées pour regrouper des sous-expressions et ajuster l'ordre d'évaluation par défaut.

2.Question

Quel est le rôle des opérateurs en JavaScript, et pouvez-vous donner des exemples de leurs catégories ?

Réponse:Les opérateurs en JavaScript effectuent une variété d'opérations telles que des opérations arithmétiques, de comparaison et de manipulation d'objet. Des exemples de catégories incluent les opérateurs arithmétiques (comme +, -, *, /), les opérateurs de comparaison (comme ==, ===, <, >) et les opérateurs logiques (comme &&, ||).

3.Question

En quoi l'opérateur d'égalité '==' de JavaScript diffère-t-il de l'opérateur d'identité '===' ?

Réponse:L'opérateur d'égalité '==' effectue une comparaison lâche, permettant la conversion de types, donc 3 == '3' renvoie true. En revanche, l'opérateur d'identité '===' vérifie à la fois la valeur et le type, ce qui signifie que 3 === '3' renvoie false car ils sont de types différents.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

4.Question

Pourquoi l'opérateur 'typeof' est-il important en JavaScript et que retourne-t-il ?

Réponse:L'opérateur 'typeof' est crucial pour déterminer le type de données d'un opérande donné, retournant des chaînes comme 'number', 'string', 'boolean', 'object', entre autres. Cela aide à réaliser une vérification de type appropriée et à gérer les erreurs dans les scripts.

5.Question

Que fait l'opérateur 'delete' en JavaScript ?

Réponse:L'opérateur 'delete' supprime une propriété d'un objet. Contrairement à l'attribution de la valeur null à la propriété, il supprime réellement la propriété elle-même de l'objet, et renverra false si la propriété n'était pas configurables ou ne pouvait pas être supprimée.

6.Question

Comment les opérateurs relationnels fonctionnent-ils sur les chaînes en JavaScript ?

Réponse:Les opérateurs relationnels tels que <, <=, > et >= comparent les chaînes en fonction de leur ordre alphabétique,

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

ce qui signifie que 'banana' < 'apple' évaluerait à false car 'b' vient après 'a' dans la séquence alphabétique.

7.Question

Pouvez-vous expliquer comment fonctionnent les opérateurs d'assignation en JavaScript ?

Réponse:Les opérateurs d'assignation en JavaScript assignent des valeurs à des variables et peuvent également effectuer des opérations. Par exemple, 'x += 2' ajoute 2 à x, tandis que 'y *= 3' multiplierait y par 3. Ils constituent une écriture abrégée pour effectuer une opération suivie d'une assignation.

8.Question

Quel est un exemple d'utilisation de parenthèses dans une expression JavaScript ?

Réponse:Considérons l'expression (1 + 2) * 3. Ici, des parenthèses sont utilisées pour s'assurer que l'addition est effectuée en premier, résultant en 3 * 3, ce qui équivaut à 9. Sans les parenthèses, les opérations suivraient leur ordre de priorité par défaut, menant à un résultat différent.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

9.Question

Comment fonctionne l'opérateur 'instanceof' en JavaScript ?

Réponse:L'opérateur 'instanceof' vérifie si un objet a été créé à partir d'une fonction constructeur spécifique. Par exemple, si obj est créé par le constructeur Date, alors obj instanceof Date évalue à true. Cela est utile pour vérifier le type des objets.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Ad

Scanner pour télécharger

Lire, Partager, Autonomiser

Terminez votre défi de lecture, faites don de livres aux enfants africains.

Le Concept

BOOKS FOR AFRICA × ×

Cette activité de don de livres se déroule en partenariat avec Books For Africa. Nous lançons ce projet car nous partageons la même conviction que BFA : Pour de nombreux enfants en Afrique, le don de livres est véritablement un don d'espoir.

La Règle

Gagnez 100 points → **Échangez un livre** → **Faites un don à l'Afrique**

Votre apprentissage ne vous apporte pas seulement des connaissances mais vous permet également de gagner des points pour des causes caritatives ! Pour chaque 100 points gagnés, un livre sera donné à l'Afrique.

Essai gratuit avec Bookey

Chapitre 4 | Section 1.5. Instructions| Questions et réponses

1.Question

De quoi est composé un programme JavaScript ?

Réponse:Un programme JavaScript est une séquence d'instructions JavaScript.

2.Question

Comment fonctionne une instruction d'expression en JavaScript ?

Réponse:Chaque expression JavaScript peut fonctionner comme une instruction autonome, telles que des affectations ou des appels de méthode, comme 's = "hello world";' ou 'x = Math.sqrt(4);'.

3.Question

Qu'est-ce qu'une instruction composée et quand est-elle utilisée ?

Réponse:Une instruction composée est constituée de plusieurs instructions enveloppées dans des accolades. Elle est utile lorsque une boucle ou une condition doit exécuter plus d'une instruction.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

4.Question

Quel est le but de l'instruction vide en JavaScript ?

Réponse:L'instruction vide, représentée par un seul point-virgule, ne fait rien, mais peut être utilisée pour définir des corps de boucle vides.

5.Question

Qu'est-ce que des instructions étiquetées et comment sont-elles utilisées ?

Réponse:Les instructions étiquetées vous permettent d'étiqueter n'importe quelle instruction, facilitant l'utilisation de 'break' ou 'continue' pour contrôler le flux dans des boucles imbriquées.

6.Question

Pouvez-vous expliquer la différence entre les instructions 'break' et 'continue' ?

Réponse:L'instruction 'break' sort de la boucle la plus intérieure, tandis que 'continue' passe à l'itération suivante, en sautant l'itération actuelle.

7.Question

Quelle est l'importance de la boucle 'do/while' ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:La boucle 'do/while' garantit que le corps s'exécute au moins une fois, car la condition est vérifiée après le corps de la boucle.

8.Question

Comment fonctionne une boucle 'for' en JavaScript ?

Réponse:Une boucle 'for' combine l'initialisation, la condition de test et la mise à jour en une seule ligne, exécutant l'instruction de manière répétée tant que la condition de test reste vraie.

9.Question

Que fait l'instruction 'for/in' ?

Réponse:La boucle 'for/in' permet d'itérer sur les propriétés d'un objet, assignant le nom de chaque propriété à une variable spécifiée.

10.Question

Quel est le rôle de l'instruction 'function' ?

Réponse:L'instruction 'function' définit une fonction nommée avec des paramètres spécifiques et un corps de code exécutable.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

11.Question

Décrivez la fonctionnalité de l'instruction 'if/else' .

Réponse:L'instruction 'if' exécute un bloc de code si la condition est vraie ; sinon, elle peut exécuter un bloc de code alternatif en utilisant 'else'.

12.Question

En quoi l'instruction 'switch' diffère-t-elle des conditionnelles standard ?

Réponse:L'instruction 'switch' évalue une expression et saute au cas qui correspond à l'expression, facilitant le branchement multiple.

13.Question

Quel est le but de la gestion d'exceptions en JavaScript ?

Réponse:La construction 'try/catch/finally' vous permet de gérer élégamment les erreurs en exécutant du code lorsque des exceptions se produisent et en veillant à ce que les actions de nettoyage soient exécutées ensuite.

14.Question

Quelle est l'importance de l'instruction 'var' ?

Réponse:L'instruction 'var' est utilisée pour déclarer des

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

variables en JavaScript, ce qui est particulièrement important pour déclarer des variables locales au sein des fonctions.

15.Question

Pourquoi l'instruction 'with' est-elle déconseillée ?

Réponse:L'instruction 'with' peut conduire à des effets secondaires complexes et non intuitifs, rendant le code moins lisible et prévisible.

Chapitre 5 | Section 1.6. JavaScript orienté objet| Questions et réponses

1.Question

Quel est le rôle d'une fonction constructeur en JavaScript ?

Réponse:Une fonction constructeur est utilisée pour créer de nouveaux objets et initialiser leurs propriétés. Elle est invoquée avec l'opérateur 'new' et utilise 'this' pour faire référence au nouvel objet créé.

2.Question

Comment fonctionne l'héritage dans les objets JavaScript ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:JavaScript utilise un mécanisme d'héritage simple où les objets héritent des propriétés et des méthodes d'un objet prototype associé à leur constructeur. Toutes les propriétés ou méthodes définies sur le prototype sont accessibles à toutes les instances de l'objet.

3.Question

Pouvez-vous expliquer l'importance de la propriété prototype dans les constructeurs ?

Réponse:La propriété prototype permet de définir des méthodes partagées entre toutes les instances d'une classe, réduisant ainsi l'utilisation de la mémoire et rendant le code plus efficace. Par exemple, en définissant 'distanceTo' sur Point.prototype, toutes les instances de Point peuvent utiliser cette méthode sans avoir chacune leur propre copie.

4.Question

Qu'est-ce qu'une méthode statique en JavaScript, et comment est-elle définie ?

Réponse:Une méthode statique est définie directement sur une fonction constructeur, plutôt que sur son prototype. Cela

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

signifie qu'elle est associée à la classe elle-même plutôt qu'à une instance de cette classe. Par exemple, 'Point.ORIGIN' est une propriété statique définie sur le constructeur Point.

5.Question

Comment pouvez-vous expliquer la valeur de retour de la méthode toString dans la classe Point ?

Réponse:La méthode toString est définie pour convertir un objet Point en une chaîne de caractères lisible. Elle retourne une représentation sous forme de chaîne des coordonnées de l'objet, facilitant ainsi son inclusion dans des messages ou des journaux, comme '(3,4)' pour un Point avec x=3 et y=4.

6.Question

Donnez un exemple de comment créer un nouvel objet Point et utiliser ses méthodes.

Réponse:Vous pouvez créer un nouvel objet Point avec 'var p = new Point(3, 4);'. Ensuite, vous pouvez calculer la distance entre ce point et l'origine avec 'var d = p.distanceTo(Point.ORIGIN);'. Cela vous permet d'utiliser dynamiquement les méthodes de l'objet dans des scénarios en

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

temps réel.

7.Question

Pourquoi la programmation orientée objet est-elle bénéfique en JavaScript ?

Réponse:La programmation orientée objet (POO) en JavaScript permet une meilleure organisation du code, favorise la réutilisation grâce à l'héritage et modélise plus efficacement des entités du monde réel. Cela conduit à un développement logiciel plus gérable et évolutif.

Chapitre 6 | Section 1.7. Expressions Régulières| Questions et réponses

1.Question

Qu'est-ce que les expressions régulières et comment sont-elles utilisées en JavaScript ?

Réponse:Les expressions régulières sont des séquences de caractères qui définissent des motifs de recherche, principalement utilisés pour la correspondance de chaînes en JavaScript. Elles utilisent la même syntaxe que Perl, permettant aux développeurs d'effectuer des opérations de chaîne

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

complexes de manière efficace. En JavaScript, elles peuvent être écrites en utilisant la notation littérale (enveloppées dans des barres obliques) ou avec le constructeur RegExp.

2.Question

Pouvez-vous expliquer ce que sont les classes de caractères dans le contexte des expressions régulières ?

Réponse:Les classes de caractères dans les expressions régulières permettent de définir un ensemble de caractères à correspondre. Par exemple, [abc] correspond à n'importe quel caractère unique 'a', 'b' ou 'c'. Vous pouvez également nier une classe avec [^abc], qui correspond à tout caractère qui n'est pas 'a', 'b' ou 'c'. Cette fonctionnalité améliore considérablement la flexibilité de la correspondance de motifs.

3.Question

Quelle est l'importance de la répétition dans les expressions régulières ?

Réponse:La répétition vous permet de définir combien de

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

fois un motif doit apparaître dans le texte. Par exemple, le caractère '*' indique que l'élément précédent peut apparaître zéro ou plusieurs fois, tandis que '+' indique une ou plusieurs fois. Cela permet des capacités de correspondance puissantes, telles que la validation des saisies utilisateur ou l'analyse de données structurées.

4.Question

Que signifie le regroupement dans les expressions régulières et pourquoi est-il important ?

Réponse:Le regroupement dans les expressions régulières se fait en utilisant des parenthèses pour combiner plusieurs éléments en une seule unité. C'est important car cela vous permet d'appliquer des quantificateurs à des phrases entières plutôt qu'à des caractères individuels. Par exemple, (abc)+ correspondrait à 'abc', 'abcabc', etc., traitant l'ensemble du groupe comme une seule entité.

5.Question

Comment les ancres affectent-elles la correspondance de motifs dans les expressions régulières ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:Les ancres restreignent la correspondance de motifs à un emplacement spécifique dans une chaîne. Par exemple, '^' signale le début et '$' la fin, donc /^Hello/ ne correspondrait à 'Hello' que s'il se trouve au début d'une chaîne. Cela est crucial pour s'assurer que les motifs ne correspondent qu'à des emplacements prescrits, améliorant ainsi la précision des opérations de recherche.

6.Question

Qu'est-ce que les assertions de look-ahead et comment peuvent-elles être utiles dans les expressions régulières ?

Réponse:Les assertions de look-ahead sont des fonctionnalités avancées qui vous permettent de correspondre à un motif uniquement si certaines conditions sont remplies après le motif, sans inclure ces conditions dans la correspondance elle-même. Par exemple, '(?=abc)' correspondrait à n'importe quoi si 'abc' le suit, mais ne consommerait pas 'abc' comme partie de la correspondance. Cela est utile dans les scénarios où la présence ou l'absence de certains motifs affecte ce que vous essayez de

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

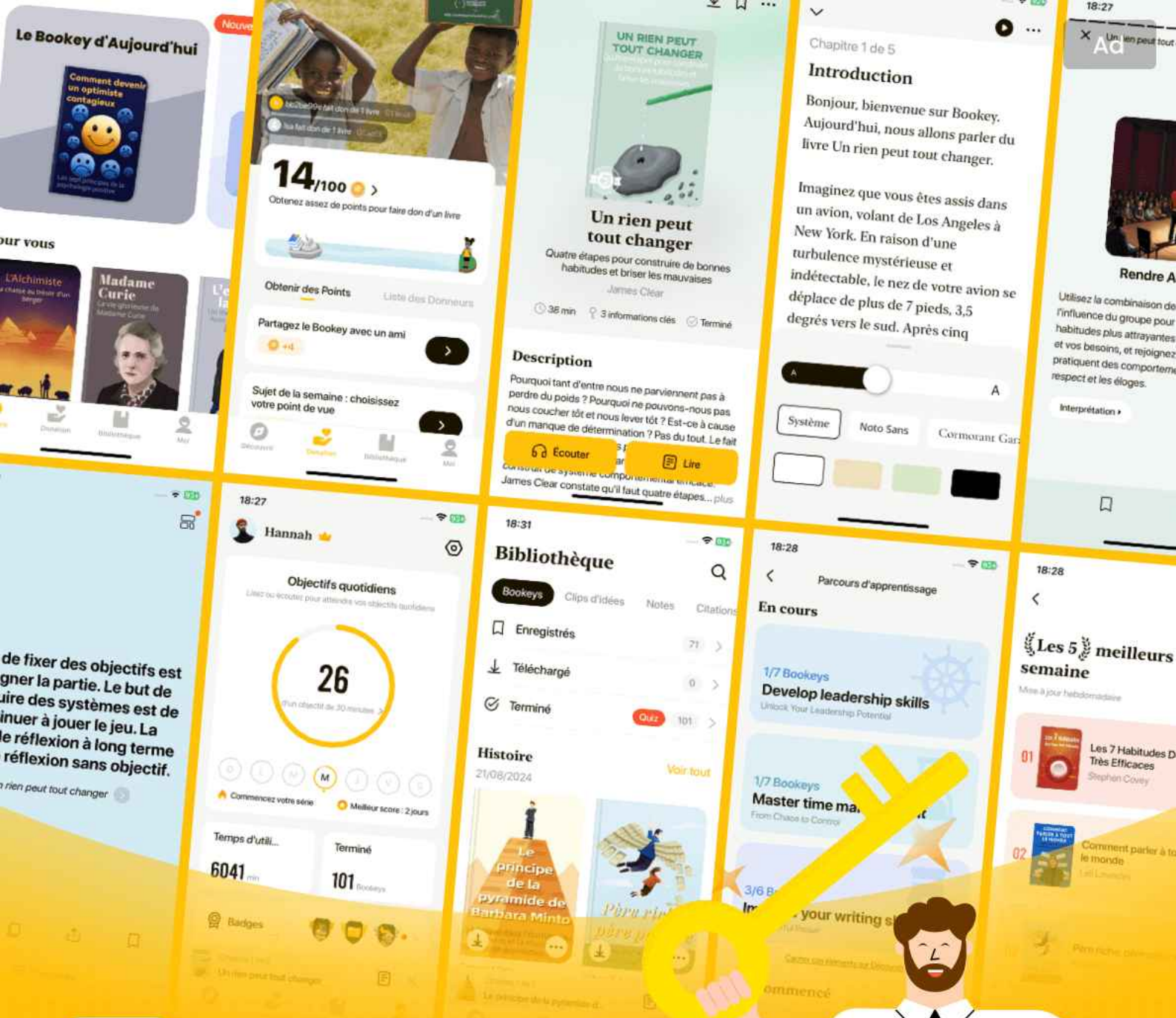
correspondre.

7.Question

Quels défis les développeurs pourraient-ils rencontrer en utilisant des expressions régulières en JavaScript ?

Réponse:Les développeurs peuvent rencontrer des difficultés en raison de la complexité et de la syntaxe nuancée des expressions régulières. Par exemple, créer une expression régulière qui interprète correctement divers motifs ou échapper à des caractères spéciaux nécessite une attention minutieuse aux détails. De plus, des expressions trop complexes peuvent entraîner des problèmes de performance ou des correspondances non intentionnelles, rendant le test et le débogage essentiels.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé



Les meilleures idées du monde débloquent votre potentiel

Essai gratuit avec Bookey

Télécharger dans App Store

DISPONIBLE SUR Google Play

Scanner pour télécharger

Chapitre 7 | Section 1.8. Versions de JavaScript| Questions et réponses

1.Question

Quelle est l'importance des versions de JavaScript et comment se rapportent-elles au développement web ?

Réponse:Les versions de JavaScript sont cruciales pour comprendre l'évolution du langage et ses capacités. En partant de JavaScript 1.0, qui était bogué et limité, jusqu'à JavaScript 1.5 introduisant la gestion des exceptions conforme à ECMA v3, chaque version a ajouté des fonctionnalités essentielles qui ont amélioré la robustesse et la fonctionnalité. Cette évolution est vitale pour les développeurs web car elle les informe sur les fonctionnalités disponibles, garantit la compatibilité entre les différents navigateurs et guide le choix des versions de JavaScript pour divers projets.

2.Question

Comment JavaScript s'intègre-t-il dans HTML et pourquoi cette intégration est-elle importante ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:JavaScript s'intègre dans HTML grâce à la balise <script>, aux gestionnaires d'événements et aux URL JavaScript. Cette intégration permet aux développeurs de créer des pages web interactives et dynamiques, améliorant ainsi l'expérience utilisateur. Par exemple, l'utilisation de <script> permet de générer du contenu dynamique, tandis que les gestionnaires d'événements permettent des interactions utilisateur en temps réel comme des clics et des soumissions de formulaires. Cette intégration fluide est essentielle pour le développement web moderne car elle transforme un HTML statique en applications dynamiques.

3.Question

Expliquez le rôle de l'objet Window dans JavaScript côté client.

Réponse:L'objet Window représente la fenêtre du navigateur dans JavaScript côté client, servant d'objet global pour accéder et manipuler des documents web. Il définit des propriétés et des méthodes globales, telles que document, qui pointe vers l'objet Document représentant le contenu affiché.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Comprendre l'objet Window permet aux développeurs de contrôler des aspects du navigateur, comme la navigation et les événements de synchronisation, améliorant ainsi l'interactivité et la gestion de l'expérience utilisateur.

4.Question

Quelles sont les fonctionnalités clés fournies par l'objet Document ?

Réponse:L'objet Document est essentiel car il représente le document HTML affiché dans le navigateur. Il permet d'accéder aux éléments du document à travers des propriétés telles que forms, images et liens. Les développeurs peuvent manipuler le contenu, gérer des événements et générer dynamiquement du HTML grâce à des méthodes telles que getElementById() et write(). Maîtriser l'objet Document facilite la création d'applications web puissantes qui réagissent aux entrées utilisateur et s'adaptent dynamiquement.

5.Question

Quels avantages le DOM W3C et le DOM IE 4 offrent-ils aux développeurs JavaScript ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:Le DOM W3C fournit une méthode standardisée pour accéder et modifier tous les éléments de document, favorisant la compatibilité entre les navigateurs modernes. Il permet aux développeurs de parcourir l'arbre du document et de manipuler divers structures HTML de manière systématique. En revanche, le DOM IE 4, bien que non standard, offre des caractéristiques uniques telles que innerHTML pour des mises à jour de contenu rapides. Comprendre les deux permet aux développeurs d'écrire des scripts compatibles et d'utiliser efficacement les fonctionnalités spécifiques aux navigateurs.

6.Question

Comment JavaScript peut-il améliorer l'interaction utilisateur grâce aux événements, et pourquoi cette fonctionnalité est-elle cruciale ?

Réponse:JavaScript améliore l'interaction utilisateur grâce à la gestion des événements, permettant aux scripts de répondre aux actions des utilisateurs comme les clics, les pressions de touches et les mouvements de souris. Cette

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

interactivité est cruciale pour les applications web modernes qui nécessitent un retour d'information en temps réel et des changements de contenu dynamiques, améliorant l'engagement et la satisfaction des utilisateurs. Par exemple, la validation de formulaires utilisant les événements onsubmit garantit l'intégrité des données en empêchant les soumissions incorrectes avant qu'elles ne se produisent.

7.Question

Quelles restrictions de sécurité sont imposées sur JavaScript côté client et pourquoi sont-elles importantes ?

Réponse:Les restrictions de sécurité sur JavaScript côté client, telles que la politique de même origine et les limitations d'accès aux fichiers, sont vitales pour protéger la vie privée des utilisateurs et prévenir les activités malveillantes. Ces restrictions garantissent que les scripts ne peuvent interagir qu'avec les documents servis depuis le même domaine, atténuant ainsi les menaces telles que le cross-site scripting (XSS). Comprendre ces limitations permet aux développeurs d'écrire des applications sécurisées

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

qui protègent les données des utilisateurs et maintiennent la confiance.

Chapitre 8 | Section 2.1. JavaScript dans HTML| Questions et réponses

1.Question

Quel est le principal objectif de la balise <script> en HTML ?

Réponse:La balise <script> est utilisée pour intégrer du code JavaScript dans des documents HTML, vous permettant d'exécuter JavaScript soit directement, soit en liant un fichier externe.

2.Question

Comment les gestionnaires d'événements peuvent-ils améliorer l'interaction utilisateur dans une application web ?

Réponse:Les gestionnaires d'événements permettent d'exécuter du code JavaScript en réponse à des actions spécifiques de l'utilisateur, comme des clics ou des événements clavier. Par exemple, utiliser un événement onclick pour afficher 'Hello World!' lorsqu'un bouton est

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

cliqué montre comment la programmation basée sur des événements peut rendre les pages web interactives.

3.Question

Quelle est l'importance de l'attribut type dans la balise <script> ?

Réponse:L'attribut type spécifie le langage de script du code inclus. Pour JavaScript, il doit être défini sur 'text/javascript' afin d'assurer la compatibilité, notamment dans des environnements où plusieurs langages de script peuvent être utilisés.

4.Question

Comment le schéma d'URL javascript: fonctionne-t-il dans les formulaires HTML ?

Réponse:Le schéma d'URL javascript: permet d'exécuter du code JavaScript lors de la soumission d'un formulaire sans recharger la page. En utilisant ce schéma avec l'opérateur void, vous pouvez exécuter des fonctions sans modifier le contenu du document.

5.Question

Pourquoi pourriez-vous avoir besoin de spécifier la

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

version de JavaScript dans la balise <script> ?

Réponse:Spécifier la version de JavaScript à l'aide d'attributs comme language ou type garantit que les navigateurs interprètent correctement le code. Cela aide à maintenir la compatibilité avec différentes versions de navigateurs qui ne peuvent prendre en charge que des fonctionnalités spécifiques de JavaScript.

6.Question

Pouvez-vous donner un exemple d'utilisation d'un gestionnaire d'événements pour améliorer la fonctionnalité d'une page web ?

Réponse:Bien sûr ! Par exemple, une zone de saisie pourrait utiliser le gestionnaire d'événements onfocus pour changer sa couleur de fond, indiquant aux utilisateurs qu'elle est prête à recevoir des saisies. Ce retour visuel améliore l'expérience utilisateur en rendant l'interaction plus fluide.

7.Question

Quel est le rôle de l'opérateur void dans les URL JavaScript ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:L'opérateur void est utilisé dans les URL JavaScript pour empêcher le navigateur de naviguer vers une nouvelle page ou de modifier le document actuel. Il permet d'exécuter des scripts sans effet visible sur la page web, ce qui est utile dans les actions de formulaire.

8.Question

Comment la balise <script> prend-elle en charge différents langages de script ?

Réponse:La balise <script> permet l'inclusion de différents langages de script en spécifiant le type MIME ou l'attribut de langue approprié. Bien que JavaScript soit le plus courant, HTML prend en charge d'autres scripts comme VBScript, ce qui le rend polyvalent pour les développeurs qui peuvent travailler dans des environnements multilingues.

Chapitre 9 | Section 2.2. L'objet Window| Questions et réponses

1.Question

Quelle est l'importance de l'objet Window dans le JavaScript côté client ?

Réponse:L'objet Window représente le contexte

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

global pour le JavaScript côté client et englobe toutes les propriétés et méthodes de premier niveau disponibles dans le navigateur. Il sert d'espace de noms pour les fonctions globales et permet aux développeurs de contrôler des aspects de l'environnement du navigateur, tels que l'affichage de dialogues, le suivi des interactions utilisateur et la navigation dans l'historique du navigateur.

2.Question

Comment créer et afficher une boîte de dialogue en utilisant l'objet Window ? Pouvez-vous fournir un exemple ?

Réponse:Vous pouvez créer des boîtes de dialogue en utilisant des méthodes comme alert(), confirm() et prompt(). Par exemple, alert('Bienvenue sur ma page d'accueil !') affichera une simple boîte de dialogue d'alerte. Pour demander une entrée à l'utilisateur, utilisez prompt('Entrez votre nom'); cela ouvrira une boîte de dialogue permettant à l'utilisateur d'entrer une ligne de texte.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

3.Question

Quelles méthodes permettent de manipuler la ligne de statut dans un navigateur web ?

Réponse:Vous pouvez manipuler la ligne de statut en utilisant les propriétés status et defaultStatus de l'objet Window. Par exemple, définir window.status = 'Chargement...' changera le texte dans la ligne de statut en 'Chargement...' jusqu'à ce qu'un autre statut soit défini.

4.Question

Pouvez-vous expliquer comment fonctionnent les minuteries en JavaScript en utilisant l'objet Window ?

Réponse:Les minuteries en JavaScript sont créées en utilisant setTimeout() et setInterval(). setTimeout() exécute une fonction une fois après un délai spécifié, tandis que setInterval() exécute de manière répétée une fonction à intervalles définis. Par exemple, var timer = setInterval(function() { console.log('Tick'); }, 1000) affichera 'Tick' dans la console chaque seconde.

5.Question

Quelle est la propriété navigator de l'objet Window et

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

comment est-elle utilisée ?

Réponse:La propriété navigator fournit des informations sur le navigateur et le système d'exploitation de l'utilisateur. Elle retourne un objet Navigator qui contient des propriétés comme appName, appVersion et userAgent. Ces informations peuvent être utilisées pour adapter les fonctionnalités en fonction du navigateur, par exemple if (navigator.appName == 'Netscape') { // Code spécifique à Netscape }.

6.Question

Comment la propriété location affecte-t-elle la navigation dans le navigateur ?

Réponse:La propriété location représente l'URL actuelle affichée dans la barre d'adresse du navigateur. Vous pouvez la lire pour obtenir l'URL actuelle ou la définir pour naviguer vers une autre page web. Par exemple, définir location.href = 'http://exemple.com' fera naviguer le navigateur vers la page web spécifiée.

7.Question

Quelles sont les méthodes disponibles pour contrôler la

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

taille et la position de la fenêtre du navigateur ?
Réponse:Les méthodes pour contrôler la fenêtre du navigateur incluent moveTo(), moveBy(), resizeTo() et resizeBy(). Celles-ci permettent aux développeurs de changer la position et la taille d'une fenêtre par programmation. Par exemple, resizeTo(800, 600) changerait la taille de la fenêtre à une largeur de 800 pixels et une hauteur de 600 pixels.

8.Question

Quel est l'objectif d'utiliser plusieurs fenêtres et cadres dans le JavaScript ?
Réponse:L'utilisation de plusieurs fenêtres et cadres permet aux développeurs de créer des applications web complexes où différents ensembles d'informations peuvent être affichés simultanément de manière conviviale. Chaque fenêtre ou cadre fonctionne avec son propre contexte d'exécution, ce qui signifie que les variables et les fonctions peuvent être limitées à leurs fenêtres respectives, favorisant un design de code modulaire.

9.Question

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Comment un script dans un cadre imbriqué peut-il accéder aux fonctions définies dans la fenêtre de niveau supérieur ?

Réponse:Un script s'exécutant dans un cadre imbriqué peut accéder aux fonctions et variables définies dans la fenêtre de niveau supérieur en utilisant la propriété top. Par exemple, si une fonction stop_scrolling() est définie dans la fenêtre de niveau supérieur, un script dans un cadre imbriqué peut appeler cette fonction en utilisant top.stop_scrolling();.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Ad

Scanner pour télécharger

Essayez l'appli Bookey pour lire plus de 1000 résumés des meilleurs livres du monde

Débloquez 1000+ titres, 80+ sujets

Nouveaux titres ajoutés chaque semaine

Brand | Leadership & collaboration | Gestion du temps | Relations & communication | Know

tégie d'entreprise | Créativité | Mémoires | Argent & investissements | Positive Psychology

Entrepreneuriat | Histoire du monde | Communication parent-enfant | Soins Personnels

Aperçus des meilleurs livres du monde

Essai gratuit avec Bookey

Chapitre 10 | Section 2.3. L'objet Document| Questions et réponses

1.Question

Quelle est l'importance de l'objet Document en JavaScript par rapport à l'objet Window ?

Réponse:L'objet Document est plus significatif que l'objet Window car, tandis que Window représente l'interface du navigateur, l'objet Document représente le document HTML affiché dans cette interface. Il sert de passerelle vers le contenu de la page web, permettant aux développeurs d'accéder et de manipuler les éléments HTML directement via le modèle d'objet document (DOM). Comprendre l'objet Document est essentiel pour un développement web efficace.

2.Question

Quels sont les différents types de DOM mentionnés et en quoi diffèrent-ils ?

Réponse:Il existe trois principaux types de DOM discutés : le DOM Legacy, le DOM W3C et le DOM IE 4. Le DOM

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Legacy représente un ancien modèle qui prend en charge des éléments de base comme les formulaires et les images, mais est limité dans sa portée. Le DOM W3C est un modèle moderne et standardisé qui permet un accès et une modification complets du contenu du document et est largement supporté par les navigateurs actuels. Le DOM IE 4 a introduit des fonctionnalités propriétaires pour la manipulation de documents qui n'étaient pas standardisées, mais qui restent influentes. Ces différences illustrent l'évolution des normes web et le besoin de compatibilité.

3.Question

Pourquoi un programmeur JavaScript devrait-il être conscient des variations du DOM ?

Réponse:Un programmeur JavaScript devrait être conscient des variations du DOM pour écrire un code compatible sur différents navigateurs et tirer parti des fonctionnalités spécifiques de manière précise. Comprendre ces différences aide à développer des applications robustes qui fonctionnent de manière cohérente, quel que soit le choix du navigateur de

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

l'utilisateur. La connaissance du DOM W3C est particulièrement importante pour maintenir les normes web actuelles, car elle garantit une plus grande compatibilité et donne un avenir au code.

4.Question

Que puis-je faire pour explorer davantage le DOM pour la programmation JavaScript ?

Réponse:Pour explorer davantage le DOM pour la programmation JavaScript, vous pouvez lire des ressources complètes telles que 'JavaScript', qui couvre toutes les fonctionnalités essentielles du DOM standard. De plus, l'expérimentation pratique à travers la création d'applications HTML et la manipulation d'éléments avec JavaScript approfondira votre compréhension et améliorera vos compétences. Consultez la documentation et les tutoriels en ligne pour rester à jour sur les meilleures pratiques et les nouvelles normes web.

Chapitre 11 | Section 2.4. Le DOM Héritage| Questions et réponses

1.Question

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Quels sont les éléments clés du DOM hérité qui améliorent l'interaction avec le document en JavaScript ?

Réponse:Le DOM hérité fournit des propriétés comme les formulaires, les images, les applets, les liens et les ancres, qui permettent d'accéder à des éléments spécifiques dans un document. Ces propriétés sont des tableaux qui contiennent des objets correspondant à chacun de ces éléments dans l'ordre où ils apparaissent dans le document. De plus, l'objet Form contient un tableau d'éléments pour les éléments de formulaire, ce qui vous permet de les référencer par index ou par nom.

2.Question

Comment fonctionne la méthode write() dans le DOM hérité et quelles en sont les limites ?

Réponse:La méthode write() permet d'afficher du texte dans le document à l'emplacement du script, ce qui permet d'inclure des balises HTML dans le texte. Cependant, elle ne peut être utilisée que pendant que le document est encore en

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

cours de chargement, car l'appeler après aurait pour effet d'effacer le contenu du document. Elle peut également être utilisée dans des gestionnaires d'événements pour différentes fenêtres, mais nécessite d'écrire l'intégralité du contenu du document et d'appeler document.close().

3.Question

Pouvez-vous expliquer comment la validation des formulaires est implémentée dans le DOM hérité en utilisant JavaScript ?

Réponse:La validation des formulaires est gérée via l'événement onsubmit de la balise <form>. En retournant false depuis le gestionnaire d'événements, il est possible d'empêcher la soumission du formulaire. Vous pouvez passer en boucle à travers les éléments du formulaire, vérifier les champs requis et alerter l'utilisateur si l'un d'eux n'est pas rempli. Par exemple, une fonction peut être créée pour itérer à travers chaque champ de saisie texte et alerter l'utilisateur s'il est vide.

4.Question

Quelles techniques peuvent être utilisées pour les rollover

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

d'images sur les pages web utilisant le DOM hérité ?
Réponse:Les rollover d'images sont réalisés en changeant la propriété src d'un objet Image en fonction des événements de souris comme onmouseover et onmouseout. Le préchargement d'images en mémoire à l'aide d'un objet Image hors écran peut améliorer l'expérience utilisateur en éliminant les délais de chargement lorsque l'image doit être affichée.

5.Question

Comment gérer les cookies dans le DOM hérité ?
Réponse:Les cookies peuvent être définis et récupérés en utilisant la propriété cookie de l'objet Document. Pour créer un cookie, vous assignez une chaîne formatée comme 'nom=valeur' à la propriété cookie. Vous pouvez également définir une date d'expiration et un chemin pour le cookie. Pour récupérer les cookies, vous pouvez diviser la chaîne de cookies du document et rechercher le nom du cookie souhaité.

6.Question

Quelle est l'importance du DOM hérité malgré ses

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

limitations ?

Réponse:Le DOM hérité fournit des fonctionnalités essentielles pour la manipulation de contenu dynamique dans les pages web, y compris la modification de la structure du document, le traitement des soumissions de formulaires, la validation des saisies utilisateur, la création de formulaires dynamiques et la gestion des cookies. Bien qu'il manque de méthodes complètes pour accéder à tous les éléments du document, il jette les bases pour des API DOM plus avancées qui abordent ces lacunes.

7.Question

Comment pouvez-vous mettre à jour dynamiquement le contenu d'un élément de formulaire dans le DOM hérité ?

Réponse:Vous pouvez mettre à jour le contenu d'un élément de formulaire en y accédant via le tableau document.forms et en définissant ses propriétés. Par exemple, pour afficher l'heure actuelle chaque seconde, vous pouvez définir la propriété value d'un élément de saisie texte dans un objet Form en utilisant la fonction setInterval de JavaScript.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 12 | Section 2.5. Le W3C DOM| Questions et réponses

1.Question

Quels sont les avantages de l'utilisation du W3C DOM par rapport au DOM hérité en JavaScript ?

Réponse:Le W3C DOM standardise la plupart des fonctionnalités du DOM hérité tout en ajoutant de nouvelles capacités, telles que la possibilité de manipuler tous les éléments du document de manière uniforme, et pas seulement des éléments à usage spécial comme les formulaires et les images. Cela facilite l'accès, la navigation et la manipulation de n'importe quelle partie du document par les scripts.

2.Question

Comment pouvons-nous accéder aux éléments d'un document en utilisant leur identifiant ?

Réponse:Vous pouvez accéder aux éléments par leur identifiant en utilisant la méthode getElementById() de l'objet Document. Par exemple, si vous avez un élément <h1

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

id='title'>Titre</h1>, vous pouvez le récupérer en JavaScript avec var t = document.getElementById('title');.

3.Question

Quelle est la structure d'un document dans le W3C DOM ?

Réponse:Le W3C DOM représente un document sous forme d'arbre, où chaque balise HTML, chaîne de texte et commentaire correspond à des nœuds dans cet arbre. Chaque nœud est un objet JavaScript avec des propriétés qui permettent la navigation et la manipulation d'autres nœuds.

4.Question

Quels sont les différents types de nœuds dans un document W3C DOM ?

Réponse:Les principaux types de nœuds pertinents pour les documents HTML comprennent : 1 pour les nœuds Élément (balises HTML), 2 pour les nœuds Texte (texte dans le document), 8 pour les nœuds Commentaire (commentaires HTML), et 9 pour les nœuds Document (le document lui-même). Ces types de nœuds aident à organiser la structure

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

du document.

5.Question

Comment pouvez-vous changer le contenu textuel d'un élément de document ?

Réponse:Le contenu textuel d'un élément de document peut être changé en définissant la propriété nodeValue d'un nœud Texte. Alternativement, vous pouvez utiliser la propriété innerHTML pour définir directement le contenu HTML, ce qui permet des structures plus complexes.

6.Question

Comment pouvons-nous manipuler la structure d'un document dans le W3C DOM ?

Réponse:Vous pouvez manipuler la structure du document en utilisant les méthodes Node pour insérer, ajouter, supprimer ou remplacer des nœuds enfants, ainsi que les méthodes Document pour créer de nouveaux nœuds Élément et Texte. Par exemple, pour ajouter un nouvel élément de liste à une liste ordonnée, vous pourriez créer un nouvel élément <li> et l'ajouter à la <ol>.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

7.Question

Pouvez-vous fournir un exemple de changement de texte dans un document ?

Réponse:Bien sûr ! Pour changer le texte de la première balise <h1> dans un document, vous utiliseriez le code suivant : var h1 = document.getElementsByTagName('h1')[0]; h1.firstChild.nodeValue = 'Nouveau titre'; N'oubliez pas que cela suppose que le <h1> ne contient que du texte simple.

8.Question

Quelle méthode peut être utilisée pour créer de nouveaux éléments dans le document ?

Réponse:La méthode createElement() de l'objet Document peut être utilisée pour créer de nouveaux éléments. Par exemple, var item = document.createElement('li'); crée un nouvel élément <li>.

9.Question

Comment pouvez-vous supprimer un élément du document ?

Réponse:Vous pouvez supprimer un élément en utilisant la

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

méthode removeChild(). Par exemple, si vous souhaitez supprimer un élément ajouté précédemment d'une liste, vous appelleriez list.removeChild(item); sur la liste parent.

10.Question

Quel est le but de la fonction embolden dans l'exemple fourni ?

Réponse:La fonction embolden crée une nouvelle balise <b> (gras) et réinsère un nœud de document spécifié à l'intérieur, rendant ainsi le nœud original en gras. Cela illustre comment manipuler la structure du document et appliquer du formatage de manière programmatique.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Ad

Scanner pour télécharger

Pourquoi Bookey est une application incontournable pour les amateurs de livres

Contenu de 30min
Plus notre interprétation est profonde et claire, mieux vous saisissez chaque titre.

Format texte et audio
Absorbez des connaissances même dans un temps fragmenté.

Quiz
Vérifiez si vous avez maîtrisé ce que vous venez d'apprendre.

Et plus
Plusieurs voix & polices, Carte mentale, Citations, Clips d'idées...

Essai gratuit avec Bookey

Chapitre 13 | Section 2.6. DOM IE 4| Questions et réponses

1.Question

Quelle est la principale différence entre le DOM d'IE 4 et le DOM W3C lors de l'accès aux éléments du document ?

Réponse:Le DOM d'IE 4 ne prend pas en charge la méthode getElementById() et permet plutôt d'accéder aux éléments via le tableau all[]. Vous pouvez accéder à un élément par son attribut id en utilisant des méthodes comme document.all['mylist'] ou document.all.mylist.

2.Question

En quoi la méthode de navigation dans l'arbre du document diffère-t-elle entre IE 4 et le DOM W3C ?

Réponse:Dans le DOM d'IE 4, vous utilisez des propriétés comme children[] et parentElement au lieu de childNodes[] et parentNode comme dans le DOM W3C. De plus, le DOM d'IE 4 ignore les commentaires et le texte du document en tant que partie de sa structure d'arbre.

3.Question

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Quelles sont les implications de la modification du contenu du document dans IE 4 en utilisant innerText et innerHTML ?

Réponse:L'utilisation de innerText remplace tout le contenu d'un élément par le texte spécifié, effaçant les balises existantes. En revanche, innerHTML vous permet de remplacer le contenu d'un élément par du HTML, mais nécessite l'invocation du parseur HTML, ce qui peut impacter l'efficacité tout en offrant plus de commodité.

4.Question

Pourquoi la compatibilité est-elle importante lors de l'écriture de scripts pour différents DOM ?

Réponse:La compatibilité est cruciale car différents navigateurs mettent en œuvre les normes DOM de manière différente. L'utilisation de tests de capacité vous permet d'écrire des scripts qui s'adaptent aux fonctionnalités disponibles dans le navigateur de l'utilisateur, garantissant ainsi le bon fonctionnement à la fois sur les navigateurs modernes et anciens.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

5.Question

Quels sont les méthodes liées à la modification de la structure du document disponibles dans le DOM d'IE 4 ?

Réponse:Les méthodes non nécessaires pour modifier la structure du document dans le DOM d'IE 4 incluent innerHTML, outerHTML et insertAdjacentHTML(). Cependant, innerHTML est la plus couramment utilisée pour sa commodité, bien qu'elle ne soit pas standard.

6.Question

Comment les développeurs devraient-ils aborder les fonctionnalités héritées du DOM, comme le DOM d'IE 4 ?

Réponse:Les développeurs devraient être conscients des fonctionnalités héritées et envisager des approches alternatives qui privilégient les pratiques standard, comme l'utilisation du DOM W3C lorsque cela est possible. Il est également bénéfique de mettre en œuvre des stratégies de dégradé élégant pour garantir que les anciens navigateurs peuvent toujours fonctionner.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 14 | Section 2.7. DHTML : Scriptage des styles CSS| Questions et réponses

1.Question

Quel est le rôle du DHTML dans le développement web ?

Réponse:Le DHTML, qui signifie HTML dynamique, combine HTML, CSS et JavaScript pour créer des expériences interactives et dynamiques sur le web. Il permet aux développeurs de manipuler le style et le comportement des éléments HTML en temps réel, améliorant ainsi l'expérience utilisateur en répondant aux actions sans recharger la page.

2.Question

Comment pouvez-vous modifier le style des éléments HTML en utilisant JavaScript ?

Réponse:Vous pouvez modifier le style des éléments HTML en accédant à la propriété 'style' d'un élément via JavaScript. Par exemple, pour changer la couleur d'un élément, vous utiliseriez 'element.style.color = "red";'. Pour définir une couleur de fond, vous utiliseriez

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

'element.style.backgroundColor = "blue";'.

3.Question

Quelle est la première étape pour animer un élément en utilisant DHTML ?

Réponse:La première étape pour animer un élément en utilisant le DHTML est de définir sa propriété de position sur 'absolute', ce qui vous permet de définir sa localisation sur la page plus librement en utilisant les propriétés top et left.

4.Question

Pourquoi est-il important de convertir des nombres en chaînes lors de la définition de certaines propriétés CSS ?

Réponse:En JavaScript, toutes les valeurs de propriété CSS sont traitées comme des chaînes, même celles qui représentent des valeurs numériques telles que la largeur ou la hauteur. Cela signifie que lorsque vous définissez des propriétés comme 'element.style.width', vous devez convertir la valeur numérique en chaîne et ajouter l'unité appropriée, généralement 'px', pour garantir qu'elle soit correctement interprétée par le navigateur.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

5.Question

Quel effet a la propriété 'visibility' sur un élément ?

Réponse:La propriété 'visibility' détermine si un élément est visible sur la page web. En la définissant sur 'hidden', il est retiré de la vue, mais l'espace qu'il occupe reste alloué, tandis qu'en la définissant sur 'visible', il apparaît à nouveau.

6.Question

Comment la fonction 'nextFrame()' illustre-t-elle l'animation DHTML ?

Réponse:La fonction 'nextFrame()' illustre l'animation DHTML en déplaçant progressivement un élément HTML de 10 pixels vers la droite toutes les 50 millisecondes. Elle utilise une boucle contrôlée par un compteur de frames pour limiter le mouvement à 20 frames, après quoi elle cache l'élément en définissant sa visibilité sur 'hidden'.

7.Question

Que se passe-t-il si vous définissez la propriété 'position' d'un élément sur 'fixed' ?

Réponse:Lorsque la propriété 'position' est définie sur 'fixed', l'élément est positionné par rapport à la fenêtre d'affichage.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Cela signifie qu'il restera au même endroit à l'écran même lorsque la page est défilée, permettant ainsi de créer des barres de navigation ou des notifications persistantes.

8.Question

Que contrôle la propriété 'zIndex' ?

Réponse:La propriété 'zIndex' contrôle l'ordre de superposition des éléments sur la page web. Les éléments avec une valeur z-index plus élevée apparaîtront au-dessus de ceux avec une valeur inférieure, permettant ainsi des conceptions en couches.

Chapitre 15 | Section 2.8. Événements et Gestion des Événements| Questions et réponses

1.Question

Quel est le but des attributs de gestionnaire d'événements en HTML, et comment fonctionnent-ils en JavaScript ?

Réponse:Les attributs de gestionnaire d'événements en HTML, tels que onclick, onsubmit et onload, sont utilisés pour spécifier des fonctions JavaScript qui doivent être exécutées en réponse à des actions spécifiques de l'utilisateur ou à des événements du

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

navigateur. Ces attributs sont attachés aux balises HTML et commencent par 'on'. Lorsque un événement se produit, le navigateur exécute le code JavaScript associé. Par exemple, si vous avez un gestionnaire d'événements 'onsubmit' dans un formulaire, il peut être utilisé pour valider les entrées de l'utilisateur avant que le formulaire ne soit soumis, améliorant ainsi l'interaction utilisateur et permettant un retour d'information en temps réel.

2.Question

Comment les gestionnaires d'événements peuvent-ils être définis différemment en JavaScript par rapport aux attributs HTML ?

Réponse:En JavaScript, les gestionnaires d'événements peuvent être définis comme de véritables fonctions assignées aux propriétés des objets document au lieu d'être écrits sous forme de chaînes de code dans les attributs HTML. Par exemple, vous pouvez créer une fonction appelée validate pour la soumission d'un formulaire et l'assigner à la propriété

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

onsubmit d'un formulaire en utilisant 'document.forms[0].onsubmit = validate;'. Cette méthode permet d'avoir un code plus propre et une meilleure programmabilité.

3.Question

Quels sont les défis posés par les différents modèles de gestion d'événements à travers les navigateurs ?

Réponse:Il existe trois principaux modèles de gestion d'événements : le modèle DOM W3C, le modèle Internet Explorer et le modèle Netscape 4. Chacun de ces modèles a des manières uniques de gérer les événements, ce qui complique la compatibilité entre navigateurs. Par exemple, dans le modèle W3C, l'objet Event est passé à la fonction gestionnaire, alors qu'en IE, il est stocké dans la propriété événement de la fenêtre. En raison de ces différences, les développeurs doivent souvent écrire du code supplémentaire ou utiliser des bibliothèques pour s'assurer que leurs fonctions de gestion d'événements fonctionnent correctement sur tous les principaux navigateurs.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

4.Question

Comment fonctionne la propagation des événements dans les modèles d'événements avancés ?

Réponse:La propagation des événements permet aux événements de voyager à travers la hiérarchie des éléments du document, déclenchant tout gestionnaire d'événements pertinent en cours de route. Dans les modèles W3C et Netscape, les événements commencent à l'objet document et se propagent jusqu'à l'élément cible, tandis que dans le modèle IE, certains événements peuvent remonter après avoir été traités. Les développeurs peuvent définir des gestionnaires d'événements à plusieurs niveaux, par exemple en ajoutant un écouteur onclick à un qui gère les clics de n'importe quel élément enfant, simplifiant ainsi la gestion des événements.

5.Question

Quelle est la signification de la méthode addEventListener() dans le modèle W3C ?

Réponse:La méthode addEventListener() dans le modèle

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

W3C permet aux développeurs d'attacher plusieurs gestionnaires d'événements à un seul type d'événement sans écraser les gestionnaires existants. Cela est significatif pour la création d'applications complexes où vous souhaitez que diverses fonctions réagissent à la même action de l'utilisateur, améliorant ainsi la modularité et la maintenabilité du code.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Ad

App Store
Coup de cœur

22k avis 5 étoiles

Scanner pour télécharger

Retour Positif

Fabienne Moreau

ue résumé de livre ne testent
ion, mais rendent également
musant et engageant.
té la lecture pour moi.

Fantastique!

Giselle Dubois

Je suis émerveillé par la variété de livres et de langues que Bookey supporte. Ce n'est pas juste une application, c'est une porte d'accès au savoir mondial. De plus, gagner des points pour la charité est un grand plus !

Fi

Le
liv
co
pr

é Blanchet

de lecture
eption de
es,
cous.

J'adore !

Isoline Mercier

Bookey m'offre le temps de parcourir les parties importantes d'un livre. Cela me donne aussi une idée suffisante pour savoir si je devrais acheter ou non la version complète du livre ! C'est facile à utiliser !"

Gain de temps !

Bookey est mon applicat
intellectuelle. Les résum
magnifiquement organis
monde de connaissance

ppli géniale !

Joachim Lefevre

adore les livres audio mais je n'ai pas toujours le temps
'écouter le livre entier ! Bookey me permet d'obtenir
n résumé des points forts du livre qui m'intéresse !!!
Quel super concept !!! Hautement recommandé !

Appli magnifique

Cette application est une bouée de sauve
amateurs de livres avec des emplois du te
Les résumés sont précis, et les cartes mer
renforcer ce que j'ai appris. Hautement re

Essai gratuit avec Bookey

Chapitre 16 | Section 2.9. Restrictions de sécurité en JavaScript| Questions et réponses

1.Question

Quelles sont les raisons des restrictions de sécurité dans JavaScript côté client ?

Réponse:Les restrictions de sécurité dans JavaScript côté client sont imposées principalement pour protéger les données des utilisateurs et garantir leur vie privée. Par exemple, empêcher les scripts de supprimer des fichiers sur le disque dur local d'un utilisateur évite les accès non autorisés et la perte potentielle de données. De même, la politique de même origine limite l'accès des scripts au contenu chargé depuis différents serveurs web, ce qui aide à prévenir des actions malveillantes telles que le vol de données et les attaques par injection de scripts. En limitant des capacités comme les soumissions d'e-mails et les téléchargements de fichiers sans le consentement de l'utilisateur, les navigateurs protègent les interactions des utilisateurs,

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

maintenant ainsi une confiance générale dans les applications web.

2.Question

Comment fonctionne la politique de même origine dans la sécurité JavaScript ?

Réponse:La politique de même origine est un mécanisme de sécurité crucial dans JavaScript côté client qui interdit aux scripts de lire les propriétés des fenêtres ou des documents chargés d'une origine différente de celle du script lui-même. Par exemple, si vous avez un script exécuté sur 'example.com', il ne peut pas interagir avec un document de 'anotherdomain.com'. Cela empêche les scripts d'accéder à des informations sensibles que l'utilisateur pourrait avoir ouvertes dans d'autres onglets ou fenêtres, sécurisant ainsi les données des utilisateurs contre les accès non autorisés.

3.Question

Les scripts JavaScript peuvent-ils fermer n'importe quelle fenêtre de navigateur ?

Réponse:Non, les scripts ne peuvent fermer que les fenêtres

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

de navigateur qu'ils ont eux-mêmes créées. Cette restriction est mise en place pour prévenir les pratiques abusives où les scripts pourraient fermer arbitrairement des fenêtres que l'utilisateur n'avait pas l'intention de fermer, ce qui pourrait être très frustrant ou trompeur pour l'utilisateur.

4.Question

Quelle est l'importance de la confirmation de l'utilisateur dans la sécurité JavaScript ?

Réponse:La confirmation de l'utilisateur agit comme un filtre crucial pour les actions qui pourraient perturber l'expérience utilisateur ou compromettre la sécurité. Par exemple, lorsqu'un script essaie de soumettre un formulaire à une adresse e-mail ou de publier des nouvelles, exiger une confirmation de l'utilisateur garantit que les utilisateurs sont conscients de ces actions, empêchant ainsi des soumissions non désirées et préservant l'intégrité des interactions utilisateurs.

5.Question

Comment les mises à jour récentes des navigateurs ont-elles influencé les normes de sécurité de JavaScript ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:Les navigateurs récents se sont adaptés aux préoccupations croissantes en matière de sécurité et à l'utilisation abusive de JavaScript en introduisant des paramètres de sécurité configurables par l'utilisateur. Cela permet aux utilisateurs d'empêcher les scripts d'effectuer des actions gênantes, telles que l'ouverture de publicités contextuelles ou la manipulation des tailles des fenêtres actuelles. Ces améliorations témoignent de la réactivité des développeurs de navigateurs aux besoins des utilisateurs pour un contrôle accru et une meilleure cybersécurité.

6.Question

Pourquoi est-il important que les navigateurs web aient des restrictions de sécurité pour JavaScript ?

Réponse:Les restrictions de sécurité sont essentielles pour maintenir un environnement de navigation sécurisé. Elles protègent les utilisateurs des actions potentiellement nuisibles qui pourraient être exécutées via des scripts malveillants. En appliquant ces règles, les navigateurs réduisent le risque de vulnérabilités pouvant compromettre

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

les données des utilisateurs, empêcher les violations de la vie privée et maintenir la confiance des utilisateurs dans les applications web.

Chapitre 17 | Tableau| Questions et réponses

1.Question

Quelles sont les différentes manières de créer un tableau en JavaScript ?

Réponse:Vous pouvez créer un tableau en utilisant le constructeur `new Array()`, qui crée un tableau vide, `new Array(n)` pour un tableau avec 'n' éléments non spécifiés, ou `new Array(e0, e1,...)` pour créer un tableau avec des éléments spécifiés. De plus, en utilisant la syntaxe littérale, vous pouvez créer et initialiser un tableau avec une liste séparée par des virgules à l'intérieur des crochets, comme `var a = [1, true, 'abc'];`.

2.Question

Comment fonctionne la propriété `length` dans les tableaux JavaScript ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:La propriété `length` représente le nombre d'éléments dans le tableau. Si les éléments ne sont pas contigus (c'est-à-dire s'il y a des 'trous' dans le tableau), la propriété `length` renverra un nombre supérieur de un à l'index le plus élevé ayant une valeur.

3.Question

Quel est l'objectif de la méthode `concat()` ? Pouvez-vous donner un exemple ?

Réponse:La méthode `concat()` est utilisée pour combiner plusieurs tableaux ou valeurs en un nouveau tableau. Par exemple, si vous avez deux tableaux `var a = [1, 2]` et `var b = [3, 4]`, en utilisant `a.concat(b)` vous obtiendrez un nouveau tableau `[1, 2, 3, 4]`.

4.Question

Expliquez l'utilisation de la méthode `join()` avec un exemple.

Réponse:La méthode `join()` convertit tous les éléments d'un tableau en une chaîne de caractères, en insérant un séparateur spécifié entre eux. Par exemple, si vous avez un tableau `var

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

fruits = ['apple', 'banana', 'cherry']`, en utilisant `fruits.join(', ')` vous obtiendrez la chaîne 'apple, banana, cherry'.

5.Question

Que se passe-t-il lorsque vous utilisez la méthode `pop()` ?

Réponse:Lorsque vous utilisez la méthode `pop()` sur un tableau, elle supprime le dernier élément du tableau et renvoie cet élément, tout en diminuant la longueur du tableau de un. Par exemple, si vous avez `var numbers = [1, 2, 3]` et que vous faites `numbers.pop()`, cela renverra '3' et `numbers` deviendra `[1, 2]`.

6.Question

Décrivez la différence entre les méthodes `shift()` et `unshift()`.

Réponse:La méthode `shift()` supprime le premier élément d'un tableau et le renvoie, décalant tous les éléments restants vers le bas. Par exemple, à partir de `var arr = [1, 2, 3]`, appeler `arr.shift()` renverra '1' et changera `arr` en `[2, 3]`. À l'inverse, `unshift()` ajoute un ou plusieurs éléments au début du tableau, décalant les éléments existants vers le haut. Si

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

vous appelez `arr.unshift(0)` après le `shift()` précédent, `arr` deviendra `[0, 2, 3]`.

7.Question

Quelle est la fonction de la méthode `splice()` ? Fournissez un exemple.

Réponse:La méthode `splice()` modifie un tableau en supprimant un certain nombre d'éléments à un index spécifié et peut également ajouter de nouveaux éléments à cet index. Par exemple, si vous avez `var arr = [1, 2, 3, 4, 5]` et que vous appelez `arr.splice(2, 2, 'a', 'b')`, cela supprimera deux éléments à partir de l'index 2 (initialement '3' et '4') et insérera 'a' et 'b' à leur place. Le tableau résultant sera `[1, 2, 'a', 'b', 5]`.

8.Question

Pouvez-vous expliquer comment fonctionne la méthode `sort()` ?

Réponse:La méthode `sort()` trie les éléments d'un tableau sur place et peut prendre une fonction optionnelle définissant l'ordre dans lequel trier les éléments. Par exemple, si vous

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

avez `var nums = [3, 1, 2]` et utilisez `nums.sort()`, cela triera le tableau en `[1, 2, 3]`. Vous pouvez également fournir une fonction de tri personnalisée, comme trier des nombres par ordre décroissant avec `nums.sort((a, b) => b - a)`.

9.Question

Que fait la méthode `reverse()` ?

Réponse:La méthode `reverse()` inverse l'ordre des éléments dans un tableau. Par exemple, si vous avez `var arr = [1, 2, 3]` et que vous appelez `arr.reverse()`, cela modifiera `arr` pour devenir `[3, 2, 1]`.

10.Question

Comment les méthodes comme `toString()` et `toLocaleString()` diffèrent-elles dans leur utilisation ?

Réponse:La méthode `toString()` convertit un tableau en une chaîne de caractères sans aucune localisation, tandis que `toLocaleString()` respecte les paramètres régionaux de l'environnement de l'utilisateur, formatant potentiellement les chaînes différemment en fonction des conventions locales. Par exemple, l'appel à `arr.toString()` pourrait renvoyer

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

'1,2,3', tandis que `arr.toLocaleString()` pourrait renvoyer '1.0, 2.0, 3.0' dans certaines locales, selon le format des nombres.

Chapitre 18 | Date| Questions et réponses

1.Question

Quel est le but du constructeur Date en JavaScript ?

Réponse:Le constructeur Date permet de créer des objets de date, qui peuvent représenter la date et l'heure actuelles, analyser des chaînes représentant des dates, ou spécifier des composants individuels de date comme l'année, le mois, le jour, les heures, etc. Cette flexibilité permet aux développeurs de manipuler les dates et les heures selon leurs besoins spécifiques.

2.Question

Comment fonctionne la méthode getTime() et pourquoi est-elle importante ?

Réponse:La méthode getTime() renvoie le nombre de millisecondes écoulées depuis minuit UTC du 1er janvier

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

1970 pour l'objet Date. Cette représentation numérique est cruciale pour effectuer des calculs et des comparaisons de dates, car elle permet aux développeurs de comprendre et de manipuler facilement des données basées sur le temps.

3.Question

Quelles sont les différences entre les méthodes de temps local et UTC dans l'objet Date ?

Réponse:Les méthodes de temps local interagissent avec les paramètres de fuseau horaire locaux de l'utilisateur, tandis que les méthodes UTC fournissent des résultats cohérents basés sur le Temps Universel Coordonné. Cette distinction est importante pour les applications qui doivent tenir compte des fuseaux horaires ou effectuer des opérations sur des dates et des heures de manière cohérente, quelle que soit la localisation de l'utilisateur.

4.Question

Pouvez-vous expliquer l'importance de la méthode getTimezoneOffset() ?

Réponse:La méthode getTimezoneOffset() renvoie la

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

différence en minutes entre le fuseau horaire local et UTC. Cela est significatif pour comprendre comment les heures locales se rapportent à l'heure universelle et pour planifier des tâches qui nécessitent une synchronisation entre différents fuseaux horaires.

5.Question

Quelles sont les conséquences d'utiliser des méthodes obsolètes comme getYear() et setYear() ?

Réponse:Utiliser des méthodes obsolètes comme getYear() et setYear() peut entraîner des problèmes de clarté et de compatibilité, car ces méthodes peuvent ne pas se comporter comme prévu dans les environnements JavaScript modernes. Il est conseillé aux développeurs d'utiliser leurs homologues mises à jour comme getFullYear() et setUTCFullYear() pour une meilleure précision et cohérence.

6.Question

Comment les méthodes statiques comme Date.parse() et Date.UTC() améliorent-elles la fonctionnalité de l'objet Date ?

Réponse:Les méthodes statiques comme Date.parse() et

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Date.UTC() étendent les capacités de l'objet Date en permettant aux développeurs de convertir facilement des chaînes en objets de date et de créer des représentations UTC des dates sans avoir à instancier d'abord un objet Date. Cela simplifie la manipulation des dates et améliore l'efficacité globale du code.

7.Question

Pourquoi est-il essentiel pour les développeurs de comprendre à la fois les méthodes get et set pour l'objet Date ?

Réponse:Comprendre les méthodes get et set pour l'objet Date est crucial pour les développeurs, car cela leur permet de récupérer et de modifier avec précision les valeurs de date et d'heure dans leurs applications. Cette connaissance est fondamentale pour des tâches telles que la planification, le formatage des dates et la garantie de calculs temporels corrects.

8.Question

Quel est l'avantage d'utiliser des méthodes comme toLocaleDateString() et toLocaleTimeString() ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:Des méthodes comme toLocaleDateString() et toLocaleTimeString() formatent les dates et les heures selon les paramètres de langue de l'utilisateur, ce qui facilite la présentation des informations de date de manière familière et compréhensible pour les utilisateurs. Cela améliore l'expérience utilisateur et assure une meilleure communication des informations liées à la date.

9.Question

Comment l'objet Date de JavaScript peut-il affecter les applications qui s'occupent de la planification ou des transactions sensibles au temps ?

Réponse:L'objet Date fournit des méthodes essentielles pour gérer et manipuler avec précision les données de date et d'heure, ce qui est crucial pour les applications qui traitent de la planification et des transactions sensibles au temps. En exploitant l'objet Date, les développeurs peuvent s'assurer que les événements sont correctement chronométrés et affichent les informations de date pertinentes de manière claire et précise pour les utilisateurs.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

10.Question

Quelles considérations les développeurs doivent-ils garder à l'esprit concernant l'heure d'été lors de l'utilisation des méthodes de l'objet Date ?

Réponse:Les développeurs doivent être conscients que certaines méthodes, comme getTimezoneOffset(), renverront des valeurs différentes selon que l'heure d'été est en vigueur ou non. Cela peut avoir un impact sur toute logique basée sur des calculs de date et d'heure, soulignant l'importance de tester ces calculs autour des dates de début et de fin de l'heure d'été.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Ad

Scanner pour télécharger

Lire, Partager, Autonomiser

Terminez votre défi de lecture, faites don de livres aux enfants africains.

Le Concept

BOOKS FOR AFRICA × ×

Cette activité de don de livres se déroule en partenariat avec Books For Africa. Nous lançons ce projet car nous partageons la même conviction que BFA : Pour de nombreux enfants en Afrique, le don de livres est véritablement un don d'espoir.

La Règle

Gagnez 100 points → **Échangez un livre** → **Faites un don à l'Afrique**

Votre apprentissage ne vous apporte pas seulement des connaissances mais vous permet également de gagner des points pour des causes caritatives ! Pour chaque 100 points gagnés, un livre sera donné à l'Afrique.

Essai gratuit avec Bookey

Chapitre 19 | Document| Questions et réponses

1.Question

Pourquoi l'objet Document est-il considéré comme l'un des objets les plus importants dans JavaScript côté client ?

Réponse:L'objet Document représente le document HTML affiché dans le navigateur, permettant aux développeurs d'interagir et de manipuler dynamiquement le contenu des pages web. Il fournit des méthodes et des propriétés essentielles pour accéder et modifier des éléments, des formulaires et d'autres composants de la page, créant ainsi un lien entre HTML et JavaScript.

2.Question

Quelle est l'importance de propriétés comme 'cookie' dans l'objet Document ?

Réponse:La propriété 'cookie' dans l'objet Document permet à JavaScript de lire et d'écrire des cookies associés au document. Cela est crucial pour maintenir l'état de session, les préférences des utilisateurs et gérer des données côté

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

client, améliorant ainsi l'expérience utilisateur sur les sites web.

3.Question

Comment les différentes implémentations de l'objet Document (comme W3C DOM, IE et Netscape) affectent-elles le développement web ?

Réponse:Les différentes implémentations de l'objet Document peuvent entraîner des problèmes de compatibilité entre les navigateurs. Par exemple, Netscape et IE ont introduit des propriétés et des méthodes non standard qui peuvent ne pas être prises en charge dans les navigateurs conformes au W3C. Cela oblige les développeurs à écrire du code spécifique aux navigateurs ou à utiliser la détection de fonctionnalités pour garantir un comportement cohérent sur différentes plateformes.

4.Question

Pourquoi des propriétés comme 'alinkColor' et 'bgColor' peuvent-elles être considérées comme obsolètes ?

Réponse:Les propriétés comme 'alinkColor' et 'bgColor' sont considérées comme obsolètes car elles reposent sur des

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

pratiques HTML plus anciennes qui ne sont plus recommandées dans le développement web moderne. Au lieu de cela, le CSS (Cascading Style Sheets) est privilégié pour le style des pages web, offrant plus de flexibilité et maintenant une séparation entre le contenu et la présentation.

5.Question

Comment l'état de chargement d'un document est-il lié à l'expérience utilisateur ?

Réponse:L'état de chargement, indiqué par la propriété 'readyState', impacte directement l'expérience utilisateur. Par exemple, si un document est encore 'en chargement', les utilisateurs peuvent voir un contenu partiel ou non réactif. Les développeurs peuvent utiliser cette information pour fournir des animations de chargement ou des espaces réservés, améliorant ainsi la performance perçue et l'interactivité de l'application.

6.Question

Quel rôle joue la propriété 'documentElement' dans l'accès à la structure HTML ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:La propriété 'documentElement' donne aux développeurs un accès direct à la balise <html> du document, servant de point de départ pour naviguer dans l'ensemble de la structure DOM. Cette propriété simplifie l'accès à la configuration générale de la page web et est essentielle pour appliquer des manipulations à l'échelle du document.

7.Question

Pourquoi est-il crucial de comprendre et de différencier les propriétés et méthodes supportées par les différentes versions de l'objet Document ?

Réponse:Comprendre les différences est crucial pour développer des applications web robustes et compatibles. Lorsque l'on utilise des fonctionnalités spécifiques à certains navigateurs ou versions de JavaScript, les développeurs peuvent éviter des pièges, garantir une accessibilité plus large et créer un code plus résilient qui s'adapte à des environnements variés.

8.Question

Comment l'objet Document a-t-il évolué depuis son introduction dans JavaScript 1.0 ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:L'objet Document a évolué de manière significative, avec des propriétés et des méthodes supplémentaires ajoutées dans les versions suivantes (par exemple, JS 1.1 et DOM Level 2). Cette évolution incorpore des pratiques standardisées du W3C, une meilleure compatibilité avec les normes web modernes et des fonctionnalités améliorées comme la création de nœuds et l'accès à du contenu dynamique.

9.Question

Quels sont les défis potentiels associés à l'utilisation de propriétés non standard définies par des navigateurs comme IE ?

Réponse:L'utilisation de propriétés non standard peut entraîner un comportement incohérent entre les différents navigateurs. Si un développeur s'appuie sur ces propriétés, l'application web peut mal fonctionner ou ne pas s'afficher correctement pour les utilisateurs utilisant des navigateurs qui ne prennent pas en charge ces fonctionnalités, compromettant ainsi l'accessibilité et l'expérience utilisateur.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

10.Question

Comment la méthode 'createElement' permet-elle aux développeurs de travailler avec JavaScript ?

Réponse:La méthode 'createElement' permet aux développeurs de créer dynamiquement de nouveaux éléments dans le document, leur permettant de construire des interfaces web interactives et réactives en temps réel. Cette capacité est fondamentale pour les applications web modernes qui nécessitent de maintenir une expérience utilisateur fluide sans rechargements constants de la page.

Chapitre 20 | Élément| Questions et réponses

1.Question

Quelle est l'importance de l'objet Element dans le DOM et comment facilite-t-il la manipulation des éléments HTML ?

Réponse:L'objet Element est essentiel dans le modèle d'objet de document (DOM) car il représente chaque élément ou balise HTML au sein d'un document. Cela permet aux développeurs d'interagir de manière programmatique avec la

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

structure, le contenu et le style d'une page web. Par exemple, en accédant à des propriétés telles que className, style, ou des méthodes comme setAttribute, les développeurs peuvent changer dynamiquement l'apparence et le comportement des éléments en fonction des interactions de l'utilisateur ou de la logique de l'application. Cette interaction relie la nature statique de l'HTML aux capacités dynamiques de JavaScript, améliorant ainsi l'expérience utilisateur.

2.Question

Comment les spécifications DOM du W3C et d'IE diffèrent-elles concernant les propriétés et méthodes disponibles pour les éléments HTML ?

Réponse:Les spécifications DOM du W3C et d'IE divergent considérablement tant au niveau des propriétés que des méthodes. Bien qu'il existe des propriétés communes comme className et tagName, le DOM du W3C respecte une norme qui est uniformément implémentée dans les navigateurs

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

conformes. En revanche, le DOM d'IE incorpore de nombreuses propriétés non standards — telles que innerHTML et outerHTML — qui ne sont pas définies par le W3C, entraînant des incohérences de comportement entre différents navigateurs. Cette variabilité oblige les développeurs à utiliser du code conditionnel pour garantir la compatibilité entre les navigateurs.

3.Question

Quel rôle jouent les gestionnaires d'événements dans les éléments HTML et comment peuvent-ils améliorer l'interaction utilisateur ?

Réponse:Les gestionnaires d'événements sont intégrés aux éléments HTML car ils définissent les diverses interactions qu'un utilisateur peut entreprendre, telles que les clics, les frappes de touches et les mouvements de souris. Par exemple, en utilisant le gestionnaire d'événements onclick, les développeurs peuvent déclencher des fonctions spécifiques en réponse aux clics des utilisateurs, permettant ainsi des mises à jour dynamiques du contenu ou le lancement

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

d'animations. Cette capacité transforme une page web statique en une application interactive, ce qui est essentiel pour créer des interfaces utilisateur réactives.

4.Question

Pourquoi est-il important de comprendre les propriétés et méthodes spécifiques fournies par les différentes implémentations du DOM ?

Réponse:Comprendre les propriétés et méthodes spécifiques des différentes implémentations du DOM est crucial pour les développeurs afin de s'assurer que leurs applications fonctionnent correctement sur divers navigateurs. Étant donné que des incompatibilités peuvent survenir en raison des variations dans la façon dont les navigateurs interprètent le DOM, les développeurs doivent être équipés de connaissances sur ces différences pour écrire un code robuste et compatible entre les navigateurs. Cette expertise aide à anticiper les problèmes pouvant survenir en raison de la dépendance à des fonctionnalités non standards dans un contexte spécifique à un navigateur.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

5.Question

Expliquez l'importance de méthodes comme getAttribute et setAttribute dans le contexte de l'objet Element.

Réponse:Les méthodes comme getAttribute et setAttribute offrent une fonctionnalité essentielle pour interagir avec les attributs HTML d'un élément. getAttribute permet aux développeurs de récupérer la valeur actuelle d'un attribut spécifique, facilitant ainsi la lecture et l'utilisation des données définies dans l'HTML. En revanche, setAttribute permet aux développeurs de modifier ou de définir des valeurs d'attribut de manière programmatique. Cela est crucial pour les applications côté client qui nécessitent des mises à jour dynamiques basées sur les interactions des utilisateurs ou sur d'autres logiques, améliorant ainsi l'interactivité et la fonctionnalité des applications web.

6.Question

Comment la propriété innerHTML diffère-t-elle de innerText en termes d'impact sur la gestion de contenu dans un document HTML ?

Réponse:La propriété innerHTML facilite la manipulation du

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

contenu HTML d'un élément, permettant aux développeurs d'insérer ou de modifier des structures HTML complètes, y compris des balises et des styles. Cela la rend puissante mais peut entraîner des risques de sécurité potentiels (comme les XSS) si elle n'est pas manipulée correctement. En revanche, innerText récupère ou modifie simplement le contenu textuel brut, éliminant toutes les balises HTML. Cette différence est vitale pour les développeurs, car comprendre quand utiliser chacune peut garantir le résultat souhaité, que ce soit pour maintenir le formatage ou simplement afficher du texte de manière statique.

Chapitre 21 | Événement| Questions et réponses

1.Question

Quel est le but de l'objet Event en JavaScript ?

Réponse:L'objet Event sert à fournir des détails sur un événement et à contrôler sa propagation, permettant ainsi la gestion des interactions utilisateur sur les pages web.

2.Question

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

En quoi les objets Event diffèrent-ils entre les différents navigateurs ?

Réponse:Des navigateurs comme IE 4, 5 et 6 utilisent des objets propriétaires qui diffèrent de l'objet Event standardisé défini dans le DOM de niveau 2. Netscape 4 possède également son propre objet propriétaire, illustrant le manque d'uniformité dans la gestion des événements à travers les différents navigateurs.

3.Question

Que indique la propriété eventPhase ?

Réponse:La propriété eventPhase indique la phase actuelle de la propagation de l'événement, qui peut être la phase de capture, au niveau de la cible, ou la phase de bulle, aidant les développeurs à comprendre comment les événements se déplacent à travers l'arbre de documents.

4.Question

L'action par défaut d'un événement peut-elle être annulée ? Comment ?

Réponse:Oui, l'action par défaut peut être annulée en utilisant

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

la méthode preventDefault() si l'événement est annulable, ce qui permet aux développeurs de contrôler le comportement des événements sur une page web.

5.Question

Quelles sont les différences de propriétés entre l'objet Event du DOM et le modèle d'événements IE ?

Réponse:Bien que les deux fournissent des fonctionnalités similaires, le modèle d'événements IE inclut des propriétés supplémentaires comme cancelBubble, qui est utilisée pour empêcher la propagation de l'événement, tandis que le modèle DOM utilise des méthodes comme stopPropagation() pour obtenir des effets similaires.

6.Question

Que représente la propriété target dans l'objet Event ?

Réponse:La propriété target représente le nœud de document qui a généré l'événement, permettant ainsi aux gestionnaires d'identifier la source de l'événement.

7.Question

Comment la position des événements de souris est-elle déterminée ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:Les événements de souris sont déterminés par des propriétés telles que clientX et clientY, qui donnent les coordonnées X et Y du pointeur de la souris par rapport à la fenêtre du navigateur.

8.Question

Dans le contexte de la gestion des événements, que signifie 'bubbling' ?

Réponse:Le bubbling est une phase de propagation de l'événement où celui-ci commence à partir de l'élément cible et se propage vers le haut à travers les ancêtres dans la hiérarchie du DOM, permettant aux éléments parents de répondre également à l'événement.

9.Question

Quelle est l'importance de définir des constantes comme Event.CAPTURING_PHASE, Event.AT_TARGET et Event.BUBBLING_PHASE ?

Réponse:Ces constantes fournissent un moyen standardisé de se référer à des phases spécifiques de la propagation des événements, aidant les développeurs à gérer et à comprendre le flux des événements plus facilement.

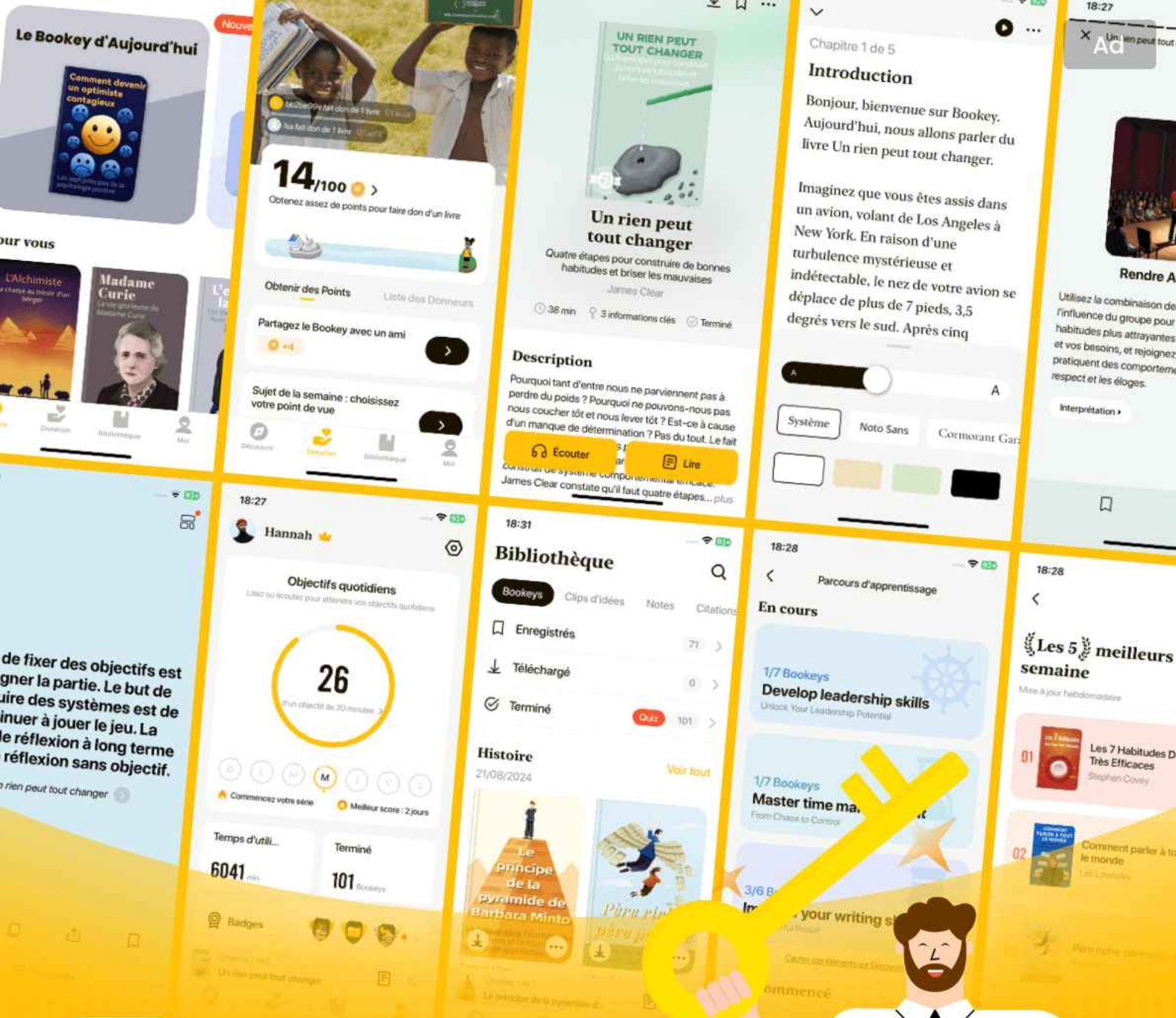
Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

10.Question

Pourquoi l'objet Event de Netscape 4 pourrait-il encore être utile malgré les normes du DOM de niveau 2 ?

Réponse:L'objet Event de Netscape 4 peut encore être utile pour les programmeurs intéressés par les événements clavier, puisque le DOM de niveau 2 ne spécifie pas les événements clavier, offrant ainsi un moyen alternatif de gérer de telles situations.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé



Les meilleures idées du monde débloquent votre potentiel

Essai gratuit avec Bookey

Télécharger dans App Store

DISPONIBLE SUR Google Play

Scanner pour télécharger

Chapitre 22 | Global| Questions et réponses

1.Question

Quelle est l'importance de l'objet Global en JavaScript, et comment est-il lié aux propriétés et méthodes définies par l'utilisateur ?

Réponse:L'objet Global en JavaScript est la pierre angulaire du langage, permettant à toutes les variables et fonctions définies au niveau supérieur de votre code de devenir des propriétés de cet objet. Cela signifie que si vous déclarez une variable dans votre script sans l'encapsuler dans un objet, elle appartient essentiellement à l'objet Global. Par exemple, si vous définissez une fonction `function greet() { return 'Hello, World!'; }`, vous pouvez l'appeler n'importe où en tant que `greet()` puisqu'elle fait intrinsèquement partie de l'objet Global. Ce principe est crucial car il souligne le fonctionnement de la portée en JavaScript ; comprendre l'objet Global aide à prévenir les

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

conflits de noms et permet aux développeurs d'utiliser les ressources globales de manière judicieuse.

2.Question

Pouvez-vous expliquer la différence entre les fonctions encodeURI() et encodeURIComponent() ? Pourquoi est-il important de comprendre leurs différences ?

Réponse:La fonction `encodeURI()` est utilisée pour encoder une URI complète (Identificateur de Ressource Uniforme) mais n'encode pas les caractères ayant des significations spéciales dans une URI, comme `#`, `?`, et `&`. D'autre part, `encodeURIComponent()` encode les caractères qui peuvent poser des problèmes dans les composants d'une URI, ce qui en fait un choix approprié pour encoder des composants individuels d'une URI. Par exemple, si vous souhaitez envoyer un paramètre de requête contenant des symboles spéciaux, vous devez utiliser `encodeURIComponent()` pour vous assurer qu'ils sont transmis en toute sécurité.

Comprendre ces différences est crucial pour éviter des

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

erreurs lors de la construction d'URIs, garantissant que vos applications interprètent et naviguent correctement entre les URLs.

3.Question

Quel rôle joue la fonction eval() en JavaScript, et quelles sont les implications de son utilisation ?

Réponse:La fonction `eval()` évalue une chaîne de code JavaScript qui lui est passée en paramètre et l'exécute comme si c'était du code réel. Par exemple, appeler `eval('2 + 2')` renverrait `4`. Cependant, il est important d'utiliser `eval()` avec prudence, car cela peut introduire des risques de sécurité, notamment si des chaînes fournies par l'utilisateur sont exécutées, ce qui peut entraîner des attaques par injection de code. De plus, cela peut rendre le débogage difficile et avoir un impact sur les performances. Par conséquent, il est généralement déconseillé de s'appuyer sur `eval()` au profit de pratiques de codage plus sûres et plus maintenables.

4.Question

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Comment des fonctions comme isNaN() et isFinite() contribuent-elles à la gestion des valeurs numériques en JavaScript ?

Réponse:Les fonctions `isNaN()` et `isFinite()` offrent un moyen de valider les valeurs numériques en JavaScript. `isNaN()` vérifie si une valeur donnée est la valeur spéciale Not-a-Number (NaN), ce qui peut aider à garantir que les opérations destinées aux variables numériques ne produisent pas d'erreurs. À l'inverse, `isFinite()` vérifie si une valeur est un nombre fini, ce qui est crucial pour gérer efficacement les limites numériques dans les applications, empêchant des problèmes tels que le débordement arithmétique ou un comportement inattendu des valeurs infinies. Ces fonctions permettent aux développeurs de gérer les entrées numériques de manière robuste, améliorant la fiabilité de leurs applications.

5.Question

Quel était le but des fonctions escape() et unescape(), et pourquoi devraient-elles être dépréciées au profit de

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

fonctions plus récentes ?

Réponse:Les fonctions `escape()` et `unescape()` étaient à l'origine utilisées pour encoder et décoder des chaînes, en remplaçant les caractères spéciaux par des représentations hexadécimales. Cependant, ces fonctions ont été dépréciées dans ECMAScript 3 en raison de leur fonctionnalité limitée et de leur potentiel à mal interpréter certains caractères. Des alternatives modernes comme `encodeURI()`, `encodeURIComponent()`, `decodeURI()`, et `decodeURIComponent()` offrent des capacités d'encodage et de décodage plus précises et complètes, les rendant plus sûres et plus efficaces pour travailler avec des URIs et garantir que les chaînes sont correctement formatées pour être transmises sur le web.

Chapitre 23 | Input| Questions et réponses

1.Question

Quelle est l'importance de la propriété 'checked' dans l'objet Input, et comment peut-elle améliorer l'expérience utilisateur dans les formulaires ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:La propriété 'checked' permet aux développeurs de déterminer si une case à cocher ou un bouton radio est sélectionné par l'utilisateur. Cela est crucial pour les formulaires nécessitant des sélections, telles que des accords ou des préférences. Définir cette propriété peut améliorer l'expérience utilisateur en mémorisant les choix à travers les sessions ou en contrôlant l'état des composants de l'interface utilisateur selon les interactions des utilisateurs. Par exemple, si un formulaire est en cours de remplissage et qu'un utilisateur revisite une section, cocher à l'avance les options pertinentes peut faire gagner du temps et réduire la frustration.

2.Question

Comment la propriété 'defaultValue' joue-t-elle un rôle dans l'utilisabilité des formulaires, et pourquoi est-elle importante pour une bonne expérience utilisateur ?

Réponse:La propriété 'defaultValue' définit le texte qui apparaît dans les champs de texte ou de mot de passe lorsque

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

le formulaire est chargé pour la première fois. Cela est important car cela fournit aux utilisateurs des exemples d'entrée attendue, guidant visuellement leurs saisies. Par exemple, placer un espace réservé comme 'Entrez votre email' aide les utilisateurs à comprendre le format requis. Une bonne utilisabilité diminue les taux d'erreur et améliore la satisfaction, rendant les formulaires moins intimidants.

3.Question

Expliquez l'importance de la propriété 'value' dans divers types d'entrées et comment elle affecte la soumission des données dans les applications web.

Réponse:La propriété 'value' contient les données qui seront transmises lors de la soumission d'un formulaire. Pour les champs de texte et de mot de passe, elle reflète le texte saisi par l'utilisateur, la rendant essentielle pour collecter les données utilisateur de manière précise. Pour les boutons, elle affiche l'étiquette, guidant les utilisateurs sur des actions telles que 'Soumettre' ou 'Réinitialiser'. Assurer la capture de la bonne valeur aide à maintenir l'intégrité des données et

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

améliore la fonctionnalité globale de l'application.

4.Question

Quels avantages les méthodes 'focus()' et 'blur()' offrent-elles pour les éléments de formulaire, notamment en termes d'accessibilité ?

Réponse:Les méthodes 'focus()' et 'blur()' sont essentielles pour améliorer l'accessibilité des formulaires. 'focus()' permet aux développeurs de diriger de manière programmatique l'attention de l'utilisateur vers des champs de saisie spécifiques, ce qui peut grandement aider les personnes utilisant la navigation au clavier ou des technologies d'assistance. Inversement, 'blur()' aide à gérer le focus du clavier, contribuant à une expérience utilisateur plus fluide. Ensemble, elles aident à créer une atmosphère plus interactive et conviviale, rendant les formulaires accessibles à un public plus large.

5.Question

Pourquoi est-il crucial pour les développeurs travaillant avec JavaScript côté client de comprendre les gestionnaires d'événements comme 'onchange' et 'onclick'

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

?

Réponse:Comprendre les gestionnaires d'événements tels que 'onchange' et 'onclick' est essentiel pour les développeurs car ils permettent l'interactivité et la réactivité dans les formulaires web. 'onchange' permet à l'application de réagir lorsque les utilisateurs terminent leur saisie, déclenchant la validation ou la logique de traitement. 'onclick' permet d'exécuter des actions lors des clics sur les boutons, ce qui est crucial pour déterminer les intentions des utilisateurs. La maîtrise de ces événements favorise une expérience utilisateur dynamique, rendant les applications plus réactives et intuitives, conduisant finalement à une plus grande satisfaction des utilisateurs.

Chapitre 24 | Calque| Questions et réponses

1.Question

Quelle est l'importance de l'objet Layer dans Netscape 4 et pourquoi est-il considéré comme non standard ?

Réponse:L'objet Layer était important car il offrait un moyen de travailler avec des éléments positionnés

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

dynamiquement en utilisant JavaScript dans Netscape 4, qui manquait à l'époque d'une norme robuste pour la gestion du positionnement. Cependant, il est considéré comme non standard car cette fonctionnalité n'a pas été adoptée dans les versions ultérieures de Netscape, comme Netscape 6, et ne respecte pas les pratiques modernes de développement web. Son utilisation est largement historique, illustrant l'évolution des technologies web.

2.Question

Comment l'objet Layer gère-t-il la visibilité et le positionnement sur une page web ?

Réponse:L'objet Layer permet aux développeurs de contrôler la visibilité des couches grâce à des propriétés telles que 'hidden' et 'visibility', permettant ainsi de montrer ou cacher dynamiquement des sections d'une page web. De plus, des propriétés comme 'left', 'top' et 'zIndex' définissent la position de la couche dans le document, permettant un contrôle précis

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

sur la mise en page et le superposition. Par exemple, définir un 'zIndex' plus élevé sur une couche garantit qu'elle apparaît au-dessus d'une autre, ce qui est crucial pour créer des présentations web interactives.

3.Question

Pouvez-vous expliquer comment fonctionnent les méthodes moveAbove et moveBelow ?

Réponse:Les méthodes 'moveAbove(autre_couche)' et 'moveBelow(autre_couche)' sont utilisées pour modifier l'ordre de superposition des objets Layer. Lorsque vous souhaitez qu'une couche intercepte visuellement une autre (comme faire apparaître un pop-up au-dessus d'un fond), vous appeleriez 'moveAbove' sur cette couche, en passant l'autre couche comme argument. À l'inverse, 'moveBelow' vous permet de placer une couche derrière une autre, contrôlant ainsi quel contenu est vu en premier par l'utilisateur. C'est un aspect fondamental du superposition dans le design web, améliorant l'expérience utilisateur.

4.Question

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Quel est le but des propriétés 'clip' dans l'objet Layer ?
Réponse:Les propriétés 'clip' définissent la zone visible d'un objet Layer. En définissant les dimensions et les coordonnées de la zone de clipping (comme clip.top, clip.bottom, clip.width et clip.height), les développeurs peuvent créer des effets où seule une portion de la couche est affichée tout en masquant le reste. Cela est particulièrement utile dans des scénarios où du contenu se chevauche, car cela permet de personnaliser les parties de la couche visibles pour les utilisateurs.

5.Question

En quoi l'objet Layer dans Netscape 4 diffère-t-il des normes web modernes comme HTML5 et CSS ?
Réponse:L'objet Layer est spécifique à Netscape 4 et a été rendu obsolète par les normes CSS modernes qui utilisent des styles comme 'position: absolute;' pour des objectifs similaires de manière cohérente à travers tous les navigateurs. Contrairement à l'objet Layer qui était peu fiable en raison de sa non-standardisation, le CSS fournit un cadre

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

bien établi pour le positionnement et la visibilité qui est universellement accepté et fonctionnel dans le design web contemporain. Cette évolution illustre un passage vers des pratiques standardisées dans le développement web.

6.Question

Que doivent considérer les développeurs lorsqu'ils travaillent avec des fonctionnalités non standard comme l'objet Layer ?

Réponse:Les développeurs doivent être prudents lorsqu'ils utilisent des fonctionnalités non standard comme l'objet Layer, car elles peuvent ne pas fonctionner dans tous les navigateurs ou versions futures. Il est essentiel de donner la priorité à la compatibilité et à la réactivité ; utiliser des fonctionnalités standard HTML5 et CSS à la place assure une accessibilité plus large et la longévité des applications web. De plus, la dépendance à des technologies obsolètes pourrait entraîner des problèmes de maintenance et une mauvaise expérience utilisateur.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Ad

Scanner pour télécharger

Essayez l'appli Bookey pour lire plus de 1000 résumés des meilleurs livres du monde

Débloquez 1000+ titres, 80+ sujets

Nouveaux titres ajoutés chaque semaine

Brand | Leadership & collaboration | Gestion du temps | Relations & communication | Know

tégie d'entreprise | Créativité | Mémoires | Argent & investissements | Positive Psychology

Entrepreneuriat | Histoire du monde | Communication parent-enfant | Soins Personnels

Aperçus des meilleurs livres du monde

Essai gratuit avec Bookey

Chapitre 25 | Lien| Questions et réponses

1.Question

Quelles sont les principales propriétés d'un objet Link en JavaScript et pourquoi sont-elles importantes ?

Réponse:Les principales propriétés d'un objet Link incluent hash, host, hostname, href, pathname, port, protocol, search et target. Ces propriétés sont importantes car elles permettent aux développeurs de manipuler et de comprendre divers composants d'une URL, facilitant un comportement dynamique dans les applications web, comme la navigation, les recherches et les interactions au sein de différents cadres ou fenêtres. Par exemple, en modifiant le `href`, les développeurs peuvent rediriger les utilisateurs vers différentes pages, améliorant ainsi l'expérience utilisateur.

2.Question

Comment les gestionnaires d'événements JavaScript pour les liens peuvent-ils améliorer l'interaction des utilisateurs ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:Les gestionnaires d'événements comme onclick, onmouseover et onmouseout offrent une manière de répondre aux actions des utilisateurs sur les liens, renforçant l'interactivité. Par exemple, avec le gestionnaire d'événements onclick, les développeurs peuvent exécuter un code JavaScript supplémentaire lorsqu'un utilisateur clique sur un lien, comme le suivi des clics pour des analyses ou la prévention de la navigation en retournant false. Cela permet d'enrichir les expériences utilisateur et de contrôler le comportement des liens en fonction du contexte.

3.Question

Pourquoi utiliseriez-vous la propriété target d'un objet Link ?

Réponse:La propriété target est utilisée pour spécifier comment un document lié doit être affiché par rapport à la fenêtre ou au cadre actuel. L'utilisation de valeurs telles que '_blank' ouvre le lien dans un nouvel onglet, tandis que '_self' l'ouvre dans le même cadre. Cela permet aux développeurs de créer une expérience de navigation contrôlée, facilitant une

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

meilleure navigation sans perdre la page actuelle.

4.Question

Pouvez-vous expliquer l'importance de la propriété search dans une URL ?

Réponse:La propriété search, qui inclut la chaîne de requête d'une URL, est significative car elle se rapporte à la manière dont les données sont envoyées au serveur pour traitement. Par exemple, dans une application de recherche, des paramètres comme `?q=JavaScript&m=10` aident à transmettre les saisies ou préférences des utilisateurs au serveur, permettant des réponses et résultats personnalisés. Cela souligne l'importance de comprendre les composants d'une URL pour un développement web efficace.

5.Question

Comment la compréhension des propriétés d'un objet Link contribue-t-elle à de bonnes pratiques de développement web ?

Réponse:Comprendre les propriétés de l'objet Link permet aux développeurs de créer des applications web dynamiques et robustes qui interagissent de manière fluide avec les

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

utilisateurs. Par exemple, en manipulant les propriétés href ou search, les développeurs peuvent changer dynamiquement le contenu sans avoir besoin de recharger la page. Cela enrichit l'expérience utilisateur et augmente l'engagement, montrant comment une compréhension approfondie de JavaScript peut conduire à de meilleures fonctionnalités et satisfaction dans les projets web.

Chapitre 26 | Math| Questions et réponses

1.Question

Quel est le but de l'objet Math en JavaScript ?

Réponse:L'objet Math sert de recueil de constantes et de fonctions mathématiques. Contrairement à d'autres objets comme Date et String, il ne représente pas une classe d'objets et ne peut pas être instancié. Au lieu de cela, il fournit des méthodes statiques qui effectuent diverses opérations mathématiques.

2.Question

Quelle est l'importance de la constante Math.PI ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:Math.PI est une constante mathématique essentielle représentant le rapport de la circonférence d'un cercle à son diamètre, soit environ 3,14159. Elle est largement utilisée dans les calculs impliquant des cercles, la trigonométrie et la géométrie.

3.Question

Comment la fonction Math.abs améliore-t-elle la gestion des données ?

Réponse:La fonction Math.abs retourne la valeur absolue d'un nombre donné, permettant aux développeurs d'ignorer facilement le signe d'un nombre. Cela est utile dans des scénarios où seule la magnitude d'un nombre est importante, comme dans les calculs de distances.

4.Question

Pouvez-vous expliquer la différence entre Math.max et Math.min ?

Réponse:Math.max retourne le plus grand des arguments fournis, tandis que Math.min retourne le plus petit. Cette fonctionnalité est essentielle pour comparer des valeurs

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

numériques et déterminer les extrêmes dans des ensembles de données.

5.Question

Comment fonctionne la fonction Math.random et quand pourrait-elle être utilisée ?

Réponse:Math.random génère un nombre réel pseudo-aléatoire compris entre 0 (inclus) et 1 (exclu). Cette fonction est particulièrement utile dans des scénarios impliquant le hasard, comme la création d'éléments de jeu aléatoires, le brassage de données ou la sélection d'utilisateurs au hasard.

6.Question

Quel est l'avantage d'utiliser Math.pow ?

Réponse:Math.pow permet l'exponentiation, ce qui signifie qu'il peut calculer n'importe quel nombre élevé à n'importe quelle puissance. Cette opération est fondamentale dans de nombreux calculs mathématiques, des carrés simples aux calculs de puissances plus complexes qui apparaissent dans des domaines comme la physique et l'ingénierie.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

7.Question

Quelle est l'utilisation pratique des fonctions Math.ceil et Math.floor ?

Réponse:Math.ceil arrondit un nombre à l'entier supérieur le plus proche, tandis que Math.floor arrondit à l'entier inférieur le plus proche. Cela peut être utile dans des applications nécessitant des valeurs entières nettes, comme dans la pagination, où il ne peut pas y avoir de pages fractionnaires.

8.Question

Comment la compréhension des fonctions mathématiques en JavaScript améliore-t-elle les compétences en programmation ?

Réponse:Comprendre les fonctions mathématiques permet aux développeurs JavaScript de manipuler efficacement les nombres, de résoudre des problèmes complexes et d'innover des solutions numériques, favorisant ainsi la créativité et la logique dans le développement de logiciels.

Chapitre 27 | Navigateur| Questions et réponses

1.Question

Quel est le but de la propriété `appCodeName` dans

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Navigator ?

Réponse:La propriété `appCodeName` fournit un surnom pour le navigateur, garantissant une compatibilité entre les différents navigateurs. Par exemple, c'est 'Mozilla' dans tous les navigateurs Netscape et aussi dans les navigateurs Microsoft pour la cohérence.

2.Question

Comment la propriété `appVersion` peut-elle être utile dans le développement JavaScript ?

Réponse:La propriété `appVersion` donne des informations détaillées sur la version du navigateur et la plateforme sur laquelle il fonctionne. En utilisant des fonctions comme `parseInt()` et `parseFloat()`, les développeurs peuvent facilement extraire les numéros de version majeurs et mineurs, ce qui peut aider à adapter le comportement des applications web à des versions spécifiques des navigateurs.

3.Question

Quelle sorte d'informations la propriété `cookieEnabled`

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

fournit-elle, et pourquoi est-ce important ?
Réponse:La propriété `cookieEnabled` indique si le navigateur a les cookies activés. Cela est crucial pour les développeurs lorsqu'ils mettent en œuvre des fonctionnalités qui reposent sur les cookies pour la gestion des sessions, les préférences utilisateur ou le suivi.

4.Question

Dans quels scénarios les propriétés `language` et `userLanguage` pourraient-elles être particulièrement utiles ?
Réponse:La propriété `language` peut aider les développeurs à personnaliser le contenu en fonction de la langue par défaut du navigateur de l'utilisateur, tandis que `userLanguage` indique la langue préférée de l'utilisateur. Cela est particulièrement bénéfique pour créer des applications web localisées qui améliorent l'expérience utilisateur.

5.Question

Pourquoi est-il significatif que la propriété `platform` puisse avoir des valeurs variées ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:Les valeurs variées de la propriété `platform` rappellent aux développeurs que leurs applications web doivent s'adapter à une gamme diversifiée de systèmes d'exploitation et de configurations matérielles. Comprendre ces variations peut influencer les décisions de conception et les stratégies de test pour garantir la compatibilité entre les plateformes.

6.Question

Que signifie la méthode `javaEnabled()` pour un navigateur, et comment cela peut-il affecter la fonctionnalité web ?

Réponse:La méthode `javaEnabled()` vérifie si Java est pris en charge dans le navigateur actuel. Cela est significatif, car certaines applications web ou applets dépendent de Java ; savoir si c'est activé peut déterminer s'il faut charger ces composants.

7.Question

Comment les propriétés et méthodes de Navigator améliorent-elles les pratiques de développement web ?

Réponse:Ces propriétés et méthodes donnent aux

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

développeurs un aperçu de l'environnement du client, leur permettant de créer des applications plus robustes et conviviales. Ils peuvent optimiser les performances, garantir la compatibilité et améliorer l'interaction utilisateur en fonction des capacités spécifiques des navigateurs.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Ad

Scanner pour télécharger

Pourquoi Bookey est une application incontournable pour les amateurs de livres

Contenu de 30min
Plus notre interprétation est profonde et claire, mieux vous saisissez chaque titre.

Format texte et audio
Absorbez des connaissances même dans un temps fragmenté.

Quiz
Vérifiez si vous avez maîtrisé ce que vous venez d'apprendre.

Et plus
Plusieurs voix & polices, Carte mentale, Citations, Clips d'idées...

Essai gratuit avec Bookey

Chapitre 28 | Node| Questions et réponses

1.Question

Quels sont les différents types de nœuds définis dans le Node DOM Level 1, et comment la propriété nodeType les classe-t-elle ?

Réponse:Les types de nœuds dans le Node DOM Level 1 incluent le Nœud Élément (nodeType = 1), le Nœud Attribut (nodeType = 2), le Nœud Texte (nodeType = 3), le Nœud Commentaire (nodeType = 8), le Nœud Document (nodeType = 9), et le Nœud DocumentFragment (nodeType = 11). La propriété nodeType nous permet d'identifier à quelle sous-classe appartient une instance de Nœud, ce qui clarifie la manière de l'interagir.

2.Question

Quelle est l'importance de la propriété attributes dans les nœuds Élément, et comment est-elle généralement utilisée ?

Réponse:La propriété attributes d'un nœud Élément est un tableau en lecture seule contenant les objets Attr représentant

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

les attributs de l'élément. Bien qu'elle puisse être accédée par numéro ou nom d'attribut, il est moins courant d'utiliser ce tableau puisque tous les attributs HTML ont également des propriétés correspondantes sur l'objet Élément, ce qui rend l'accès direct généralement plus simple.

3.Question

Expliquez l'objectif de la méthode addEventListener et ses paramètres. Comment cela se rapporte-t-il à la gestion des événements dans le DOM ?

Réponse:La méthode addEventListener est utilisée pour enregistrer un écouteur d'événements pour un nœud. Ses paramètres incluent type (une chaîne qui spécifie le type d'événement, comme 'click'), listener (la fonction de gestion de l'événement qui sera déclenchée) et useCapture (un booléen indiquant la phase de propagation de l'événement). Cela est essentiel pour gérer les interactions utilisateur et rendre l'application web réactive aux événements.

4.Question

Comment fonctionne la méthode insertBefore, et quel rôle joue-t-elle dans la manipulation de l'arbre DOM ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:La méthode insertBefore permet d'ajouter un nœud newChild dans l'arbre du document juste avant un nœud refChild spécifié qui est déjà enfant de son parent. Cette méthode est cruciale pour réorganiser la structure du DOM, car elle peut mettre à jour la représentation visuelle du document de manière dynamique sans avoir besoin de recréer des éléments.

5.Question

Décrivez la méthode cloneNode et son paramètre deep. Pourquoi pourrait-on utiliser cette fonctionnalité ?

Réponse:La méthode cloneNode retourne une copie du nœud spécifié. Le paramètre deep détermine s'il faut également cloner les descendants du nœud, créant ainsi une copie complète de l'arborescence (si vrai). Cela est utile pour dupliquer des éléments, notamment dans des scénarios où l'on souhaite préserver une structure complexe et ses attributs.

6.Question

Quelle est l'importance de la méthode normalize dans le contexte des nœuds Texte ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:La méthode normalize nettoie un nœud en fusionnant les nœuds Texte adjacents et en supprimant les nœuds vides. Cela permet de rendre le contenu plus ordonné, garantissant que la structure du document n'est pas encombrée par des nœuds inutiles, ce qui peut améliorer les performances et rendre une manipulation ultérieure plus efficace.

7.Question

Pouvez-vous décrire la relation entre parentNode et childNodes dans l'arbre DOM ?

Réponse:La propriété parentNode fait référence au parent immédiat du nœud, tandis que childNodes renvoie à la collection de nœuds qui sont des enfants directs d'un nœud. Ces relations illustrent la nature hiérarchique du DOM, permettant aux développeurs de naviguer et de manipuler les éléments en fonction de leur contexte structurel.

8.Question

Quels sont quelques méthodes pour supprimer un nœud du DOM, et comment fonctionnent-elles ?

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:Les méthodes removeChild et removeEventListener servent à gérer les nœuds dans le DOM. removeChild prend un ancien nœud enfant comme argument et le supprime de son parent. Cette opération aide à modifier dynamiquement le contenu d'une page web. removeEventListener, en revanche, détache un écouteur d'événements précédemment attaché, permettant de contrôler les interactions d'événements pour un élément.

Chapitre 29 | Nombre| Questions et réponses

1.Question

Quel est le but du constructeur Number en JavaScript ?

Réponse:Le constructeur Number, lorsqu'il est appelé avec l'opérateur 'new', crée un nouvel objet Number et convertit son argument en une valeur numérique. S'il est utilisé sans 'new', il sert de fonction de conversion qui convertit directement son argument en un nombre.

2.Question

Quelles sont les constantes clés définies dans l'objet

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Number ?

Réponse:Les constantes clés incluent : Number.MAX_VALUE (le plus grand nombre représentable, ~1.79E+308), Number.MIN_VALUE (le plus petit nombre représentable, ~5E-324), Number.NaN (valeur not-a-number), Number.NEGATIVE_INFINITY (valeur d'infini négatif) et Number.POSITIVE_INFINITY (valeur d'infini positif). Ces constantes fournissent des limites importantes et des valeurs particulières pour les calculs numériques.

3.Question

Comment fonctionne la méthode toExponential en JavaScript ?

Réponse:La méthode toExponential convertit un nombre en une chaîne de caractères en notation exponentielle, en s'assurant qu'il y a un chiffre avant le point décimal et un nombre spécifié de chiffres après. Le paramètre 'digits' peut être compris entre 0 et 20, et il ajuste le nombre en conséquence en arrondissant ou en complétant avec des zéros

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

si nécessaire.

4.Question

Pouvez-vous expliquer la différence entre les méthodes toFixed et toLocaleString ?

Réponse:La méthode toFixed renvoie une représentation sous forme de chaîne d'un nombre avec un nombre fixe de décimales, défini par le paramètre 'digits'. En revanche, toLocaleString formatte le nombre selon les conventions locales, ce qui peut inclure différents symboles pour les virgules décimales et les séparateurs de milliers, le rendant plus adapté aux affichages conviviaux.

5.Question

Quelle est la signification de Number.NaN en JavaScript ?

Réponse:Number.NaN représente une valeur 'not-a-number', qui est essentielle pour la gestion des erreurs et la validation des calculs numériques en JavaScript. Elle indique qu'une valeur n'est pas un nombre valide, ce qui est important pour garantir l'intégrité des opérations mathématiques.

6.Question

Comment convertir un nombre en chaîne dans une base

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

différente en JavaScript ?

Réponse:Vous pouvez utiliser la méthode toString de l'objet Number, en spécifiant la base (ou le radix). Le radix peut être n'importe quel entier entre 2 et 36, vous permettant de convertir le nombre en binaire, décimal, hexadécimal ou d'autres systèmes numériques. Si aucun radix n'est fourni, il par défaut à la base 10.

7.Question

Quelles sont les implications de l'utilisation des constantes Number comme Number.MAX_VALUE et Number.MIN_VALUE dans votre code ?

Réponse:Utiliser des constantes comme Number.MAX_VALUE et Number.MIN_VALUE aide les développeurs à définir des limites sur les calculs numériques, à éviter les erreurs de dépassement ou de sous-dépassement, et à écrire un code plus robuste et résistant aux erreurs. Ces constantes offrent une protection contre le dépassement des capacités numériques de JavaScript.

8.Question

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Dans quels scénarios utiliserait-on la méthode toPrecision ?

Réponse:La méthode toPrecision est particulièrement utile lorsque vous devez présenter des nombres avec un nombre spécifique de chiffres significatifs, par exemple dans des rapports scientifiques, des états financiers, ou chaque fois qu'un contrôle précis sur la représentation des données numériques est nécessaire.

Chapitre 30 | Objet| Questions et réponses

1.Question

Quelle est l'importance du constructeur Object en JavaScript ?

Réponse:Le constructeur Object est fondamental en JavaScript car il crée un objet vide auquel des propriétés arbitraires peuvent être ajoutées. Cette flexibilité est cruciale pour la programmation dynamique, permettant aux développeurs de créer des structures capables de contenir divers types de données et propriétés selon les besoins.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

2.Question

Comment les méthodes 'hasOwnProperty', 'isPrototypeOf' et 'propertyIsEnumerable' améliorent-elles les interactions avec les objets en JavaScript ?

Réponse:Ces méthodes fournissent des outils essentiels pour gérer les propriétés des objets et l'héritage en JavaScript. 'hasOwnProperty' permet de vérifier si une propriété appartient directement à un objet, garantissant l'intégrité des données. 'isPrototypeOf' aide à comprendre les chaînes de prototypes, facilitant ainsi l'héritage. 'propertyIsEnumerable' offre un aperçu de la visibilité des propriétés lors de l'énumération, ce qui est vital pour les processus d'itération, comme les boucles.

3.Question

Comment 'toString()' diffère-t-il selon les types d'objets en JavaScript ?

Réponse:Bien que la méthode 'toString()' de base dans la classe Object offre une représentation générique, différents types d'objets comme Number ou Boolean remplacent

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

généralement cette méthode pour fournir une sortie plus significative. Par exemple, appeler 'toString()' sur un Number retourne le nombre sous forme de chaîne, ce qui est plus utile pour les développeurs cherchant des sorties spécifiques.

4.Question

Pourquoi une sous-classe pourrait-elle remplacer la méthode 'toLocaleString()' ?

Réponse:Une sous-classe pourrait remplacer 'toLocaleString()' pour fournir une représentation en chaîne spécifique à la locale, améliorant ainsi l'utilisabilité pour l'internationalisation. Par exemple, un objet Date pourrait formater les dates différemment en fonction des paramètres de locale de l'utilisateur, garantissant que les applications peuvent répondre efficacement à des groupes d'utilisateurs divers.

5.Question

Quel rôle joue la méthode 'valueOf()' dans les objets JavaScript ?

Réponse:La méthode 'valueOf()' est conçue pour retourner la

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

valeur primitive associée à un objet lorsque nécessaire, par exemple lors d'opérations nécessitant un type non-objet. Par exemple, lorsqu'un objet est utilisé dans un contexte mathématique, invoquer 'valueOf()' permet à l'opération de le convertir de manière transparente au type primitif pertinent.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Ad

App Store
Coup de cœur

22k avis 5 étoiles

Scanner pour télécharger

Retour Positif

Fabienne Moreau

ue résumé de livre ne testent
ion, mais rendent également
musant et engageant.
té la lecture pour moi.

Fantastique!

Giselle Dubois

Je suis émerveillé par la variété de livres et de langues que Bookey supporte. Ce n'est pas juste une application, c'est une porte d'accès au savoir mondial. De plus, gagner des points pour la charité est un grand plus !

Fi

Le
liv
co
pr

é Blanchet

de lecture
eption de
es,
cous.

J'adore !

Isoline Mercier

Bookey m'offre le temps de parcourir les parties importantes d'un livre. Cela me donne aussi une idée suffisante pour savoir si je devrais acheter ou non la version complète du livre ! C'est facile à utiliser !"

Gain de temps !

Bookey est mon applicat
intellectuelle. Les résum
magnifiquement organis
monde de connaissance

ppli géniale !

Joachim Lefevre

adore les livres audio mais je n'ai pas toujours le temps
'écouter le livre entier ! Bookey me permet d'obtenir
n résumé des points forts du livre qui m'intéresse !!!
Quel super concept !!! Hautement recommandé !

Appli magnifique

Cette application est une bouée de sauve
amateurs de livres avec des emplois du te
Les résumés sont précis, et les cartes mer
renforcer ce que j'ai appris. Hautement re

Essai gratuit avec Bookey

Chapitre 31 | RegExp| Questions et réponses

1.Question

Quel est l'objectif d'utiliser des expressions régulières en JavaScript ?

Réponse:Les expressions régulières servent à la correspondance de motifs dans les chaînes de caractères, ce qui permet de rechercher, remplacer et manipuler du texte en fonction de motifs spécifiés. Cela peut être particulièrement utile pour des tâches comme la validation des saisies utilisateur, l'analyse de fichiers texte ou la mise en œuvre de fonctionnalités de recherche.

2.Question

Comment la propriété 'global' d'un RegExp influence-t-elle le processus de correspondance ?

Réponse:La propriété 'global', lorsqu'elle est définie sur true, indique que l'expression régulière cherchera toutes les correspondances dans la chaîne au lieu de s'arrêter après la première correspondance. Cela signifie que si vous utilisez le

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

drapeau global (g) dans votre RegExp, des méthodes comme exec() renverront toutes les correspondances trouvées.

3.Question

Que fait la propriété 'ignoreCase' dans le contexte des expressions régulières ?

Réponse:La propriété 'ignoreCase' permet une correspondance insensible à la casse. Cela signifie que si un RegExp est créé avec l'attribut 'i', il traitera les lettres majuscules et minuscules comme équivalentes lors des correspondances, ce qui facilite l'appariement de chaînes quelle que soit leur casse.

4.Question

Pouvez-vous expliquer la signification de la propriété 'lastIndex' dans un RegExp global ?

Réponse:La propriété 'lastIndex' est cruciale pour les objets RegExp globaux car elle garde une trace de la position dans la chaîne où la prochaine correspondance commencera. Après avoir appelé exec() ou test() sur un RegExp global, cette propriété est mise à jour pour pointer vers le caractère

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

juste après la dernière correspondance, permettant une itération correcte à travers la chaîne sans manquer de correspondances.

5.Question

Quand utiliseriez-vous la propriété 'multiline' d'un RegExp ?

Réponse:La propriété 'multiline' est utile lorsqu'on travaille avec des chaînes contenant des caractères de nouvelle ligne. En définissant l'attribut 'm' dans votre RegExp, la correspondance de motif traitera chaque ligne comme une chaîne distincte, ce qui affecte le comportement des ancres de début (^) et de fin ($) pour correspondre à la fois au début et à la fin de chaque ligne, plutôt qu'à celui de l'ensemble de la chaîne.

6.Question

Quel est le résultat de l'utilisation de la méthode 'exec()' sur une chaîne avec un RegExp, et comment pouvez-vous interpréter ses résultats ?

Réponse:Lorsque la méthode 'exec()' est utilisée sur une chaîne avec un RegExp, elle renvoie un tableau qui inclut le

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

texte entièrement apparié comme premier élément, ainsi que tous les groupes capturés comme éléments suivants. Si aucune correspondance n'est trouvée, elle renvoie null. Ce tableau inclut également une propriété 'index' qui indique où dans la chaîne d'origine la correspondance a été trouvée, ce qui facilite la localisation de la correspondance dans la chaîne.

7.Question

Quelles sont les différences entre 'test()' et 'exec()' lors de l'utilisation de RegExp ?

Réponse:La méthode 'test()' renvoie un booléen indiquant si le RegExp correspond à une partie de la chaîne, ce qui en fait un moyen rapide de vérifier une correspondance. En revanche, 'exec()' fournit un tableau de correspondance détaillé incluant les groupes capturés, rendant l'information plus complète mais aussi plus complexe. Utilisez 'test()' pour des vérifications simples et 'exec()' lorsque vous avez besoin d'informations de correspondance plus détaillées.

Chapitre 32 | Sélection| Questions et réponses

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

1.Question

Quelle est l'importance de la propriété `selectedIndex` dans un objet Select ?

Réponse:La propriété `selectedIndex` est essentielle car elle permet aux développeurs de suivre quelle option l'utilisateur a sélectionnée dans une liste déroulante. Elle renvoie la valeur d'index de l'option actuellement sélectionnée, ce qui peut être utile pour la logique conditionnelle dans les scripts. Par exemple, si un utilisateur sélectionne 'Option 2', `selectedIndex` renverrait 1 (puisque l'indexation commence à zéro), vous permettant d'exécuter un code spécifique ou de manipuler l'interface en fonction du choix de l'utilisateur.

2.Question

Comment ajouter des options à un élément Select en JavaScript ?

Réponse:Pour ajouter des options à un élément Select, vous pouvez utiliser le constructeur `Option()` pour créer une

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

nouvelle option, puis l'ajouter au tableau `options` de l'objet Select. Par exemple :

```javascript
var selectElement = document.getElementById('mySelect');
var newOption = new Option('Texte de la nouvelle option', 'valeurNouvelleOption');
selectElement.options[selectElement.options.length] = newOption;
```

Ce snippet de code ajoute effectivement une nouvelle option à la fin de la liste de sélection, augmentant l'interaction et les choix pour l'utilisateur.

3.Question

Dans quels scénarios utiliseriez-vous la méthode `remove` sur un objet Select ?

Réponse:Vous pourriez généralement utiliser la méthode `remove` lorsque vous devez mettre à jour dynamiquement le contenu d'une liste de sélection en fonction des interactions

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

de l'utilisateur ou de la logique de l'application. Par exemple, si un utilisateur choisit son pays dans une liste et que vous souhaitez filtrez les pays non pertinents en fonction de ce choix, vous pouvez retirer les options non pertinentes de l'élément Select pour améliorer l'expérience utilisateur.

4.Question

Expliquez l'objectif des gestionnaires d'événements comme `onchange` et `onfocus` dans le contexte des éléments Select.

Réponse:Les gestionnaires d'événements tels que `onchange` et `onfocus` sont cruciaux pour améliorer l'interactivité dans les applications web. Le gestionnaire d'événements `onchange` est déclenché lorsque l'utilisateur sélectionne ou modifie son choix, permettant une rétroaction immédiate ou des mises à jour de l'interface en fonction de la sélection effectuée. Cela peut être utilisé pour charger dynamiquement du contenu connexe. Le gestionnaire d'événements `onfocus`, quant à lui, est invoqué lorsque l'utilisateur clique ou navigue vers l'élément Select, souvent utilisé pour mettre en évidence

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

l'élément ou fournir des instructions à l'utilisateur, améliorant ainsi l'aisance d'utilisation.

5.Question

Quelles sont les propriétés en lecture seule de l'objet Select et pourquoi sont-elles importantes ?

Réponse:L'objet Select contient plusieurs propriétés en lecture seule, telles que `form`, `length` et `type`. Ces propriétés fournissent des informations essentielles sur le contexte et la fonctionnalité de l'élément Select. Par exemple, `form` permet d'accéder à l'objet Form contenant le Select, ce qui est utile pour gérer plusieurs éléments de formulaire. `length` donne le compte des options disponibles, ce qui peut aider à valider l'entrée de l'utilisateur ou à ajuster dynamiquement l'interface en fonction du nombre de sélections disponibles. La propriété `type` spécifie si plusieurs sélections sont autorisées, guidant ainsi les développeurs sur la façon de gérer les entrées utilisateur de manière appropriée.

6.Question

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Comment la méthode `add` peut-elle améliorer l'expérience utilisateur dans les formulaires ?

Réponse:La méthode `add` améliore l'expérience utilisateur en permettant aux développeurs d'insérer des options dans les éléments Select à des positions spécifiques. Cela est particulièrement utile dans les scénarios où l'ordre des options est important, par exemple pour s'assurer qu'un élément 'le plus pertinent' apparaît en haut d'une liste en fonction de l'activité ou des préférences de l'utilisateur. En gérant les options disponibles de manière programmatique, les développeurs peuvent créer une interface plus intuitive et efficace qui répond dynamiquement aux besoins des utilisateurs.

Chapitre 33 | Chaîne| Questions et réponses

1.Question

Quelle est l'importance de comprendre comment manipuler les chaînes de caractères en JavaScript ?

Réponse:Comprendre la manipulation des chaînes de caractères en JavaScript est essentiel, car cela

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

permet aux développeurs de gérer et de transformer efficacement les données textuelles. Cela inclut des tâches telles que rechercher des caractères spécifiques, concaténer des chaînes et effectuer des conversions de casse, qui sont essentielles tant dans le développement front-end que backend. Maîtriser ces techniques aide à créer des applications dynamiques et à garantir une meilleure expérience utilisateur.

2.Question

Comment la méthode 'slice' améliore-t-elle la manipulation des chaînes ?

Réponse:La méthode 'slice' est particulièrement utile pour extraire des portions spécifiques d'une chaîne sans modifier la chaîne originale. Elle permet aux développeurs d'obtenir des sous-chaînes en spécifiant des indices de début et de fin, en tenant compte des indices négatifs qui comptent à partir de la fin de la chaîne. Par exemple, en découpant la chaîne 'Hello World!' avec slice(6, 11), on obtient 'World', ce qui est

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

crucial pour des tâches telles que l'extraction de sous-chaînes dans les saisies utilisateur ou le traitement de données.

3.Question

Pourquoi la méthode 'indexOf' est-elle essentielle lors de la manipulation des chaînes ?

Réponse:La méthode 'indexOf' est essentielle car elle fournit un moyen de localiser la première occurrence d'une sous-chaîne dans une chaîne plus grande, en renvoyant sa position. Cela est crucial pour la validation des données, les fonctionnalités de recherche et la correspondance de motifs, permettant ainsi aux développeurs de créer des applications plus intelligentes en permettant des recherches ciblées dans le texte.

4.Question

Dans quels scénarios utiliseriez-vous la méthode 'replace', et comment fonctionne-t-elle ?

Réponse:La méthode 'replace' est inestimable lorsque vous devez mettre à jour ou modifier des parties d'une chaîne en fonction de motifs ou de textes spécifiques. Par exemple, si

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

vous souhaitez changer toutes les occurrences de 'pomme' dans 'J'aime la tarte aux pommes' en 'orange', vous pouvez utiliser replace(/pomme/g, 'orange'), qui utilise une expression régulière pour s'assurer que toutes les occurrences soient modifiées. Cette méthode facilite les mises à jour de contenu dynamiques, comme la désinfection des saisies utilisateur ou la génération de nouveaux textes en fonction des préférences des utilisateurs.

5.Question

Quelles sont les meilleures pratiques pour créer un nouvel objet String et quand devez-vous l'utiliser ?

Réponse:Créer un nouvel objet String en utilisant 'new String(s)' n'est généralement pas recommandé, car cela encapsule la valeur dans un objet plutôt que de créer une chaîne primitive, ce qui peut entraîner un comportement inattendu. La meilleure pratique est d'utiliser des littéraux de chaîne ou le constructeur String sans 'new'. Utilisez-le avec parcimonie lorsque vous avez besoin d'un objet avec des méthodes de chaîne, mais pour la plupart des cas, restez sur

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

des chaînes primitives pour des raisons de performance et de simplicité.

6.Question

Comment les méthodes 'toLowerCase' et 'toUpperCase' contribuent-elles à la cohérence des données ?

Réponse:Les méthodes 'toLowerCase' et 'toUpperCase' sont essentielles pour maintenir la cohérence des données, en particulier dans les scénarios de saisie utilisateur où la sensibilité à la casse peut entraîner des divergences. Par exemple, convertir une saisie utilisateur en minuscules avant le stockage garantit que les comparaisons sont uniformes, évitant des problèmes tels que 'Nom d'utilisateur' contre 'nom d'utilisateur'. Cette pratique améliore non seulement l'intégrité des données, mais améliore également l'expérience utilisateur en évitant les erreurs liées à la casse.

7.Question

Quel rôle jouent les expressions régulières dans les méthodes telles que 'match' et 'search' ?

Réponse:Les expressions régulières (regex) sont des outils

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

puissants utilisés dans les méthodes 'match' et 'search' pour effectuer des correspondances de motifs complexes sur des chaînes. Ces méthodes s'appuient sur les regex pour identifier des séquences spécifiques, valider des formats ou extraire des informations significatives à partir de textes plus volumineux. La maîtrise des regex, combinée à ces méthodes, renforce la capacité d'un développeur à mettre en œuvre des fonctionnalités de recherche sophistiquées, des validations de données et des tâches de traitement de texte.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Ad

Scanner pour télécharger

Lire, Partager, Autonomiser

Terminez votre défi de lecture, faites don de livres aux enfants africains.

Le Concept

BOOKS FOR AFRICA × ×

Cette activité de don de livres se déroule en partenariat avec Books For Africa. Nous lançons ce projet car nous partageons la même conviction que BFA : Pour de nombreux enfants en Afrique, le don de livres est véritablement un don d'espoir.

La Règle

Gagnez 100 points → **Échangez un livre** → **Faites un don à l'Afrique**

Votre apprentissage ne vous apporte pas seulement des connaissances mais vous permet également de gagner des points pour des causes caritatives ! Pour chaque 100 points gagnés, un livre sera donné à l'Afrique.

Essai gratuit avec Bookey

Chapitre 34 | Style| Questions et réponses

1.Question

Quelle est la relation entre les propriétés CSS et les propriétés de style du DOM en JavaScript ?

Réponse:Les propriétés de l'objet Style en JavaScript reflètent étroitement celles des attributs CSS, avec certaines conventions de nommage adaptées à la syntaxe de JavaScript. Par exemple, des propriétés CSS comme 'font-family' sont converties en 'fontFamily' dans JavaScript, sans tirets et avec une majuscule pour les mots suivants. Cela permet aux développeurs de manipuler les styles CSS directement via le DOM de JavaScript.

2.Question

Quels sont les éléments à considérer lors du travail avec des propriétés de style ayant des valeurs numériques ?

Réponse:Lorsqu'il s'agit de propriétés CSS ayant des valeurs numériques, il est crucial de convertir entre chaînes de caractères et nombres. Utilisez parseFloat() pour récupérer la

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

valeur numérique d'une chaîne, et lorsque vous définissez une propriété, assurez-vous de reconvertir le nombre en chaîne tout en incluant les unités nécessaires (comme 'px'); par exemple, définir la largeur d'un élément pourrait ressembler à 'element.style.width = "100px".'

3.Question

Pourquoi certaines propriétés CSS pourraient-elles ne pas être prises en charge dans tous les navigateurs ?

Réponse:Les différents navigateurs peuvent avoir des niveaux de prise en charge variables pour les propriétés CSS, ce qui signifie que même si les propriétés de style sont définies au sein de l'objet Style, leur comportement peut différer. C'est pourquoi les développeurs doivent toujours vérifier la compatibilité entre les navigateurs et consulter des ressources CSS externes pour des informations fiables.

4.Question

Comment le modèle d'objet en JavaScript aide-t-il à styliser les pages web de manière dynamique ?

Réponse:Le modèle d'objet JavaScript permet une

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

manipulation en temps réel des styles CSS, ce qui signifie que les développeurs peuvent changer l'apparence des éléments web de manière dynamique en fonction des interactions de l'utilisateur ou d'autres conditions. Cela est fondamental pour créer des designs réactifs et des applications web interactives, améliorant l'expérience utilisateur en permettant un retour visuel immédiat.

5.Question

Quelle est l'importance de la propriété 'cssFloat' en JavaScript ?

Réponse:La propriété 'cssFloat' est une adaptation au sein de l'objet Style en JavaScript pour gérer les attributs float CSS, puisque 'float' est un mot réservé en JavaScript. Cela souligne l'importance d'adapter les propriétés CSS à la syntaxe de JavaScript et d'assurer que les développeurs puissent toujours contrôler efficacement les propriétés de mise en page via le scripting.

6.Question

Comment les développeurs peuvent-ils s'assurer qu'ils font référence avec précision aux styles CSS lorsqu'ils

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

utilisent JavaScript ?

Réponse:Les développeurs devraient se référer à des références CSS à jour pour comprendre les propriétés disponibles et leur syntaxe. Ils doivent également prendre en compte les règles qui régissent la conversion des propriétés CSS en propriétés JavaScript, en s'assurant de suivre les conventions de nommage appropriées pour éviter les erreurs dans leur code.

Chapitre 35 | Window| Questions et réponses

1.Question

Quel est le principal objectif de l'objet Window dans JavaScript côté client ?

Réponse:L'objet Window agit comme l'objet global pour tout JavaScript côté client, fournissant des propriétés et des méthodes qui facilitent l'interaction avec le navigateur et permettent aux développeurs de créer un contenu dynamique.

2.Question

Comment la fonction 'alert' illustre-t-elle l'interactivité de

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

JavaScript ?

Réponse:La fonction 'alert' affiche une boîte de dialogue avec un message, mettant en pause l'exécution du script jusqu'à ce que l'utilisateur ferme la boîte. Cela exemplifie la capacité de JavaScript à créer des expériences interactives dans les applications web.

3.Question

Quelles sont les comportements particuliers associés à la propriété 'location' de l'objet Window ?

Réponse:La propriété 'location' peut à la fois récupérer l'URL actuelle de la fenêtre du navigateur et, lorsqu'elle se voit attribuer une nouvelle chaîne d'URL, indique au navigateur de charger et d'afficher l'URL spécifiée, permettant ainsi la navigation entre différentes pages.

4.Question

Pourquoi est-il important de comprendre que JavaScript est un langage sensible à la casse ?

Réponse:Savoir que JavaScript est sensible à la casse aide à prévenir les erreurs liées à la nomination des variables et des

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

fonctions, garantissant que les programmeurs utilisent de manière cohérente la bonne casse pour faire référence à leurs identifiants.

5.Question

Quelle est l'importance de la propriété 'navigator' dans l'objet Window ?

Réponse:La propriété 'navigator' fournit des détails importants sur le navigateur web de l'utilisateur, y compris son nom et sa version, ce qui est crucial pour écrire un code compatible avec plusieurs navigateurs.

6.Question

De quelle manière les fonctions définies par l'utilisateur peuvent-elles améliorer les capacités d'un programme JavaScript ?

Réponse:Les fonctions définies par l'utilisateur permettent un code modulaire et réutilisable, permettant aux développeurs d'encapsuler la logique et de réduire la redondance en appelant la même fonction plusieurs fois avec différents arguments au besoin.

7.Question

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Comment le concept de 'prototype' contribue-t-il à la programmation orientée objet en JavaScript ?

Réponse:Le prototype permet aux objets d'hériter des propriétés et des méthodes d'autres objets, facilitant une forme d'héritage qui favorise la réutilisation du code et une gestion efficace de la mémoire en évitant de dupliquer les méthodes pour chaque instance.

8.Question

Quel rôle joue la gestion des événements dans JavaScript côté client ?

Réponse:La gestion des événements permet à JavaScript de réagir aux interactions des utilisateurs avec les éléments HTML (comme les clics ou les pressions sur les touches), permettant des réponses dynamiques telles que la mise à jour du contenu de la page sans avoir besoin de recharger complètement la page.

9.Question

Expliquez le concept de 'portée' en JavaScript et son importance.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Réponse:La portée définit la visibilité et la durée de vie des variables en JavaScript. Comprendre la portée est crucial pour gérer l'accès aux variables et éviter les conflits dans des programmes plus vastes, en particulier avec la distinction entre variables globales et locales.

10.Question

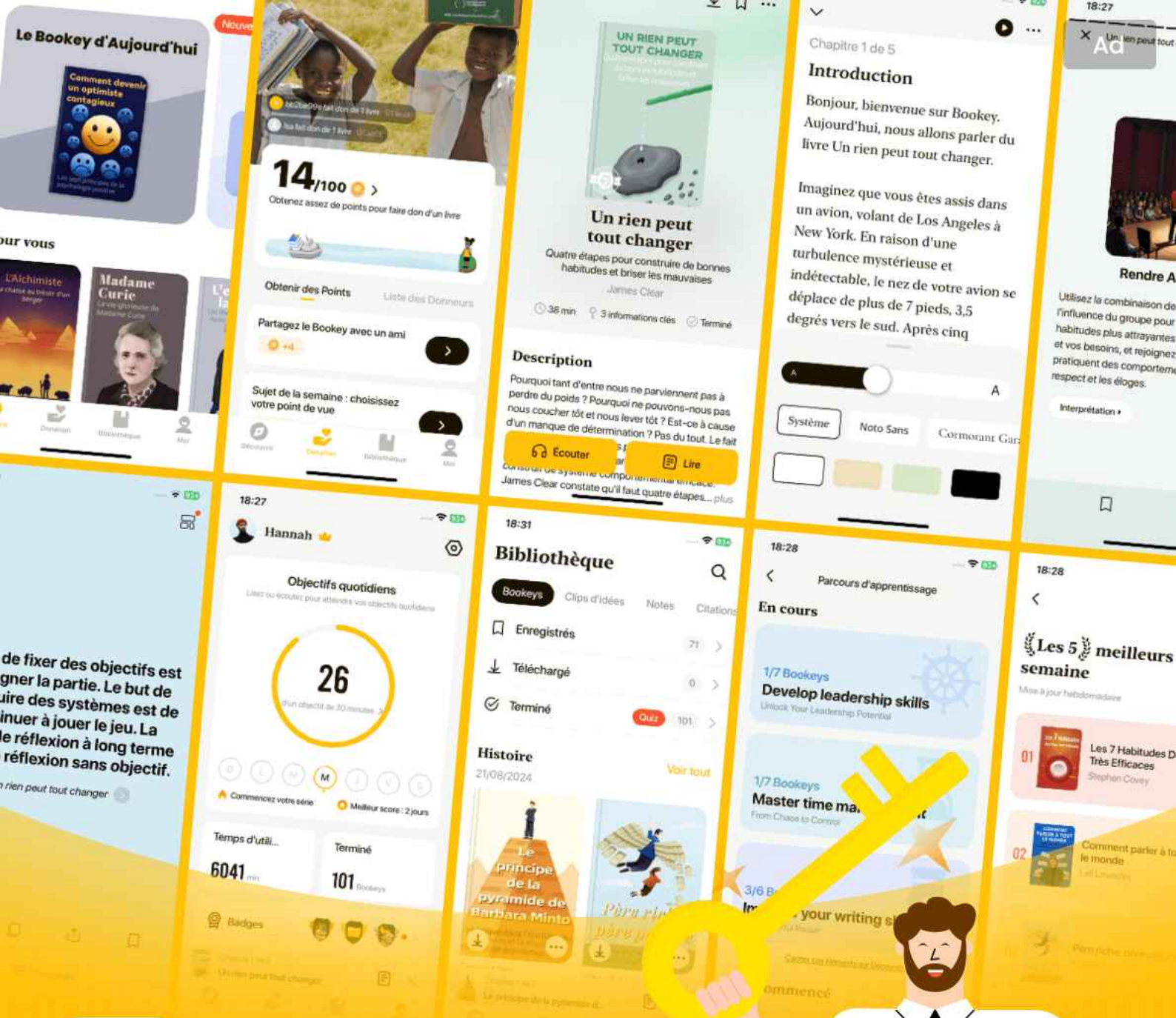
Pourquoi la compréhension des types de données en JavaScript est-elle essentielle pour les développeurs ?

Réponse:Comprendre les types de données est vital car ils dictent comment les valeurs peuvent être utilisées, manipulées et comparées en JavaScript, influençant la performance et la correction dans la logique de programmation.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé



Les meilleures idées du monde débloquent votre potentiel

Essai gratuit avec Bookey

Télécharger dans App Store

DISPONIBLE SUR Google Play

Scanner pour télécharger

JavaScript Quiz et test

Vérifier la réponse correcte sur le site de Bookey

Chapitre 1 | Section 1.1. Syntaxe| Quiz et test

1. La syntaxe de JavaScript ressemble à celle de Java, qui est dérivée de C et C++.
2. JavaScript ne fait pas la distinction entre majuscules et minuscules pour les mots clés et les identifiants.
3. Les identifiants en JavaScript peuvent commencer par un chiffre.

Chapitre 2 | Section 1.3. Types de Données| Quiz et test

1. JavaScript prend en charge trois types de données primitifs : les nombres, les booléens et les chaînes de caractères ; et deux types de données composés : les objets et les tableaux.
2. Le type booléen en JavaScript a trois valeurs possibles : vrai, faux et nul.
3. Les chaînes de caractères en JavaScript sont modifiables, ce qui signifie que les modifications altèrent la chaîne

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

originale.

Chapitre 3 | Section 1.4. Expressions et Opérateurs| Quiz et test

1. JavaScript utilise les mêmes opérateurs que C et C++ sans opérateurs supplémentaires.
2. L'opérateur '+' en JavaScript peut être utilisé à la fois pour l'addition et la concaténation de chaînes.
3. En JavaScript, l'opérateur 'typeof' ne peut être utilisé que sur des nombres.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 4 | Section 1.5. Instructions| Quiz et test

1. Chaque expression en JavaScript peut être une instruction autonome.
2. Une instruction vide est représentée par des accolades ({ }) et ne fait rien.
3. L'instruction 'for/in' est utilisée pour parcourir les propriétés d'un objet.

Chapitre 5 | Section 1.6. JavaScript orienté objet| Quiz et test

1. Les objets JavaScript sont des tableaux associatifs qui lient des valeurs à des propriétés nommées.
2. Une fonction constructeur est utilisée pour créer une nouvelle classe en JavaScript, et elle est invoquée sans l'opérateur 'new'.
3. Les propriétés définies sur le prototype d'un constructeur ne sont pas héritées par les objets créés avec ce constructeur.

Chapitre 6 | Section 1.7. Expressions Régulières| Quiz et test

1. Les expressions régulières de JavaScript utilisent une syntaxe similaire à celle de Java.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

2. Dans les expressions régulières, le caractère '.' correspond à n'importe quel caractère sauf un saut de ligne.
3. Le quantificateur '+' dans les expressions régulières correspond à zéro ou une fois.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 7 | Section 1.8. Versions de JavaScript| Quiz et test

1. JavaScript 1.0 était la première version de JavaScript implémentée dans Netscape 2 et est maintenant obsolète.
2. JScript 5.5 est équivalent à JavaScript 1.4 et est conforme à ECMA v2.
3. JavaScript peut être inclus dans HTML via la balise <script>, des gestionnaires d'événements et des URL JavaScript.

Chapitre 8 | Section 2.1. JavaScript dans HTML| Quiz et test

1. JavaScript peut être intégré dans des fichiers HTML grâce à des scripts, des gestionnaires d'événements et des URL JavaScript.
2. L'attribut `language` est la méthode recommandée pour spécifier le langage de script dans la balise <script>.
3. JavaScript ne peut être exécuté que par la balise <script> et ne peut pas être déclenché par des gestionnaires d'événements ou des URL.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 9 | Section 2.2. L'objet Window| Quiz et test

1. L'objet Window est l'objet global dans JavaScript côté client.
2. La méthode `setInterval()` est utilisée pour exécuter le code uniquement une fois à un intervalle spécifié.
3. La propriété `navigator` fournit des informations sur le système de l'utilisateur, y compris le nom du navigateur.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 10 | Section 2.3. L'objet Document| Quiz et test

1. Chaque objet Window a une propriété document qui fait référence à un objet Document, ce qui est crucial puisqu'elle représente le document HTML affiché dans la fenêtre du navigateur.
2. Le DOM W3C n'est pas standardisé et ne permet pas un accès complet à tout le contenu du document.
3. Le DOM IE 4 a été développé par Microsoft et offrait des fonctionnalités puissantes pour la manipulation du contenu du document qui n'étaient pas standardisées.

Chapitre 11 | Section 2.4. Le DOM Héritage| Quiz et test

1. Le DOM hérité permet d'accéder à diverses propriétés et contenus du document.
2. Les propriétés de l'objet Document incluent une méthode pour interagir directement avec tous les types de balises HTML.
3. La propriété cookie peut être utilisée pour définir et récupérer les cookies du document en JavaScript.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Chapitre 12 | Section 2.5. Le W3C DOM| Quiz et test

1. Le W3C DOM permet aux scripts de manipuler uniquement des formulaires et des images.
2. Vous pouvez trouver des éléments par leur identifiant unique en utilisant la méthode getElementById().
3. La propriété nodeType indique le type d'un nœud avec des valeurs telles que 1 pour le texte et 2 pour l'élément.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 13 | Section 2.6. DOM IE 4| Quiz et test

1. Le DOM d'IE 4 est un modèle d'objet de document standard.
2. Les éléments dans le DOM d'IE 4 peuvent être accessibles via le tableau all[] de l'objet document.
3. La traversée de l'arbre de document dans le DOM d'IE 4 utilise la propriété children[] au lieu de childNodes[].

Chapitre 14 | Section 2.7. DHTML : Scriptage des styles CSS| Quiz et test

1. Le HTML dynamique (DHTML) combine HTML, CSS et JavaScript pour des modifications de style dynamiques.
2. En JavaScript, les attributs CSS avec des tirets sont accessibles en utilisant la même convention de nommage avec des tirets.
3. L'attribut 'float' en CSS est appelé 'cssFloat' en JavaScript.

Chapitre 15 | Section 2.8. Événements et Gestion des Événements| Quiz et test

1. Les attributs de gestionnaire d'événements en HTML commencent par 'on'.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

2. Retourner faux depuis un gestionnaire d'événements empêchera l'action de onmouseover dans tous les cas.
3. Le modèle W3C permet d'enregistrer plusieurs gestionnaires d'événements pour un seul type d'événement en utilisant addEventListener().

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 16 | Section 2.9. Restrictions de sécurité en JavaScript| Quiz et test

1. Les scripts peuvent accéder aux propriétés des fenêtres et des documents depuis n'importe quel serveur web en raison de la politique de même origine.
2. JavaScript ne peut pas manipuler la propriété value de l'élément de formulaire FileUpload.
3. Les configurations des navigateurs peuvent permettre aux utilisateurs de bloquer les publicités pop-up et de manipuler les tailles de fenêtre en raison des restrictions de sécurité JavaScript en évolution.

Chapitre 17 | Tableau| Quiz et test

1. La méthode constructeur 'new Array()' crée un tableau vide en JavaScript.
2. Les tableaux en JavaScript ne peuvent être créés qu'en utilisant la méthode constructeur 'new Array()', et non avec une syntaxe littérale.
3. La méthode 'pop()' retire le premier élément d'un tableau en

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

JavaScript.

Chapitre 18 | Date| Quiz et test

1. L'objet Date en JavaScript peut être créé en utilisant différents formats, y compris un timestamp en millisecondes.
2. La méthode `getYear()` de l'objet Date renvoie l'année complète (par exemple, 2023).
3. La méthode `Date.UTC(...)` renvoie la représentation en millisecondes de la date et de l'heure locales spécifiées.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 19 | Document| Quiz et test

1. L'objet Document a été introduit dans JavaScript 1.0 et a été amélioré dans JavaScript 1.1.
2. La propriété 'alinkColor' de l'objet Document est actuellement recommandée pour une utilisation.
3. La méthode 'createElement(tagName)' est utilisée pour créer un nouveau nœud Element dans le document.

Chapitre 20 | Élément| Quiz et test

1. L'objet Element représente un élément ou une balise HTML dans un document et hérite des propriétés de l'objet Node tel que défini dans le niveau 1 du DOM.
2. La propriété 'all[]' est une propriété DOM W3C utilisée pour accéder aux objets Element descendants.
3. La méthode 'getAttribute(name)' est utilisée pour supprimer un attribut nommé d'un objet Element.

Chapitre 21 | Événement| Quiz et test

1. L'objet Event fournit des détails sur un événement et contrôle sa propagation à travers différents

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

environnements de navigateur.

2. Toutes les propriétés de l'objet Event sont en lecture-écriture, ce qui permet de les modifier pendant le traitement des événements.
3. Dans le modèle IE, l'événement peut être accédé via la propriété event de l'objet Window.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 22 | Global| Quiz et test

1. L'objet Global en JavaScript est représenté par l'objet Window dans le scripting côté client.
2. La fonction isNaN(n) vérifie si n est un nombre.
3. La fonction encodeURIComponent(s) encode tous les caractères dans une URI.

Chapitre 23 | Input| Quiz et test

1. L'objet Input en JavaScript représente les éléments HTML `<input>` et fournit des propriétés pour leurs attributs.
2. La propriété 'defaultChecked' indique si un champ de saisie de texte était coché à l'origine.
3. Le gestionnaire d'événements 'onchange' est activé lorsque le focus s'éloigne après avoir modifié le texte.

Chapitre 24 | Calque| Quiz et test

1. L'objet Layer est une fonctionnalité standard utilisée dans tous les navigateurs.
2. Vous pouvez référencer un objet Layer en utilisant document.layers[i] ou document.layers[nom-du-layer].

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

3.La méthode load(src, width) est utilisée uniquement pour redimensionner le layer à des dimensions spécifiques.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 25 | Lien| Quiz et test

1. L'objet Link en JavaScript hérite de la classe Element.
2. La propriété 'search' de l'objet Link n'inclut pas le point d'interrogation dans sa valeur.
3. Le gestionnaire d'événements 'onclick' peut être utilisé pour empêcher de suivre un lien s'il renvoie false dans JavaScript 1.1.

Chapitre 26 | Math| Quiz et test

1. L'objet Math en JavaScript a un constructeur et ses fonctions sont des méthodes d'un objet.
2. Math.PI est une constante qui représente JavaScript.
3. Math.max() renverra -Infinity si aucun argument n'est fourni.

Chapitre 27 | Navigateur| Quiz et test

1. La propriété appCodeName de l'objet navigator renvoie généralement 'Mozilla' pour les navigateurs Netscape et Microsoft.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

2.La propriété userLanguage spécifie la langue par défaut de l'utilisateur sous forme de valeur numérique.

3.La méthode javaEnabled() renvoie true si Java n'est pas supporté dans le navigateur actuel.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 28 | Node| Quiz et test

1. Un nœud dans un arbre de documents peut être de types tels que Attr, Comment, Document, DocumentFragment, Element ou Text.
2. La propriété nodeType d'un objet Node spécifie la valeur de son type de nœud mais n'est pas en lecture seule.
3. La méthode `addEventListener(type, listener, useCapture)` est utilisée pour supprimer un écouteur d'événement d'un nœud.

Chapitre 29 | Nombre| Quiz et test

1. Le constructeur Number ne peut créer un nouvel objet Number que s'il est utilisé avec l'opérateur new.
2. Number.MAX_VALUE représente le plus petit nombre représentable en JavaScript.
3. La méthode toFixed renvoie une représentation sous forme de chaîne d'un nombre avec un nombre spécifié de chiffres décimaux.

Chapitre 30 | Objet| Quiz et test

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

1. Le constructeur `new Object();` crée un objet vide auquel on peut ajouter des propriétés arbitraires.
2. La méthode `valueOf()` pour le type Object renvoie sa valeur primitive si elle est disponible. Pour Object, elle renvoie un objet différent au lieu de l'objet lui-même.
3. La méthode `toLocaleString()` fournit une représentation sous forme de chaîne générique de l'objet sans permettre aux sous-classes de modifier son comportement.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 31 | RegExp| Quiz et test

1. Les expressions régulières (RegExp) ne peuvent être définies que par un constructeur en JavaScript.
2. La méthode 'test' de RegExp renvoie vrai s'il y a une correspondance, sinon elle renvoie faux.
3. La propriété d'instance 'global' indique si le RegExp est utilisé pour une correspondance insensible à la casse.

Chapitre 32 | Sélection| Quiz et test

1. La propriété 'selectedIndex' renvoie -1 si aucune option n'est sélectionnée.
2. La méthode 'remove(n)' supprime le premier élément du tableau d'options lorsque n est 0.
3. La méthode 'blur()' est utilisée pour obtenir le focus clavier sur un objet Select.

Chapitre 33 | Chaîne| Quiz et test

1. La manipulation de chaînes a été introduite dans JavaScript 1.0.
2. La méthode `substring(from, to)` extrait les caractères entre

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

`to` et `from`.

3. La propriété `length` d'une chaîne en JavaScript est une propriété en lecture seule qui représente le nombre de caractères dans la chaîne.

Plus de résumé du livre

Scanne pour télécharger

Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play

Chapitre 34 | Style| Quiz et test

1. L'objet Style en JavaScript permet de manipuler directement les propriétés CSS définies par la spécification CSS2.
2. La propriété float est conservée sous le nom 'float' dans l'objet Style pour maintenir la cohérence avec les normes CSS.
3. Toutes les propriétés CSS supportées par l'objet Style sont garanties d'être implémentées par tous les navigateurs actuels.

Chapitre 35 | Window| Quiz et test

1. La méthode `alert()` est utilisée pour afficher un message dans une boîte de dialogue en JavaScript.
2. La propriété `navigator` fournit des informations spécifiques au navigateur de l'utilisateur.
3. Les gestionnaires d'événements en JavaScript sont définis à l'aide d'attributs dans les balises HTML.

Plus de résumé du livre
Scanne pour télécharger
Écouter le résumé

Scannez pour télécharger

Téléchargez l'appli Bookey pour profiter

Plus de 1000 résumés de livres avec des quiz

Essai gratuit disponible !

Télécharger dans App Store

DISPONIBLE SUR Google Play