

Apprendre JavaScript - Objets pré-définis

Introduction

JavaScript fournit des objets pré-définis qui sont utilisés pour effectuer des tâches courantes. Les objets `Window`, `Document`, `Element`, `String`, `Array`, `Date`, `Math`, etc. sont des exemples d'objets pré-définis.

Pour lister dans la console tous les objets pré-définis, vous pouvez utiliser la méthode `console.dir`.

L'objet `Window`

L'objet `Window` représente la fenêtre du navigateur. Il est le point d'entrée global pour JavaScript dans le navigateur. Toutes les variables globales et les fonctions sont définies sur l'objet `Window`.

<https://developer.mozilla.org/en-US/docs/Web/API/Window>

Propriétés importantes

- `window.innerHeight` : la hauteur de la fenêtre du navigateur. Ceci inclut la barre de défilement.
- `window.innerWidth` : la largeur de la fenêtre du navigateur. Ceci inclut la barre de défilement.
- `window.clientHeight` : la hauteur de la zone d'affichage du navigateur.
- `window.clientWidth` : la largeur de la zone d'affichage du navigateur.
- `window.location` : l'URL de la page actuelle. Ceci est utile pour rediriger l'utilisateur vers une autre page.
- `window.document` : l'objet `Document` qui représente le contenu de la page. Même chose que `document`.
- `window.navigator` : l'objet `Navigator` qui fournit des informations sur le navigateur.
- `window.localStorage` : l'objet `Storage` qui permet de stocker des données dans le navigateur.
- `window.sessionStorage` : l'objet `Storage` qui permet de stocker des données de session dans le navigateur.

Méthodes importantes

- `window.setTimeout` : exécute une fonction après un certain délai.
- `window.setInterval` : exécute une fonction à intervalles réguliers.
- `window.open` : ouvre une nouvelle fenêtre de navigateur.
- `window.close` : ferme la fenêtre de navigateur actuelle.
- `window.scrollTo` : fait défiler la fenêtre à une position spécifiée.
- `window.scrollBy` : fait défiler la fenêtre par un nombre spécifié de pixels.
- `window.scroll` : fait défiler la fenêtre à une position spécifiée.
- `window.print` : ouvre la boîte de dialogue d'impression pour imprimer la page.

Objet `Document`

L'objet `Document` représente le contenu de la page web. Il est le point d'entrée pour accéder à tout le contenu de la page web.

<https://developer.mozilla.org/en-US/docs/Web/API/Document>

Propriétés importantes

- `document.body` : l'élément `body` de la page.
- `document.head` : l'élément `head` de la page.
- `document.title` : le titre de la page.
- `document.URL` : l'URL de la page.

Objet `ElementHTML`

L'objet `ElementHTML` représente un élément HTML. Il est le point d'entrée pour accéder à un élément HTML. Tous les éléments HTML ont des propriétés et des méthodes communes, incluant les éléments `input`, `button`, `div`, `p`, `span`, `ul`, `li`, et même `document` qui est l'élément racine de la page.

Propriétés importantes

- `element.innerHTML` : le contenu HTML de l'élément.
- `element.textContent` : le contenu textuel de l'élément.
- `element.tagName` : le nom de la balise de l'élément.
- `element.id` : l'attribut `id` de l'élément.
- `element.classList` : la liste des classes de l'élément. Ceci est un objet `DOMTokenList`.
- `element.dataset` : l'objet `DOMStringMap` qui représente les attributs `data-*` de l'élément.
- `element.children` : la liste des éléments enfants de l'élément. Ceci est un objet `HTMLCollection`.
- `element.parentElement` : l'élément parent de l'élément.

Méthodes importantes

- `element.querySelector` : récupère le premier élément qui correspond à un sélecteur CSS.
- `element.querySelectorAll` : récupère tous les éléments qui correspondent à un sélecteur CSS.
- `element.appendChild` : ajoute un élément enfant à l'élément.
- `element.removeChild` : supprime un élément enfant de l'élément.
- `element.cloneNode` : crée une copie de l'élément.
- `element.addEventListener` : ajoute un écouteur d'événement à l'élément.
- `element.removeEventListener` : supprime un écouteur d'événement de l'élément.
- `element.closest` : récupère l'ancêtre le plus proche qui correspond à un sélecteur CSS.

Il existe des éléments HTML spécifiques qui ont des propriétés et des méthodes spécifiques. Par exemple, l'élément `input` a une propriété `value` qui représente la valeur de l'élément, et une méthode `focus` qui met le focus sur l'élément. Les formulaires ont des méthodes `submit` et `reset` qui soumettent et réinitialisent le formulaire.

Vous pouvez trouver la liste complète des éléments HTML et de leurs propriétés et méthodes dans la documentation de MDN.

<https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement>

Objet **Math**

L'objet **Math** fournit des méthodes pour effectuer des opérations mathématiques.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math

Propriétés importantes

- **Math.PI** : la valeur de π .

Méthodes importantes

- **Math.abs** : retourne la valeur absolue d'un nombre, donc sa valeur sans signe (positif).
- **Math.ceil** : retourne le plus petit entier supérieur ou égal à un nombre.
- **Math.floor** : retourne le plus grand entier inférieur ou égal à un nombre.
- **Math.round** : retourne la valeur d'un nombre arrondi à l'entier le plus proche.
- **Math.max** : retourne le plus grand nombre d'une liste de nombres.
- **Math.min** : retourne le plus petit nombre d'une liste de nombres.
- **Math.random** : retourne un nombre aléatoire entre 0 (inclus) et 1 (exclus).
- **Math.pow** : retourne la puissance d'un nombre.
- **Math.sqrt** : retourne la racine carrée d'un nombre.
- **Math.sin** : retourne le sinus d'un angle en radians.
- **Math.cos** : retourne le cosinus d'un angle en radians.
- **Math.tan** : retourne la tangente d'un angle en radians.

Sélectionner un nombre aléatoire d'éléments dans un tableau

Pour sélectionner un nombre aléatoire d'éléments dans un tableau, vous pouvez utiliser la méthode **Math.random** avec la méthode **Math.floor**.

```
let elements = ["Un", "Deux", "Trois", "Quatre", "Cinq"];

let randomIndex = Math.floor(Math.random() * elements.length);

console.log(elements[randomIndex]);
```

Objet **Date**

L'objet **Date** représente une date et une heure. La gestion du temps en programmation est complexe, mais l'objet **Date** fournit des méthodes pour effectuer des opérations courantes.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date

Méthodes importantes

- `Date.now` : retourne le nombre de millisecondes écoulées depuis le 1er janvier 1970 00:00:00 UTC.
- `Date.parse` : analyse une chaîne de caractères représentant une date et retourne le nombre de millisecondes écoulées depuis le 1er janvier 1970 00:00:00 UTC.
- `Date.UTC` : retourne le nombre de millisecondes écoulées depuis le 1er janvier 1970 00:00:00 UTC pour une date spécifiée en temps universel coordonné (UTC).
- `date.getDate` : retourne le jour du mois pour la date spécifiée.
- `date.getDay` : retourne le jour de la semaine pour la date spécifiée.
- `date.getFullYear` : retourne l'année pour la date spécifiée.
- `date.getHours` : retourne l'heure pour la date spécifiée.
- `date.getMilliseconds` : retourne les millisecondes pour la date spécifiée.
- `date.getMinutes` : retourne les minutes pour la date spécifiée.
- `date.getMonth` : retourne le mois pour la date spécifiée.
- `date.getSeconds` : retourne les secondes pour la date spécifiée.

Créer une date

Pour créer une date, vous pouvez utiliser le constructeur `Date` avec une chaîne de caractères représentant une date, ou avec des paramètres pour l'année, le mois, le jour, l'heure, les minutes, les secondes et les millisecondes.

```
let date1 = new Date("2021-01-01");
let date2 = new Date(2021, 0, 1); // Le mois commence à 0
let date3 = new Date(2021, 0, 1, 12, 0, 0, 0); // 12:00:00
```

Formater une date

Pour formater une date, vous pouvez utiliser les méthodes `getDate`, `getMonth`, `getFullYear`, `getHours`, `getMinutes`, `getSeconds` et `getMilliseconds` pour récupérer les parties de la date, et les concaténer dans une chaîne de caractères.

```
let date = new Date();

let jour = date.getDate();
let mois = date.getMonth() + 1; // Le mois commence à 0
let annee = date.getFullYear();
let heures = date.getHours();
let minutes = date.getMinutes();
let secondes = date.getSeconds();

console.log(jour + "/" + mois + "/" + annee + " " + heures + ":" + minutes + ":" + secondes);
```