

Manipuler les attributs et les styles des éléments via le DOM en JavaScript

Grâce au DOM, nous allons également pouvoir tester si un élément possède un attribut, récupérer la valeur d'un attribut donné, ajouter, modifier ou supprimer des attributs. Nous allons également pouvoir manipuler les styles CSS de nos éléments.

Tester la présence d'attributs

La méthode `hasAttribute()` de l'interface `Element` nous permet de tester la présence d'un attribut en particulier pour un élément. Cette méthode prend en argument le nom de l'attribut qu'on recherche et renvoie la valeur booléenne `true` si l'élément possède bien cet attribut ou `false` sinon.

Pour vérifier si un élément possède des attributs ou pas (quels qu'ils soient), on utilisera plutôt la méthode `hasAttributes()` de cette même interface. Cette méthode retourne à nouveau une valeur booléenne (`true` si l'élément possède au moins un attribut ou `false` si l'élément ne possède pas d'attribut).

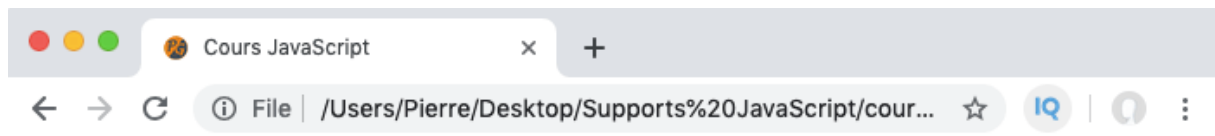
```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours JavaScript</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="cours.css">
    <script src='cours.js' async></script>
  </head>

  <body>
    <h1>Titre principal</h1>
    <p id='p1' class='blue'>Un paragraphe <span>avec un span</span></p>
    <div>
      <p id='p2'>Un paragraphe dans le div</p>
      <p>Un autre paragraphe dans le div</p>
    </div>
    <p>Un autre paragraphe</p>
    <p id='vide'></p>
  </body>
</html>
```

```
let p1 = document.querySelector('p');
let vide = document.getElementById('vide');

/*Si p1 possède des attributs, hasAttributes() renvoie true et on exécute le code
*de la condition*/
if(p1.hasAttributes()){
    vide.textContent = 'p1 possède des attributs';
}

//Si p1 possède un attribut id, hasAttribute() renvoie true
if(p1.hasAttribute('id')){
    vide.textContent += ' dont un attribut id';
}
```



Titre principal

Un paragraphe avec un span

Un paragraphe dans le div

Un autre paragraphe dans le div

Un autre paragraphe

p1 possède des attributs dont un attribut id

Récupérer la valeur ou le nom d'un attribut ou définir un attribut

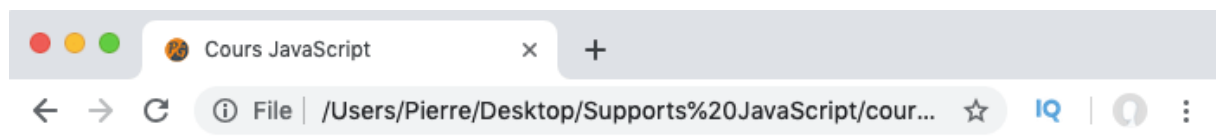
La propriété `attributes` de l'interface `Element` renvoie la liste des attributs d'un (nœud) élément spécifié. La liste d'attributs est renvoyée sous la forme « clef / valeur » et est un objet de l'interface `NamedNodeMap`.

L'interface `NamedNodeMap` est une interface qui sert à représenter des ensembles d'objets de l'interface `Attr`. L'interface `Attr` sert à représenter les attributs des éléments du DOM sous forme d'objets.

On va pouvoir récupérer les noms et valeurs de chaque attribut en utilisant une boucle `for` pour itérer dans cette liste et les propriétés `name` et `value` de l'interface `Attr`.

```
let p1 = document.querySelector('p');
let vide = document.getElementById('vide');

if(p1.hasAttributes()){
  let attP1 = p1.attributes; // Liste des attributs de p1
  vide.innerHTML = 'Liste des attributs de p1 : <br>'
  //La propriété length renvoie le nombre d'attributs
  for(let i = 0; i < attP1.length; i++){
    vide.innerHTML += attP1[i].name + ' = ' + attP1[i].value + '<br>';
  }
}
```



Titre principal

Un paragraphe avec un span

Un paragraphe dans le div

Un autre paragraphe dans le div

Un autre paragraphe

Liste des attributs de p1 :
id = p1
class = blue

Si on ne souhaite récupérer que les noms des attributs d'un élément, on peut également utiliser la méthode `getAttributeNames()` qui renvoie les noms des attributs d'un élément sous forme de tableau (type `Array`).

Pour ne récupérer que la valeur d'un attribut donné pour un élément, on utilisera plutôt la méthode `getAttribute()`. Cette méthode renvoie la valeur de l'attribut donné si celui-ci existe ou `null` ou la chaîne de caractères vide dans le cas contraire.

```
let p1 = document.querySelector('p');
let vide = document.getElementById('vide');

//Si p1 possède des attributs...
if(p1.hasAttributes()){
    //On récupère leurs noms dans un tableau
    let attP1 = p1.getAttributeNames();
    vide.innerHTML = 'Attributs de p1 : <br>';

    //Pour chaque élément du tableau...
    for(nom of attP1){
        //On récupère la valeur associée au nom de l'attribut
        let valeur = p1.getAttribute(nom);
        //On affiche le nom et la valeur de l'attribut
        vide.innerHTML += nom + ' = ' + valeur + '<br>';
    }
}
```



Titre principal

Un paragraphe avec un span

Un paragraphe dans le div

Un autre paragraphe dans le div

Un autre paragraphe

Attributs de p1 :

id = p1

class = blue

Pour ajouter un nouvel attribut ou changer la valeur d'un attribut existant pour un élément, nous allons cette fois-ci utiliser la méthode `setAttribute()` à laquelle on va passer en arguments le nom et la valeur de l'attribut à ajouter ou à modifier.

Notez que pour obtenir ou pour définir la valeur d'un attribut `class` ou `id` en particulier, on va également pouvoir utiliser les propriétés `className` et `id` de l'interface `Element`.

```

<!DOCTYPE html>
<html>
  <head>
    <title>Cours JavaScript</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="cours.css">
    <script src='cours.js' async></script>
    <style>
      .blue{
        color: blue;
      }
    </style>
  </head>

  <body>
    <h1>Titre principal</h1>
    <p id='p1' class='blue'>Un paragraphe <span>avec un span</span></p>
    <div>
      <p id='p2'>Un paragraphe dans le div</p>
      <p>Un autre paragraphe dans le div</p>
    </div>
    <p>Un autre paragraphe</p>
    <p id='vide'></p>
  </body>
</html>

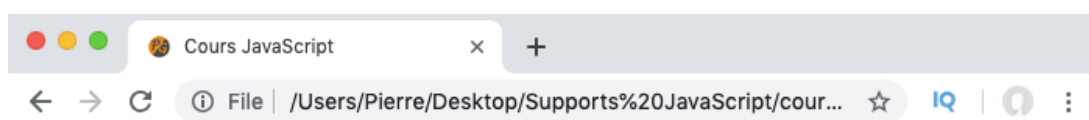
```

```

let p1 = document.querySelector('p');
let p2 = document.getElementById('p2');
let vide = document.getElementById('vide');

p2.setAttribute('class', 'blue');
vide.innerHTML += 'class : ' + p1.className + '<br>id : ' + p1.id;

```



Titre principal

Un paragraphe avec un span

Un paragraphe dans le div

Un autre paragraphe dans le div

Un autre paragraphe

class : blue

id : p1

JS 13c

Réalisation : Guillaume DELACROIX Formateur AFPA

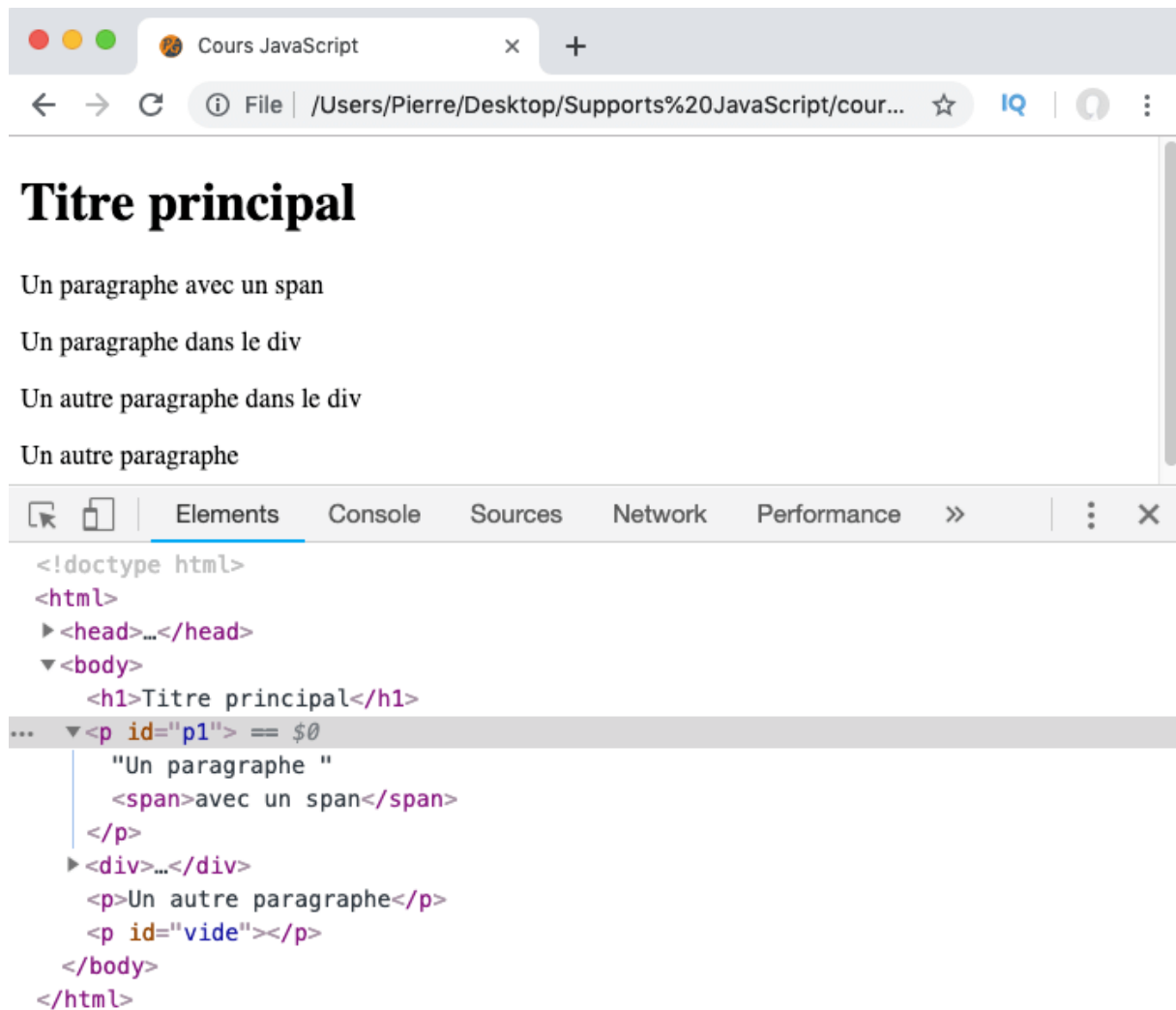
02 février 2023

Afpa

Supprimer un attribut

Pour supprimer un attribut d'un élément, nous allons pouvoir utiliser la méthode `removeAttribute()` de l'interface `Element`. Cette méthode va prendre en argument le nom de l'attribut à supprimer.

```
let p1 = document.querySelector('p');  
p1.removeAttribute('class');
```



Inverser la valeur logique d'un attribut

On va également pouvoir inverser la valeur logique d'un attribut de type booléen avec la méthode `toggleAttribute()` de `Element`. Cette méthode va pouvoir s'avérer très pratique pour activer ou désactiver des fonctions liées à la valeur d'un booléen.

Modifier les styles d'un élément

Finalement, on va pouvoir modifier les styles d'un élément grâce à la propriété `style` de l'interface `HTMLElement` qui est une interface représentant les nœuds de type `Element` seulement dans le document et qui hérite de `Element`.

Pour être tout à fait précis ici, vous devez savoir que le DOM possède une interface pour chaque élément HTML : l'interface `HTMLFormElement` pour les éléments de formulaire, l'interface `HTMLAnchorElement` pour les éléments de liens, l'interface `HTMLDivElement` pour les éléments `div`, etc.

Chaque interface spécifique à un élément va hériter de `HTMLElement` et donc, par extension, de `Element`, de `Node` et de `EventTarget`.

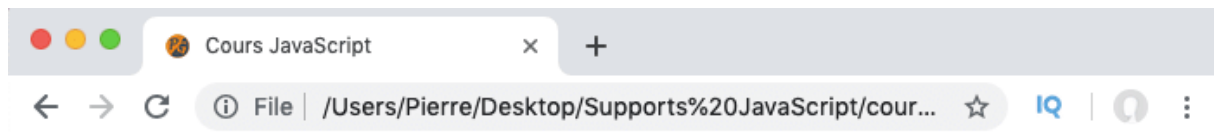
La propriété `style` de `HTMLElement` va nous permettre de définir les styles en ligne d'un élément (les styles vont être placés dans la balise ouvrante de l'élément directement).

Cette propriété retourne un objet à partir duquel on va pouvoir utiliser des propriétés JavaScript représentant les propriétés CSS. Ces propriétés respectent la norme lower camel case : elles doivent être écrites sans espace ni tiret, avec une majuscule au début de chaque mot sauf pour le premier : la propriété CSS `background-color`, par exemple, s'écrira `backgroundColor`.

```
let p1 = document.querySelector('p');
let p2 = document.getElementById('p2');

p1.style.color = 'crimson'; //Nuance de rouge
p1.style.fontSize = '20px';

p2.style.backgroundColor = 'orange';
```



Titre principal

Un paragraphe avec un span

Un paragraphe dans le div

Un autre paragraphe dans le div

Un autre paragraphe

