

Manipulation du DOM avec Vanilla Javascript

Vanilla Javascript

Vanilla Javascript, est le nom pour désigner l'utilisation de javascript sans aucune librairie.

il est plus verbeux à écrire mais beaucoup plus rapide que JQuery lors de l'exécution.

```
1  //cacher un élément vi VanillaJS
2  const el = document.getElementById('test');
3  el.style.display = 'none';
4
5  //cacher un élément via JQuery
6  $('#test').hide();
7
```

Documentation

Vous pouvez trouver toute la documentation sur la manipuler DOM (via l'interface ***Document***) sur le site de mozilla

<https://developer.mozilla.org/fr/docs/Web/API/Document>



MDN **MOZILLA
DEVELOPER
NETWORK**

Interroger le DOM

Voici des méthodes de l'interface ***Document*** permettant de récupérer un ou des éléments

- ***getElementById***
- ***getElementByClassName***
- ***getElementsByName***
- ***getElementsByTagName***
- ***querySelector***
- ***querySelectorAll***

Interroger le DOM

document.***getElementById()*** permet de renvoyer un objet ***Element*** dont la propriété id correspond à la chaîne de caractères spécifiée.

Exemple

```
1  const elementById = document.getElementById("val4");  
2  console.log(elementById)
```

Interroger le DOM

document.***getElementsByClassName()*** renvoie un objet de type tableau de tous les éléments enfants qui ont tous les noms de classe données.

Exemple

```
1 //rechercher dans tout le document les éléments qui ont la classe header
2 const headers = document.getElementsByClassName('header');
3 console.log(headers);
4
5 //rechercher dans l'element tab1 tous les éléments qui ont la classe header
6 const element = document.getElementById("tab1");
7 const headers2 = element.getElementsByClassName('header');
8 console.log(headers2);
9 |
```

Interroger le DOM

document.***getElementsByTagName()*** renvoie un objet de type tableau de tous les éléments enfants qui ont tous le nom de balise donné.

Exemple

```
1 //rechercher dans tout le document les éléments qui ont le nom du tag donné
2 const divs = document.getElementsByTagName('div');
3 console.log(divs);
4
5 //rechercher dans l'element tabl tous les éléments qui ont le nom du tag donné
6 const element = document.getElementById("containerForm");
7 const divs2 = element.getElementsByTagName('div');
8 console.log(divs2);
9 |
```

Interroger le DOM

document.***querySelector()*** retourne le premier Element dans le document correspondant au sélecteur - ou groupe de sélecteurs - spécifié(s).

Exemple

```
1 //rechercher le premier élément de type texte
2 const text = document.querySelector('input[type="text"]');
3 console.log(text);
4
5 //récupère le premier th du #tab1
6 const th = document.querySelector('#tab1 th');
7 console.log(th);
8
```


Interroger le DOM

document.***querySelectorAll()*** retourne tableau de NodeList des éléments dans le document correspondant au sélecteur - ou groupe de sélecteurs - spécifié(s).

Exemple

```
1 //recherche tous les elements qui ont le tag input
2 const inputs = document.querySelectorAll('input');
3 console.log(inputs);
4
5 //recherche tous les elements qui sont dans la tab1 et qui ont le tag th
6 const ths = document.querySelectorAll('#tab1 th');
7 console.log(ths);
8 |
```

Exercices

Récupérer le premier **label** se trouvant dans le bloc **#containerForm**

Résultat à obtenir:

```
<label>Texte</label>
```

Exercices

Récupérer tous les éléments qui ont la classe *header*

Résultat à obtenir:

```
< ▼ NodeList(4) [th.header, td.header, td.header, td.header] ⓘ  
  ► 0: th.header  
  ► 1: td.header  
  ► 2: td.header  
  ► 3: td.header  
    length: 4  
  ► __proto__: NodeList
```

Exercices

Trouver deux possibilités de récupérer toutes les **div** de la page

Résultat à obtenir:

```
▼ HTMLCollection(4) [div#cube, div#containerForm, div, div, cube: div#cube, containerForm: div#containerForm] ⓘ  
  ► 0: div#cube  
  ► 1: div#containerForm  
  ► 2: div  
  ► 3: div  
    length: 4  
  ► containerForm: div#containerForm  
  ► cube: div#cube  
  ► __proto__: HTMLCollection
```

Modifier le DOM

Vous pouvez mettre à jour un attribut grâce à la méthode `setAttribute()`, l'exemple ci dessous va rajouter une classe à l'élément ayant l'id **val2**

```
1  
2 document.getElementById('val2').setAttribute('class', 'redColor');  
3
```

Modifier le DOM

Insérer un li dans une liste

```
1  const _li = document.createElement("li");  
2  _li.innerText = 'Vacances';  
3  document.querySelector('#list').appendChild(_li);  
4  |
```

Supprimer un élément dans le DOM

```
1  const li = document.querySelector('#list li');  
2  li.remove();  
3
```

Dès que vous avez un objet **Element** vous pouvez utiliser la méthode **remove** pour supprimer votre élément du **DOM**