

Javascript

DOM

(Document Object Model)

Plan

- Accéder aux éléments du DOM
 - `querySelector`
 - `querySelectorAll`
 - La boucle `foreach`
- Appliquer un style à un élément
- La propriété `classList`
- Activer et désactiver un bouton

Accéder aux éléments d'un document(querySelector)

Cette méthode représente le **premier élément** HTML correspondant au **sélecteur CSS** spécifié en argument.

Exemple:

```
var x=document.querySelector("p span");
```

//référence le premier span se trouvant à l'intérieur d'un paragraphe

```
var y=document.querySelector(".dd");
```

//référence le premier élément ayant la classe dd.

Accéder aux éléments d'un document(querySelectorAll)

- Cette méthode renvoie tous les éléments (**tableau d'éléments**) qui correspondent au sélecteur spécifié.
- Exemple:

```
var y=document.querySelectorAll(".dd");  
//référence tous les éléments ayant la classe dd.  
//Y est un tableau d'objets
```

forEach

- Cette méthode est utilisé pour les tableau (c'est un écriture récente)

```
var tab=new Array(12,15,20,18)

tab.forEach(element => {
    alert(element)
});

// ou bien
tab.forEach(function(element){
    alert(element)
});
```

- Pour chaque élément (**element**) du tableau **tab**, on affiche sa valeur.
- Le résultat: affiche les valeurs **12,15,20 et 18**

Appliquer un style CSS

Ajouter ou modifier un style d'un élément en javascript

- Pour changer le style d'un élément, il suffit d'utiliser la propriété **style**, en lui affectant une valeur à une propriété de style css.

```
UnElement.style.mapropriete=valeur;
```

Ajouter ou modifier un style d'un élément en javascript

- *Exemple:*

```
document.getElementById("d1").style.color="blue";
```

- Le texte de l'élément qui possède l'attribut **id** égal à **d1** sera **coloré en bleu**.

Ajouter ou modifier un style d'un élément en javascript

- Les propriétés qui sont composés comme background-color, font-size, text-transform,...seront utilisés en javascript comme suit: le tiret (-) est enlevé et la première lettre du deuxième mot doit être en majuscule.
- Exemple: background**C**olor, font**S**ize, text**T**ransform,...
- Utilisation:

```
document.getElementById("d1").style.fontSize="16px";
```

Appliquer un style aux éléments avec le nom de classe en JavaScript

```
<body onload="mafct()">
<div class="blue-bg">
  <label>User Name: </label>
  <input type="text" id="username" name="UserName">
</div>
</body>
```

```
document.getElementsByClassName("blue-bg")[0].style.backgroundColor = "green";
```

Appliquer un style aux balises en JavaScript

```
<div >
  <label>User Name: </label>
  <input type="text" id="username" name="UserName">
</div>
<div>
  <label>Password: </label>
  <input type="password" name="password">
</div>
<div>
  <label>Description: </label>
  <textarea name="description"></textarea>
</div>
```

```
document.getElementsByTagName("div")[0].style.backgroundColor = "green";
```

Modifier plusieurs styles à la fois

- Si on a un grand nombre de styles à modifier par javascript, ça devient lourds d'écrire chaque propriété dans une ligne. On utilise donc la syntaxe suivante:

```
MonElement.style.cssText="propriété1 : valeur ; propriété2:valeur ;...."
```

```
var x=document.getElementById("c1")  
x.style.cssText="color:red;background-color:yellow;"
```

classList

- **classList** est une **propriété** en lecture seule qui renvoie une collection de classes CSS d'un élément :

Récupérer les classes CSS d'un élément

- Supposons que nous avons un élément **div** avec deux classes : **main** et **red**

```
<div id="content" class="main red">JavaScript classList</div>
```

- Le code suivant affiche la liste des classes du div :

```
let div = document.querySelector('#content');  
let T= div.classList  
  
for(cl of T){  
  alert(cl)           //affiche main et red  
}
```

Ajouter une ou plusieurs classes à la liste des classe d'un élément

- Pour ajouter une nouvelle classe à un élément, on utilise la méthode `add()` :
- Par exemple le code suivant ajoute la classe `info` à la liste des classe du `div` :

```
let div = document.querySelector('#content');  
div.classList.add('info');
```

- On peut ajouter plusieurs classes à la fois :

```
let div = document.querySelector('#content');  
div.classList.add('info', 'red');
```

Supprimer les classes d'un élément

- Pour supprimer une classe d'un élément, on utilise la méthode **remove()** :

```
let div = document.querySelector('#content');  
div.classList.remove('visible');
```

- On peut supprimer plusieurs classes à la fois :

```
let div = document.querySelector('#content');  
div.classList.remove('visible', 'info');
```


Remplacer une classe d'un élément

- Pour remplacer une classe par une autre on peut utiliser la méthode **replace()** :

```
let div = document.querySelector('#content');  
div.classList.replace('info', 'warning');
```

Vérifier si un élément possède une classe

- Pour vérifier qu'un élément possède une classe spécifique, on utilise la méthode **contains()**

```
let div = document.querySelector('#content');  
div.classList.contains('visible'); // true
```

Désactiver un bouton et le réactiver

- Pour désactiver un bouton, on utilise la propriété **disabled** et on lui affecte la valeur **true**:

```
let b=document.querySelector("input[type='button']")
b.disabled=true
```

- Pour activer le bouton, on utilise la même propriété **disabled** avec la valeur **false**.

```
let b=document.querySelector("input[type='button']")
if(b.disabled==true){
  b.disabled=false
}
```

Détecter la touche du clavier

- Soit un input comme suit:

```
<input type="text" onkeypress="tes()" >
```

- On veut savoir si on a tapé la touche Entre du clavier ou non:

```
function tes(){  
    console.log(event)  
    if(event.keyCode===13){  
        alert("ok")  
    }  
}
```

- Event contient des informations sur l'événement déclenché, le code la touche Entre est 13

L'opérateur this

- L'opérateur **this** permet de **référencer l'élément courant** (une zone de texte, un paragraphe, une image,...), pour lequel on applique un style ou on récupère la valeur, ...
- Exemple:

```
<td>Nom :</td>  
<td><input type="text" id="nom" class="c1" onblur="Mafct(this)"></td>
```

- **this**: c'est la zone de texte d'id nom, pour laquelle on applique l'événement blur

```
function Mafct(zone){  
  alert(zone.value)  
}
```

- Le paramètre **zone** va être remplacé par la zone de texte d'id nom, dans cet exemple