

Remplacer et supprimer un élément du DOM

Remplacer un nœud par un autre

- Pour remplacer un nœud, on utilisera la méthode **replaceChild()** qui va remplacer un certain nœud par un autre.
- Cette méthode va prendre en arguments le nœud de remplacement et le nœud qui doit être remplacé. Notez que si le nœud de remplacement existé déjà dans le DOM, il sera d'abord retiré de son emplacement d'origine.

Remplacer un nœud par un autre

- Exemple:

```
var cont=document.getElementById("div");  
// s étant un span déjà inclus dans le div  
var x=document.getElementById("s");  
var monp=document.createElement("p");  
monp.innerHTML="mon paragraphe";  
  
//replace x par monp  
cont.replaceChild(monp,x);
```

Supprimer un élément du DOM

- Pour supprimer totalement un nœud du DOM, on peut utiliser la méthode **removeChild()** qui va supprimer un nœud enfant passé en argument d'un certain nœud parent.

```
let b = document.body;  
let p1 = document.getElementById('p1');  
let p2 = document.getElementById('p2');  
  
//Supprime p1 du DOM et renvoie le noeud supprimé  
let eltDel = b.removeChild(p1);
```

Supprimer un élément du DOM

- On peut également utiliser la méthode **remove()** qui permet tout simplement de retirer un nœud de l'arborescence du DOM.

```
let b = document.body;  
let p1 = document.getElementById('p1');  
let p2 = document.getElementById('p2');  
  
//Supprime p2 du DOM  
p2.remove()
```

Les événements avec une autre méthode

- Dans cette méthode, on écrit rien dans le code HTML, mais on écrit l'événement dans le script comme suit:

```
var element= document.getElementById("nom")
element.onblur=function(){
    alert("j' ai quitté la zone")
}
```

C'est-à-dire que lorsque on a l'événement **blur** arrive sur **element**, cela déclenche une fonction anonyme permettant d'afficher un message

```
var element= document.getElementById("nom")
element.onblur=mafonction;

function mafonction(){
    alert("j' ai quitté la zone")
}
```

Dans ce cas, on a pas utilisée une fonction anonyme mais **une fonction nommé** puis appelée l'événement **blur** arrive sur **element**

addEventListener

- **addEventListener** est une méthode qui attache un gestionnaire d'événements à un élément. La syntaxe est comme suit:

```
Element.addEventListener( NomEvenement , maFonction )
```

- **Element**: est l'objet sur lequel on va appliquer l'événement.

addEventListener

- En utilisant une fonction anonyme:

```
var element= document.getElementById("nom")
element.addEventListener("blur",function(){
    document.body.style.backgroundColor="black"
});
```

- En utilisant une fonction nommée:

```
var element= document.getElementById("nom")
element.addEventListener("blur",mafonction)

function mafonction(){
    document.body.style.backgroundColor="black"
};
```

Si on veut réaliser un traitement sur le même élément, on peut utiliser le mot **this**

addEventListener

```
var t=document.querySelectorAll("#nom,#prenom,#age")

for(var i=0;i<t.length ;i++){
    t[i].addEventListener("blur",function(){
        this.style.backgroundColor="black"
    })
}
```

- Chaque zone de texte ayant successivement les id: nom, prenom et age sont concernés par l'événement blur
- Si un d'eux reçoit l'événement blur, il y aura un changement dans son style

forEach

- Cette méthode est utilisée pour les tableaux (c'est une écriture récente)

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

tab.forEach(function(el){
    alert(el)
})
```

- Pour chaque élément (**el**) du tableau **tab**, on affiche sa valeur

forEach

- Chaque zone de texte ayant successivement les id: nom, prenom et age sont concernés par l'événement blur
- Si un d'eux reçoit l'événement blur, il y aura un changement dans **son** style

```
var t=document.querySelectorAll("#nom,#prenom,#age")

t.forEach(function(el){
    el.onblur=function(){
        el.style.backgroundColor="black"
    }
})
```