

Javascript

Manipuler le DOM

Manipuler le DOM en Javascript

- DOM = Document object model (
https://fr.wikipedia.org/wiki/Document_Object_Model)

Manipuler le DOM en Javascript

- DOM = Document object model (https://fr.wikipedia.org/wiki/Document_Object_Model)
- Javascript permet de modifier, de créer, de supprimer, de cloner n'importe quel élément sur la page.
- Il peut ajouter, modifier et lire n'importe quel **attribut** d'un élément

Modification d'un élément

- On ne peut modifier un élément html que si il existe une fois la page chargée
- On peut modifier à volonté un élément

Modification d'un élément

- On ne peut modifier un élément html que si il existe une fois la page chargée
- On peut modifier à volonté un élément
- On peut modifier un élément créé avec Javascript

Exemple : Modification d'un élément

- Modification du contenu d'un élément div :

```
document.getElementById('monElementHTML').innerHTML = "Mon nouveau  
texte";
```

Ici, ajout ou modification du texte contenu dans le div appelé
"monElementHTML"

Création d'un élément

- On ne peut créer un élément html via Javascript qu'une fois la page complètement chargée.
- Un élément créé doit ensuite être intégrée à la page via une série d'instructions.

Création d'un élément

- On ne peut créer un élément html via Javascript qu'une fois la page complètement chargée.
- Un élément créé doit ensuite être intégrée à la page via une série d'instructions.
- Avant l'insertion, il est possible d'ajouter de nombreux attributs au nouvel élément.

Exemple : Création d'un élément

- Création puis insertion d'un nouveau DIV dans une page web.

```
var div = document.createElement("div");  
document.body.appendChild(div);
```

Ici, le nouvel élément est stocké dans la variable appelée "div"
On utilise l'instruction appendChild pour ajouter le nouveau div **à la suite** des autres éléments HTML de la page

Suppression d'un élément

- On ne peut supprimer un élément que si il existe dans la page et qu'il peut être identifié de manière précise (via son ID par exemple)

Suppression d'un élément

- On ne peut supprimer un élément que si il existe dans la page et qu'il peut être identifié de manière précise (via son ID par exemple)
- On peut supprimer un élément créé dynamiquement en javascript
- Si un élément contenant d'autres éléments est supprimé, les éléments contenu dans cet élément seront supprimés également

Exemple : Suppression d'un élément

- Suppression d'un div :

```
var divSupprime = document.getElementById ( 'monElementHTML' );  
document.body.removeChild (divSupprime) ;
```

Ici, l'élément que l'on souhaite supprimer est identifié puis stocké dans une variable.

On utilise l'instruction `removeChild` pour retirer l'élément de la page web

Modification des attributs d'un élément

- On peut modifier ou ajouter un nouvel attribut à un élément
- Il est possible de modifier des sous attributs d'un attribut directement

Exemple : Modification de l'attribut Width d'une image

- Modification de la largeur d'une image :

```
var monImage = document.getElementById('monImage');  
monImage.width = 450+"px";
```

Ici, notre image est identifié par son ID “monImage”, l'élément est stocké dans la variable “monImage”

Puis l'attribut “width” de notre élément est modifié pour prendre la valeur “450px” soit une largeur de 450 pixels

Modification de sous attributs, cas de l'attribut Style

- Certains attributs comme "Style" possèdent de sous attributs
- Il est possible de modifier des sous attributs d'un attribut directement

Exemple : Modification de sous attributs, cas de l'attribut Style

- Modification de la largeur d'une image en utilisant le sous attribut "width" de l'attribut "Style"

```
monImage.style.width = "100px";
```

Modification de la largeur d'une image en utilisant le sous attribut "width" de l'attribut "style", la nouvelle largeur de notre image va être de 100 pixels.

<https://jsfiddle.net/rz8Lpq9u/2/>

Conclusion

- L'attribut style possède de nombreux sous attributs, à vous de les découvrir !
- Modifier le style des éléments en javascript permet de créer des applications web complexes (jeux video, animations, présentations etc...)

Exemple d'une condition en Javascript

