

420JJAJQ

Programmation mobile

DOM



CÉGEP DE JONQUIÈRE

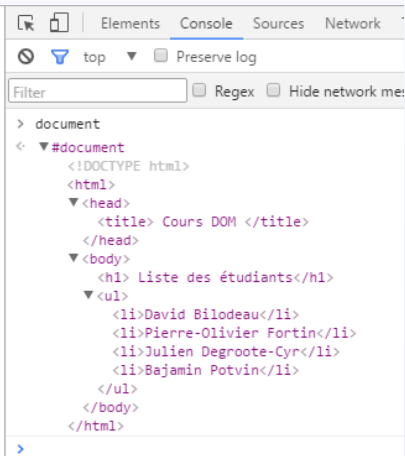
Département d'Informatique

MT. Moutacalli

DOM, Modèle Objet de Document, est une structure d'arbre, présente dans la mémoire du navigateur, qui représente le document.

Liste des étudiants

- David Bilodeau
- Pierre-Olivier Fortin
- Julien Degroote-Cyr
- Bajamin Potvin



DOM : API (Application Programming Interface) qui nous fournit les moyens d'interagir avec l'arbre-document.

C'est une recommandation du W3C pour gérer le contenu de documents XML et HTML en particulier.

- Il fournit une représentation structurée du document ;
- Il codifie la manière dont un script peut accéder à cette structure.

Récupérer le fichier index.html et dans la console:

- 1 Afficher le titre de la page
- 2 Modifier le titre de la page à : DOM

- 1 Afficher le titre de la page :
`document.title`
- 2 Modifier le titre de la page à : DOM
`document.title = "DOM"`

- 1 **getElementById** permet de sélectionner un élément par son *id* dans une page.
Créer un script qui permet d'ajouter "Les" avant le titre.
- 2 **getElementsByClassName** permet de sélectionner tous les éléments d'une classe.
- 3 **getElementsByTagName** permet de sélectionner tous les éléments avec la même Tag.

Créer un script qui permet d'ajouter "M." avant tous les étudiants.

- **getElementById** permet de sélectionner un élément par son *id* dans une page.

Créer un script qui permet d'ajouter "Les" avant le titre:

Dans *index.html* ajouter:

```
<script src = "modif.js"> </script>
<h1 id="t1"> Liste des étudiants</h1> et
<button type="button"
onclick="modifier()">Modifier</button>
```

Dans *modif.js* ajouter:

```
function modifier(){
    var titre = document.getElementById('t1');
    titre.innerHTML = "Les " + titre.innerHTML;
}
```

- **getElementsByClassName** permet de sélectionner tous les éléments d'une classe.
Créer un script qui permet d'ajouter "M." avant tous les étudiants.:

Dans *index.html*, pour chaque étudiant, ajouter:

```
<li class = "et">David Bilodeau</li>
```

Dans *modif.js* :

```
function modifier(){  
    var etuds =  
    document.getElementsByClassName("et");  
    for (var i = 0; i < etuds.length; i++)  
        etuds[i].innerHTML = "M. " + etuds[i].innerHTML;  
}
```


- **getElementsByTagName** permet de sélectionner tous les éléments avec la même Tag.

Créer un script qui permet d'ajouter "M." avant tous les étudiants.:

Dans *modif.js* :

```
function modifier(){  
    var etuds =  
        document.getElementsByTagName("LI");  
    for (var i = 0; i < etuds.length; i++)  
        etuds[i].innerHTML = "M. " + etuds[i].innerHTML;  
}
```

Pour éviter de changer tous les "li" on peut sélectionner d'abord un "ul" par son id:

```
var etuds = document.getElementById("e").getElementsByTagName("LI");
```

Les récents navigateurs offrent deux nouvelles fonctions de sélection:

- 1 **querySelector** permet de sélectionner un élément, comme dans le CSS, par son *id* avec "#id", par le nom de la classe avec ".class" ou directement par la Tag: "Tag"

```
var titre = document.querySelector("#t1");  
var titre = document.querySelector(".et");
```

- 2 **querySelectorAll** permet de sélectionner plusieurs éléments:

```
var etuds = document.querySelectorAll(".et");  
var etuds = document.querySelectorAll("LI");  
var etuds = document.querySelectorAll("#e LI");
```

On peut créer du nouveau HTML contenu via Javascript:

- 1 Créer l'élément en utilisant la fonction:
document.createElement(nom_Balise)
- 2 Spécifier l'emplacement de l'élément créé en étant fils de...:
parent.appendChild(Fils)
parent.insertBefore(Fils,frère)

- Ajouter le titre "h2" 420JAJQ à la fin du document
- Ajouter le avant le bouton

Création:

```
var p=document.createElement("h2");  
p.innerHTML="420JJAJQ";
```

- L'ajouter à la fin du document:

```
document.body.appendChild(p);
```

- L'ajouter avant le bouton :

```
var bouton = document.querySelector("#b");  
document.body.insertBefore(p,bouton);
```

Jusqu'à présent, toutes les fonctions que nous avons créées sont lancées à partir d'un bouton.

Javascript utilise les événements pour détecter les actions sur des objets et lancer la fonction appropriée.

Pour cela, on utilise un **gestionnaire d'événement** avec la syntaxe suivante:

onEvenement="Action_Javascript_ou_Fonction()";

Ex: *onclick="modifier()"*

Les événements les plus utilisés:

| Événement | Description |
|-------------|--|
| onchange | Au changement d'un élément HTML |
| onclick | Au click de l'utilisateur sur un élément HTML |
| onmouseover | l'utilisateur survole avec la souris un élément HTML |
| onmouseout | l'utilisateur déplace la souris dehors un élément HTML |
| onkeydown | l'utilisateur appuie sur une touche clavier |
| onload | Le navigateur termine le chargement de la page |

Modifier le script attaché à la page index.html de tel sorte que si on survole le titre il devient : Nouvelle Liste et quand la souris n'est pas sur le titre il contient Liste des Étudiants.

Modifier le script attaché à la page index.html de tel sorte que si on survole le titre il devient : Nouvelle Liste et quand la souris n'est pas sur le titre il contient Liste des Étudiants:

```
document.querySelector("#t1").onmouseover=function(){  
  this.innerHTML = "Nouvelle Liste";  
}  
document.querySelector("#t1").onmouseout=function(){  
  this.innerHTML = "Liste des Étudiants";  
}
```

Attention: afin que le script puisse trouver un élément, il doit être appelé, dans la page html, après lui.

Pour attribuer un événement à un objet dans un script, on peut aussi utiliser **addEventListener**.

Syntaxe: **element.addEventListener("événement",fonction);**

Modifier, en utilisant `addEventListener`, le contenu du titre à Nouvelle Liste au click.

Modifier, en utilisant `addEventListener`, le contenu du titre à Nouvelle Liste au click:

```
document.querySelector("#t1").addEventListener("click",modifier);  
function modifier(){  
  this.innerHTML = "Nouvelle Liste";  
}
```

Pour supprimer un élément, on utilise **parentNode.removeChild**.

Syntaxe: **element.parentNode.removeChild(element);**

Ex: Au click sur le bouton, supprimer le titre:

Dans la fonction du scripte :

```
var titre = document.querySelector("#t1");  
titre.parentNode.removeChild(titre);
```

- Ajouter des étudiants à partir d'une zone de texte (ne pas permettre des noms vides).

Le bouton devient Ajouter au lieu de modifier.

- Supprimer un étudiant si on click dessus.