



Révisions JavaScript

Vanilla JS

Dans ce TD, vous allez revoir rapidement quelques bases du langage JavaScript et quelques éléments de DOM.

Exercice 1. Liens Il existe une très grande quantité de sites sur JavaScript mais on pourra consulter avec profit :

- <https://developer.mozilla.org/fr/docs/Web/JavaScript>
- https://developer.mozilla.org/fr/docs/Web/JavaScript/Une_r%C3%A9introduction_%C3%A0_JavaScript
- <http://javascript.info/>
- <http://html5demos.com/>

Et vous pouvez lire l'excellente série *You don't know JS* : <https://github.com/getify/You-Dont-Know-JS>

Exercice 2. Prise en main - Débogage

- Créez un répertoire JS dans votre home
- placez vous à l'intérieur et editez une première page *exo2.html* toute simple :

```
1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Essais JS</title>
6 </head>
7 <body>
8   <script>
9     console.log("<h1>Généré en JavaScript</h1>");
10  </script>
11 </html>
```

- Ouvrez la console JS et observez
- et avec un `confirm()` :

```
1 window.confirm("<h1>Généré en JavaScript</h1>");
```

- L'appel à `window` est-il obligatoire ?

- Les ; sont-ils obligatoires en fin de ligne en JS ?
- Le formatage html est-il utile dans la console ou dans une alert ?
- Pour le debogage on utilisera l'une de ces méthodes en préférant la console.

Exercice 3. Un Objet en JavaScript

Construisons un objet simple :

```
1 let etu = new Object();
2 etu.nom = "Alpha";
3 etu.prénom = "Julie";
4 console.log(etu);
5 console.log(etu['nom']);
6 console.log(etu['prénom']);
```

- Ajoutez l'affichage de leurs types.
- On peut parcourir les propriétés d'un objet :

```
1 for (p in etu){
2     console.log(p + " " + etu[p]);
3 }
```

- On aurait pu aussi déclarer l'objet à la manière d'un dictionnaire Python :

```
1 etudiant={
2     'prénom': 'Jean-Claude',
3     'nom': 'Green',
4     'age': 22
5 };
6 console.log(etudiant);
```

- Affichez les clefs et les valeurs de ce dictionnaire.

Il est aussi possible de déclarer les variables en JS avec const ou var . Voir le cours pour étudier la différence ...

Exercice 4. Manipulation du DOM

Dans cet exercice, on va voir comment créer un document au moment du chargement de la page. Le principe va être de créer les nœuds du document (i.e. les <h1>, <section>, etc.) puis les relier entre eux pour créer l'arborescence de la page web.

Voici deux fonctions utiles pour créer un nœud :

- `document.createElement(tag)`
- `document.createTextNode(value)`

Et une fonction qui permet de relier les nœuds entre eux (il y en a d'autres..) :

- `parent.appendChild(elem)`

Par exemple, on pourra créer un titre contenant le texte "Mon Titre" en faisant :

```
1 let titre=document.createElement("h1");
2 let texte=document.createTextNode("Mon Titre");
3 titre.appendChild(texte);
4 document.body.appendChild(titre);
```

- 4.1 Essayez ce code sur une page vide au travers de la console web.
- 4.2 Que se passe-t-il si vous faites ensuite : `titre.style.color="blue"` ?
- 4.3 Ajoutez un paragraphe contenant un texte.
- 4.4 Mettez tout ceci dans un fichier html. Vérifiez que ça fonctionne.
- 4.5 Comment faire pour avoir un mot du paragraphe en rouge ?
- 4.6 Ajoutez un fichier css avec une classe *bleu* (le texte correspondant est en bleu) et une classe *souligne* (le texte correspondant est souligné). En utilisant la propriété `classList` des noeuds, mettez le titre en bleu et souligné.
- 4.7 Créez une fonction qui prend en entrée un noeud et un dictionnaire, et ajoute à ce noeud la liste html (i.e. `` et ``) dont chaque item est une clef du dictionnaire suivie de la valeur correspondante. Essayez cette fonction.

Exercice 5. Les selecteurs CSS

On a vu déjà plusieurs manières de sélectionner des éléments particuliers d'une page Web en JS :

- Par Id avec `getElementById()`
- Par nom de tag avec `getElementsByTagName()`
- Par nom de classe avec `getElementsByClassName()` qui prend en entrée une ou plusieurs classes de style

A ces différentes manières, il faut en ajouter une très importante qui est d'utiliser les sélecteurs CSS.

Exercice 6. Pratique des selecteurs CSS en JavaScript

En JavaScript, nous pouvons utiliser les selecteurs CSS via les fonctions :

- `querySelectorAll(CSSsel)` qui renvoie les éléments correspondant au selecteur CSS passé en argument
- `querySelector(CSSsel)` qui renvoie le premier élément trouvé dans le document qui correspond au selecteur.

Considérons par exemple la page suivante :

```
1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Selection de liens en JS</title>
6 </head>
7 <body>
8 <header>
9 <nav id="menu">
10  <ul>
11    <li> <a href="page1.html"> Page 1 </a> </li>
12    <li> <a href="page2.html"> Page 2 </a> </li>
13    <li> <a href="page3.html"> Page 3 </a> </li>
14    <li> <a href="page4.html"> Page 4 </a> </li>
15  </ul>
16 </nav>
17 <section>
18 <article class="my">
19 <div id="un">
20 Contenu Bla bla ...
21 </div>
22 <div id="deux">
23 Contenu Bli bli ...
24 </div>
25 </article>
26 </section>
27 </body>
28 </html>
```

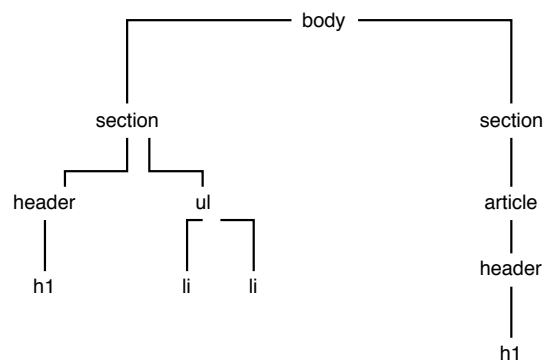
Pouvez-vous y sélectionner à l'aide d'un script JS :

1. Tous les éléments de la page
2. Tous les liens de la page
3. Tous les liens dans une liste de type ul

4. Le dernier lien dans le menu
5. L'article de classe "my"
6. La deuxième div de l'article de classe "my"

Exercice 7. Le DOM : parcours

7.1 Créez une page html avec juste un squelette correspondant à l'arborescence suivante :



De plus :

- donnez l'id "titre1" au titre h1 de la première section et "titre2" au deuxième titre,
- mettez les deux titres h1 dans une classe "titre",

7.2 Dans la console du navigateur, essayez d'accéder à la liste des enfants du body : `document.body.children` . Pouvez vous parcourir cette liste avec la syntaxe `for/of` ?

7.3 Affichez les balises de niveau 2 (les enfants des enfants de body).

7.4 Ajoutez à toutes les balises de niveau 2 un texte étant le contenu de l'attribut `nodeName` du noeud.

7.5 à l'aide des méthodes déjà citées :

- `getElementById`
- `getElementsByTagName`
- `getElementsByName`
- `getElementsByClassName`
- `querySelector` et `querySelectorAll`

Affichez les noms des parents des éléments appartenant à la classe "titre".

7.6 Quelle instruction javascript faut-il écrire pour supprimer tous les éléments appartenant à la classe titre ? (On peut supprimer un enfant d'un noeud à l'aide de la méthode `removeChild`)

7.7 Quel type d'objet renvoie la méthode `getElementById` ? Et la méthode `getElements-ByTagName` ? Pourquoi est-ce différent ?

7.8 Comment récupérer dans un tableau tous les éléments de type `<p>` dont un des parents est une section ?

Exercice 8. JQuery

Reprendre à présent les exercices précédents en utilisant JQuery à la place des méthodes natives de JavaScript.