

Les selecteurs CSS en JavaScript

Vanilla JS et JQuery

Dans ce TD, vous allez vous familiariser avec les selecteurs CSS et leur utilisation à partir de JavaScript.

Exercice 1. 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 selecteurs CSS. Petit rappel sur les selecteurs CSS :

- `#main` pour sélectionner un élément dont l'ID est *main*
- `a` pour sélectionner tous les liens
- `.rouge` pour sélectionner tous les éléments comportant la classe de style *rouge*
- `input[type="search"]` pour sélectionner les champs input ayant l'attribut *type* valant *search*
- `*` pour sélectionner tous les éléments de la page

On peut aussi faire des sélections plus complexes comme :

- `div#main` pour sélectionner tous les éléments enfants de la div dont l'ID est *main*
- `li a` pour sélectionner les liens qui sont dans des *ul* (descendants d'un *li* dans l'arborescence)
- `div > a` pour ne sélectionner que les *a* qui sont enfants (directs) d'une *div*

Consultez : <http://www.w3schools.com/cssref/trysel.asp> pour vous entraîner un peu plus. On peut utiliser les pseudos classes (*visited*, *hover*, *first-letter*, sélectionner le premier ou dernier élément, des frères et soeurs (*siblings*), etc.

Exercice 2. 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.

Attention, pour des raisons de protection des données, beaucoup de navigateurs refusent l'usage des pseudos-classes `:link` ou `:visited` qui exposent l'historique de la navigation. 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>
```

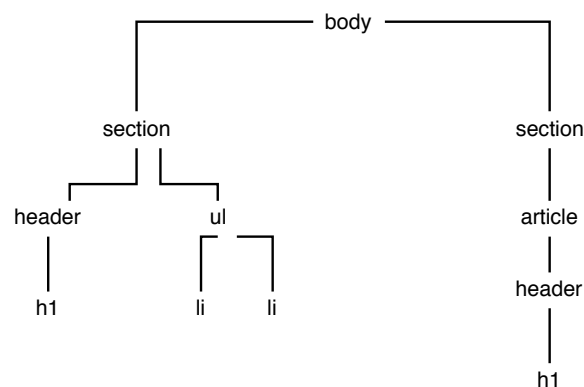
Pouvez-vous y sélectionner à l'aide d'un script JS :

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 3. Le DOM : parcours

3.1 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",

3.2 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` ?

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

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

3.5 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".

3.6 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`)

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

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

Exercice 4. Changement dynamique de feuille de style

- Comment sélectionner tous les liens vers des feuilles de styles dans un document ?
- Supposons une page avec 1 seule feuille de style, `style1.css`. Une autre de style `style2.css` devra la remplacer au click sur un bouton. Ecrivez la fonction JS correspondante.

Exercice 5. JQuery

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