

## Atelier 12 Javascript 2/2 : Navigateur

### Introduction :

Dans cette Atelier nous continuons avec la suite des exercices Javascript.

### Navigateur 1 :

#### Enfants DOM

Regardez cette page :

```
1 <html>
2 <body>
3   <div>Users:</div>
4   <ul>
5     <li>John</li>
6     <li>Pete</li>
7   </ul>
8 </body>
9 </html>
```

Pour chacun des éléments suivants, donnez au moins un moyen d'y accéder :

- Le noeud `<div>` du DOM ?
- Le noeud `<ul>` du DOM ?
- Le deuxième `<li>` (avec Pete) ?

On peut accéder au nœud `<div>` en passant par `childNodes` :

```
▼ childNodes: NodeList(5)
  ▶ 0: text
  ▶ 1: div
  ▶ 2: text
  ▶ 3: ul
  ▶ 4: text
```

Pareille pour `<ul>`

```
▼ childNodes: NodeList(5)
  ▶ 0: text
  ▶ 1: div
  ▶ 2: text
  ▶ 3: ul
  ▶ 4: text
```

Pour le deuxième `<li>` on le trouve dans `parentNode` :

```
▼ parentNode: document
  ▶ location: Location {ancestorOrigins: DOMStringList(1), href: "file:///C:/Testjavascript/testscript.htm", ...}
  ▶ activeElement: body
  ▶ activeViewTransition: null
  ▶ adoptedStyleSheets: Proxy(Array) {}
  ▶ alinkColor: ""
  ▼ all: HTMLAllCollection(7)
    ▶ 0: html
    ▶ 1: head
    ▶ 2: body
    ▶ 3: div
    ▶ 4: ul
    ▶ 5: li
    ▶ 6: li
```

## La question des frères et sœurs

Si `element` – est un nœud élément arbitraire du DOM ...

- Est-il vrai que `elem.lastChild.nextSibling` est toujours `null` ?
- Est-il vrai que `elem.children[0].previousSibling` est toujours `null` ?

- Oui `elem.lastChild.nextSibling` est toujours `null`
- Non `elem.children[0].previousSibling` n'est pas toujours `null`

# Sélectionner toutes les cellules diagonales

Écrivez le code pour colorer toutes les cellules du tableau diagonal en rouge.

Vous devrez obtenir toutes les diagonales `<td>` de la `<table>` et les colorer en utilisant le code :

```
1 // td doit être la référence à la cellule du tableau
2 td.style.backgroundColor = 'red';
```

Le résultat devrait être :

1:1	2:1	3:1	4:1	5:1
1:2	2:2	3:2	4:2	5:2
1:3	2:3	3:3	4:3	5:3
1:4	2:4	3:4	4:4	5:4
1:5	2:5	3:5	4:5	5:5

## Recherche d'éléments

Voici le document avec le tableau et formulaire

Comment trouver ?...

1. Le tableau avec `id="age-table"`.
2. Tous les éléments `label` dans ce tableau (il devrait y en avoir 3).
3. Le premier `td` dans ce tableau (avec le mot "Age").
4. Le `form` avec `name="search"`.
5. Le premier `input` dans ce formulaire.
6. Le dernier `input` dans ce formulaire.

Ouvrez la page [table.html](#) dans un onglet à part et utilisez les outils du navigateur pour cela.

### 1. Html → Body → form → table#age-table

```
hidden: false
id: "age-table"
inert: false
```

### 2.

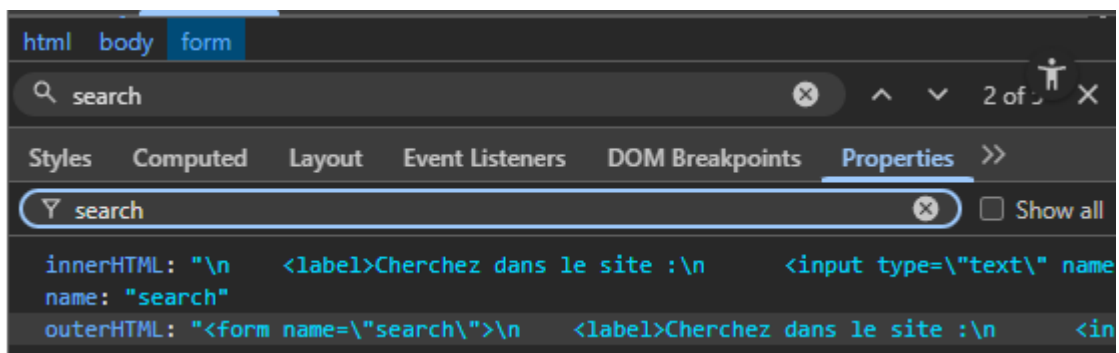
```
html body form table#age-table tbody tr td#age-list label
```

Les éléments de cette catégories sont : moins de 18 | 18-50 | plus de 50.

3. `html body form table#age-table tbody tr td#age-list`

Le premier td est age-list

4.



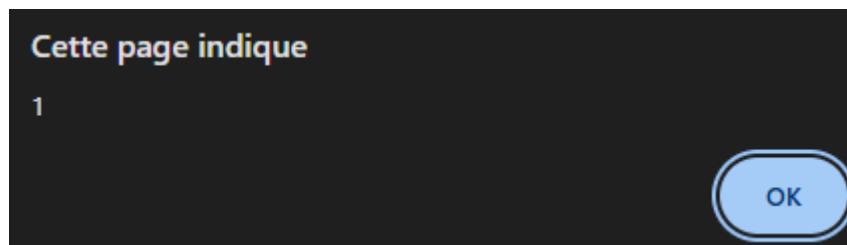
5.

## Compter les descendants

Qu'affiche le script ?

```
1 <html>
2
3 <body>
4   <script>
5     alert(document.body.lastChild.nodeType);
6   </script>
7 </body>
8
9 </html>
```

Le script affiche 1 :

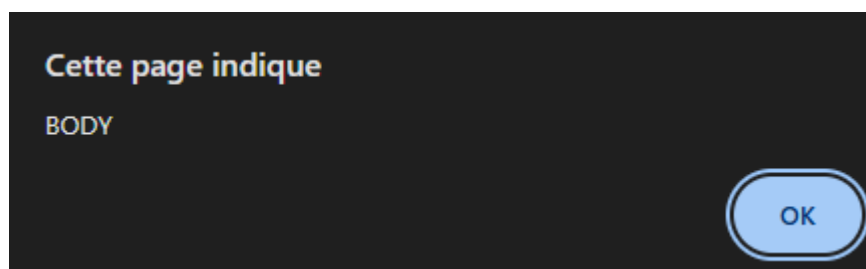


## Balise dans le commentaire

Qu'affiche ce code ?

```
1 <script>
2   let body = document.body;
3
4   body.innerHTML = "<!--" + body.tagName + "-->";
5
6   alert( body.firstChild.data ); // Qu'est ce qu'il y a ici ?
7 </script>
```

Le code affiche :



## Où est le "document" dans la hiérarchie ?

À quelle classe appartient le `document` ?

Quelle est sa place dans la hiérarchie DOM ?

Hérite-t-il de `Node` ou `Element`, ou peut-être de `HTMLElement` ?

## Navigateur 2 :

### Obtenez l'attribut

Écrivez le code pour sélectionner l'élément avec l'attribut `data-widget-name` dans le document et pour lire sa valeur.

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5   <div data-widget-name="menu">Choose the genre</div>
6
7   <script>
8     /* your code */
9   </script>
10 </body>
11 </html>
```

## Rendre les liens externes orange

Mettez tous les liens externes en orange en modifiant leur propriété `style`.

Un lien est externe si :

- Son `href` contient `://`
- Mais ne commence pas par `http://internal.com`.

Exemple :

```
1 <a name="list">the list</a>
2 <ul>
3   <li><a href="http://google.com">http://google.com</a></li>
4   <li><a href="/tutorial">/tutorial.html</a></li>
5   <li><a href="local/path">local/path</a></li>
6   <li><a href="ftp://ftp.com/my.zip">ftp://ftp.com/my.zip</a></li>
7   <li><a href="http://nodejs.org">http://nodejs.org</a></li>
8   <li><a href="http://internal.com/test">http://internal.com/test</a></li>
9 </ul>
10
11 <script>
12   // setting style for a single link
13   let link = document.querySelector('a');
14   link.style.color = 'orange';
15 </script>
```

the list

- <http://google.com>
- </tutorial.html>
- <local/path>
- <ftp://ftp.com/my.zip>
- <http://nodejs.org>
- <http://internal.com/test>

```
<script>
let links = document.querySelectorAll('a');
for (let link of links) {
  let href = link.getAttribute('href');
  if
(href && href.includes('://') && !href.startsWith('http://internal.com')) {
    link.style.color = 'orange';
  }
}
</script>
```

## createTextNode vs innerHTML vs textContent

Nous avons un élément DOM vide `elem` et une chaîne de caractères `text`.

Lesquelles de ces 3 commandes feront exactement la même chose ?

1. `elem.append(document.createTextNode(text))`
2. `elem.innerHTML = text`
3. `elem.textContent = text`

La 1 et la 3 feront la même chose

### Effacer l'élément

Créez une fonction `clear(elem)` qui supprime tout de l'élément.

```
1 <ol id="elem">
2   <li>Hello</li>
3   <li>World</li>
4 </ol>
5
6 <script>
7   function clear(elem) { /* votre code */ }
8
9   clear(elem); // efface la liste
10 </script>
```

```
function clear(elem) {
  elem.replaceChildren();
}
```



## Pourquoi "aaa" reste-t-il ?

Dans l'exemple ci-dessous, l'appel `table.remove()` supprime le tableau du document.

mais si vous l'exécutez, vous pouvez voir que le texte "aaa" est toujours visible.

Pourquoi cela se produit-il ?

```
1 <table id="table">
2   aaa
3   <tr>
4     <td>Test</td>
5   </tr>
6 </table>
7
8 <script>
9   alert(table); // la table, comme il se doit
10
11   table.remove();
12   // pourquoi y a-t-il encore "aaa" dans le document ?
13 </script>
```