



Intro à JavaScript

Vanilla JS

Dans ce TD, vous allez commencer à vous familiariser avec le langage JavaScript les outils 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/>

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     alert("<h1>Généré en JavaScript</h1>");
10  </script>
11 </html>
```

- puis affichons le même message dans la console JS :

```
1 console.log("<h1>Généré en JavaScript</h1>");
```

- Ouvrez la console JS en tapant CTRL MAJ K (ou ALT CMD K sous Mac) dans les outils de développement de votre navigateur et allez consulter le résultat
- 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 débogage on utilisera l'une de ces méthodes en préférant la console.

Exercice 3. Les types de base en JavaScript

- Tapez le code suivant dans une page *exo3.html*

```
1 const s = "Bonjour";
2 console.log("s est du type: " + typeof s);
3 const i = 2;
4 console.log("i est du type: " + typeof i);
5 const f = 0.003;
6 console.log("f est du type: " + typeof f);
7 const b = true;
8 console.log("b est du type: " + typeof b);
```

- Puis ajoutez des opérations problématiques :

```
1 const infini = 2.0/0;
2 console.log(infini);
3 let grand = Math.pow(2,10000);
4 console.log(grand);
5 let impossible = "azerty"/2;
6 console.log(impossible);
```

affichez les résultats et les types obtenus. Commentez.

- Construisons à présent une fonction et un objet simple :

```
1 let inc = function(x) {return x+1 ;}
2 console.log(inc);
3
4 let etu = new Object();
5 etu.nom = "Alpha";
6 etu.prénom = "Julie";
7 console.log(etu);
8 console.log(etu['nom']);
9 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 clé du dictionnaire suivie de la valeur correspondante. Essayez cette fonction.

Exercice 5. Fonction mystère

```
1  function mystery() {
2    const s = Math.random().toString(36).slice(2);
3    return s.length > 10 ? s : mystery();
4  }
```

Expliquez son fonctionnement en consultant la documentation de JavaScript sur MDN.

Exercice 6. Affichez/Masquer un élément dans la page

— Créez un fichier `mapage.html` dans votre home :

```
1  <!doctype html>
2  <html>
3  <head>
4    <meta charset="utf-8">
5    <title>Afficher/Masquer en JS</title>
6  </head>
7  <body>
8    <header>
9      <h1>Exercice 3</h1>
10     <p id="pp"> paragraphe à masquer ... </p>
11   </header>
12   <input type="button" id="bt" value="Masquer" onclick="
13     cache()"/>
13 </html>
```

- Puis on va sélectionner le paragraphe de l'entête avec JS pour le masquer en ajoutant une section `<script>` à la fin de la page :

```
1 <script>
2 function cache(){
3     const pp = document.getElementById("pp");
4     pp.hidden=false;
5 }
6 </script>
```

- Complétez maintenant ce script pour que le bouton puisse alternativement masquer puis cacher le paragraphe.

Exercice 7. Changer de couleur

On veut changer la couleur de fond d'une page au click sur le bouton. On sélectionnera pour cela l'élément `body` en changeant sa propriété `background` :

```
1 function changeBackground(color) {
2     document.body.style.background = color;
3 }
```

Comment faire en sorte que la couleur soit aléatoirement générée à chaque nouveau click ?

Exercice 8. Selection par Tag

- On considère à présent une page contenant quelques liens comme :

```
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     <ul>
9         <li> <a href="page1.html"> Page 1 </a> </li>
10        <li> <a href="page2.html"> Page 2 </a> </li>
11        <li> <a href="page3.html"> Page 3 </a> </li>
12        <li> <a href="page4.html"> Page 4 </a> </li>
13    </ul>
```

- En utilisant la fonction `getElementsByTagName`, construisons la liste des liens de la page en utilisant cette fonction :

```
1     function selectLiens(){
2         const liens = document.getElementsByTagName("a");
3         return liens;
4     }
```

- Lancez cette fonction dans un script placé en fin de page qui comptera combien de liens comporte la page, affichera ce nom dans le document et ouvrira une popup JavaScript à l'aide de `alert()` en y affichant les liens trouvés en parcourant liens à l'aide d'un `for`.
- Que se passe-t-il si on lance ces instructions de la page sur l'évènement `onload`? Faire ceci en plaçant les instructions dans un bloc :

```
1     <script>
2         window.onload = function(){
3             // placez ici votre code JS
4         }
5     </script>
```

- Une solution simple pour que cet évènement se lance uniquement quand le document est chargé (ou tout au moins quand le DOM est en place) est :

```
1     <script>
2         (
3             function() {
4                 // placez ici votre code JS
5             }
6         ) ();
7     </script>
```

Est-ce que les liens s'affichent bien dans ce cas? Vous pouvez commencer à voir que JavaScript n'est pas du tout Java et que sa syntaxe peut s'avérer un peu déroutante. Pour essayer de comprendre, ce bloc équivaut à la séquence suivante en nommant la fonction :

```
1     <script>
2         let load =function() {
3             // placez ici votre code JS
4         }
5         load();
6     </script>
```

Exercice 9. Selection par nom de classe et modification d'attributs

- Prenez une page html avec une image de classe de style beau.

- Sélectionnez l'image par sa classe et changez d'image lorsqu'on cliquera sur elle en modifiant son attribut `src`.
- Comment changer le style d'un paragraphe à l'aide de JavaScript ?
- Comment changer le texte du paragraphe ?

Exercice 10. Récupération des infos client

Ecrire un petit programme en JavaScript lancé depuis une page HTML permettant de récupérer toutes les infos "client" :

- Navigateur, OS, langue, résolution écran.
- Date de dernière modification du document et date de chargement.
- liste des plugins installés, etc.