

TD 5 : JavaScript

28 octobre 2015

Objectif: Ce TD a pour but de découvrir le langage JavaScript. À l'issue de ce TD vous devrez savoir :

- encapsuler des scripts dans une page HTML ;
- réaliser des fonctions ;
- modifier le contenu d'un élément HTML ;
- modifier les attributs d'un élément HTML ;
- modifier le style (CSS) d'un élément HTML.

1 Faisons les présentations

JavaScript est un langage de scripts principalement utilisé dans (ou pour) des pages web. Il permet de faire une multitude de choses côté client : modifier une page à la volée, interagir avec l'utilisateur, etc.

Pour qu'un navigateur puisse différencier le code JavaScript du reste du document HTML, on l'encapsule dans une balise spécifique :

```
<script>
  // Ici on écrira notre script.

  /*
  Au fait, la ligne ci-dessus est un commentaire.
  Et même ces deux lignes le sont !
  */
</script>
```

Dans une même page HTML on peut insérer plusieurs scripts. De plus, on peut décider de les placer n'importe où dans le document.

Malgré cette liberté, il est de bonne pratique de placer tout le code dans un même script situé au bas de l'élément `body`. Cela permet d'améliorer le chargement de la page car, de cette façon, l'affichage des éléments HTML n'est pas bloqué par le chargement des scripts.

Par exemple :

```

<!DOCTYPE html>
<html>
  [...]
  <body>
    [...]
    <script>
      // On ouvre une fenêtre pop-up
      alert("Bonjour. Ca va ?");
    </script>
  </body>
</html>

```

Comme en CSS, on peut également (et on doit, généralement) externaliser le code JavaScript, qui sera contenu dans un (ou, si nécessaire, plusieurs) fichier(s) « *.js ».

Par exemple, dans le code HTML on écrira :

```
<script src="tous_mes_scripts.js"></script>
```

et dans le fichier « tous_mes_scripts.js » :

```
alert("Bonjour. Ca va ?");
```

► **Exercice 1 :** Créez une page javascript.html et un fichier scripts.js.

À l'aide des exemples précédents, modifiez les deux fichiers pour faire en sorte qu'en ouvrant la page HTML, deux fenêtres d'alerte contenant les messages suivants soient affichées :

- « Vous êtes le millième à visiter cette page... » ;
- « Non, je rigole : je le dis à tous ceux qui entrent ! ».

► **Exercice 2 :** Modifiez le fichier scripts.js en ajoutant la ligne suivante :

```
alert(2+2); // sans guillemets
```

Quel est le dernier message d'alerte affiché ?

Avec JavaScript on peut écrire directement dans une page HTML en utilisant la méthode `document.write()`.

Par exemple, le code :

```
document.write("<p>Alouette, gentille alouette, \  
<br>Alouette, je te plumerai.</p>");
```

affiche le refrain d'une fameuse comptine française. Remarquez que pour diviser en deux lignes une chaîne de texte (le texte entre guillemets) trop longue, on a utilisé un antislash.

2 Quelle heure est-il, Mademoiselle ?

Demander l'heure n'est pas l'apanage des séducteurs, surtout juste après le passage à l'heure d'hiver (ou en Turquie cette semaine). Le code :

```
Date();
```

nous permet de récupérer la date courante. On peut aussi décider de définir une nouvelle date en utilisant une des syntaxes suivantes (et d'autres encore) :

```
uneDate4 = new Date("1977-05-25");
uneDate5 = new Date("21/05/1980");
uneDate6 = new Date("1983/05/25");
uneDate1 = new Date("May 19 1999");
uneDate2 = new Date("16 May 2002");
uneDate3 = new Date("2005 May 19");
uneDate7 = new Date("December 18 2015");
```

Comme vous pouvez le remarquer, on peut écrire la date dans la forme courte (que des chiffres) soit “à l’anglaise” (MM-DD-YYYY) soit “à la chinoise” (YYYY-MM-DD), mais pas “à la française” (DD-MM-YYYY)!

► **Exercice 3 :** Définissez dans votre fichier `scripts.js` la date « `maDateDeNaissance` ». Ensuite, affichez sur l’écran la phrase « Je célèbre mon anniversaire le *DD* du mois *MM*. », où *DD* et *MM* sont respectivement le jour et le mois de la variable `maDateDeNaissance` (utilisez les méthodes `getDate()` et `getMonth()`).

Le mois que vous avez inséré, correspond-il au mois affiché? Pourquoi ce décalage?

3 Du déjà vu

La syntaxe de JavaScript ressemble à s’y méprendre à celle du langage C.

On peut définir une variable (à l’aide du mot-clé `var`), l’initialiser, l’utiliser pour des opérations arithmétiques, etc.

Par exemple, le code :

```
var n = 4;
var m;
n++;
m = (n + 3) * 2.5;

document.write(m)
```

affichera « 20 » dans notre page web.

Ainsi, tout ce qui est test et boucle est identique au langage C.

Par exemple, on peut écrire une instruction conditionnelle comme dans le code :

```
if (age<26 && etudiant==true) {
    monMessage = "Vous avez droit au tarif jeune étudiant.";
} else if (age<26) {
    monMessage = "Vous avez droit au tarif jeune.";
} else {
    monMessage = "Vous n'avez droit à aucun tarif réduit.";
}
```

ou une boucle `for` comme dans le code :

```
var comptine = "";
for (var i = 1; i < 100; i++) {
  comptine += "<p>" + i + " éléphants qui se balançaient \
sur une toile d'araignée.</p>";
}
```

De même pour les boucles `while` et `do/while` (voir http://www.w3schools.com/js/js_loop_while.asp pour plus de détails).

Comme en C, en JavaScript on peut définir des *functions*, c'est-à-dire des blocs de code conçus pour effectuer des tâches particulières.

Par exemple :

```
function maPresqueSomme(a,b) {
  var somme = a + " plus " + b + " font " + a+b;
  return somme;
}
```

Pour des raisons historiques, on écrit le code JavaScript en utilisant la notation *lowerCamel-Case* (casse “en chameau” avec la première lettre en minuscule) : `maSomme`, `monJoliScript`, `uneDate`, etc.¹.

► **Exercice 4** : En vous aidant des vos connaissances en C (et de http://www.w3schools.com/js/js_switch.asp, si nécessaire), écrivez une fonction « `afficheDateEnFrancais` » qui prend en paramètre une date et l’affiche selon le standard français.

Par exemple, la date « `Wed Oct 28 2015 10:30:23 GMT+0100 (CET)` » devra être traduite en « `Mercredi 28 octobre 2015, 10h30` ».

4 Petits dialogues entre JavaScript et HTML

La méthode `document.write()` nous permet d’écrire dans une page HTML, mais cette technique n’est pas très raffinée.

JavaScript nous permet d’accéder à tous les éléments du document² et d’ajouter, changer ou enlever ces éléments, leur attributs et leur styles CSS.

La méthode la plus facile pour accéder à un élément du document est à travers son identifiant. Pour ce faire, on utilisera le code `document.getElementById(id_de_l_element)`.

Par exemple, si notre page HTML contient le code

```
<h1 id="mon_titre">Mon ancien titre</h1>
```

on peut trouver cet élément en utilisant, dans le fichier `.js`, le code :

```
x = document.getElementById("mon_titre");
```

1. Pour le code HTML et CSS on a préféré dans les TDs précédents la notation avec le tiret bas : `mon_film_prefere`, `mon_element_arrondi`, etc.

2. Formalement du **DOM**, ou **M**odèle **O**bjets du **D**ocument.

4.1 Différents façons d’obtenir des éléments (avec un interlude sur les tableaux)

Comme l’identifiant d’un élément est unique, la méthode `document.getElementById()` donne un seul résultat.

En JavaScript on peut aussi chercher *tous* les éléments ayant la même balise, avec la méthode `document.getElementsByTagName()`, ainsi que *tous* les éléments ayant une même classe, avec la méthode `document.getElementsByClassName()`.

Par exemple, le script

```
y = document.getElementsByTagName("h1");
```

nous permet de trouver tous les titres de niveau 1 du document.

Comme en général il y a plusieurs éléments ayant la même balise, les résultats de notre recherche seront stockés dans un *tableau*. Un tableau est une variable particulière³ qui contient plusieurs valeurs, indexées à partir de 0.

Par exemple, dans le code ci-dessus, `y[0]` contiendra le premier titre de niveau 1, `y[1]` le deuxième, etc. Le dernier titre de niveau 1 sera contenu dans `y[y.length - 1]` (comme vous en douterez, la méthode `length` permet de récupérer la longueur du tableau).

4.2 Changer le contenu

Une fois l’élément trouvé, on peut changer son contenu HTML (*interne*, c-à-d sans les balises délimitant l’élément) en utilisant la méthode `innerHTML`.

Par exemple, si on ajoute au script vu au début de la section la ligne

```
x.innerHTML = "Mon nouveau titre";
```

le texte du titre de niveau 1 d’identifiant « `mon_titre` » dans la page HTML sera changé en “Mon nouveau titre”.

► **Exercice 5 :** Reprenez la page créée dans le TD précédent et, à l’aide d’un script JavaScript, changez le titre de la bannière en “Mon premier cours de Programmation Web”.

4.3 Changer le valeur d’un attribut HTML

Pour changer le valeur d’un attribut on utilise la syntaxe `x.attribut = "valeur"`. Supposons, par exemple, le morceau de code HTML suivant :

```

```

et le code JavaScript :

```
document.getElementById("dessin").src = "une_autre_image.png";
```

Comme résultat dans notre page web, l’image « `une_image.png` » sera remplacée par l’image « `une_autre_image.png` ».

► **Exercice 6 :** Implantez l’exemple précédent en utilisant les images « `soleil.png` » et « `lune.png` ».

3. En fait un *objet*, mais on essaiera de ne pas utiliser de gros mots aujourd’hui.

4.4 Changer le style CSS

D'une façon similaire à ce qui a été fait jusqu'à présent, on peut changer le style CSS d'un élément HTML en utilisant la syntaxe `x.style.propriété = "valeur"`;

Par exemple, le code :

```
d = document.getElementById("mon_div");
d.style.backgroundImage = "url('nouveau_paysage.png')";
```

change l'image d'arrière plan de l'élément `mon_div`. Remarquez qu'on transforme en lowerCamelCase toutes les propriétés CSS (on écrit `backgroundImage` à la place de `background-image`).

4.5 Mangez-moi, Buvez-moi, Cliquez-moi

Jusqu'à maintenant on a changé les éléments HTML (ou leur style) indépendamment des actions de l'utilisateur. En effet, celui-ci en général ne voit pas que le rendu final.

En JavaScript on peut décider d'exécuter un script quand un certain événement (causé, par exemple, par l'utilisateur) se produit.

Le code suivant permet de créer un bouton et de changer l'attribut d'un élément (dans notre cas la source de l'image d'identifiant « image ») quand on clique dessus :

```
<button onclick="getElementById('image').src =
'nouvelle_image.png'">Changez l'image !</button>
```

► **Exercice 7 :** En utilisant deux boutons (ou si vous préférez les images « `eat_me.png` » et « `drink_me.png` ») créez un script qui permet de changer la taille (horizontale et verticale) de l'image « `alice.png` », agrandissant l'image si on clique sur le premier et la réduisant si on clique sur le second.

4.6 Ce que le jour doit à la nuit

Et si on pouvait changer notre taille tout seul, sans avoir besoin de gâteaux ou de boissons magiques? En utilisant la syntaxe `this.méthode` on peut activer un script après un événement qui arrive au même élément.

Par exemple le code :

```
<button onclick="this.innerHTML = 'Fait !'">
Changez le texte</button>
```

permet de changer le texte sur le bouton en cliquant... sur le bouton même.

► **Exercice 8 :** Modifiez l'Exercice 6 pour changer l'image quand on clique dessus.

Et si on voulait passer du jour à la nuit un nombre indéfini de fois? (pensez à utiliser des structures conditionnelles)

4.7 De l'autre côté du miroir

D'autres événements qu'on peut gérer sont le passage et le relâchement de la souris. Pour cela, on utilisera les méthodes `onmouseover` et `onmouseout`. La syntaxe est la même que dans les sections précédentes.

► **Exercice 9** : Modifiez l'Exercice 7 pour tourner horizontalement l'image d'Alice quand on passe la souris dessus ou la tourner verticalement quand on conserve le clic (pensez à réviser les transformations 2D et 3D vues dans le TD précédent⁴).

5 Des photos *vraiment* en désordre

Pour effectuer des opérations mathématiques, il peut être utile d'utiliser l'objet⁵ `Math`. Avec les méthodes `max()` et `min()` on peut récupérer respectivement la valeur la plus élevée et la valeur la plus basse dans une liste d'arguments.

Par exemple, dans le script suivant :

```
x = Math.min(0, 1, 2, 3, 5, 8, -5, 13);
```

la valeur de la variable `x` sera `-5`.

La méthode `round()` arrondit un nombre à l'entier le plus proche.

Testez le script suivant :

```
document.write(Math.round(2.4) + " plus " + Math.round(2.4) +  
" égale " + (Math.round(2.4 + 2.4)));
```

Une méthode très utile, pas seulement en Mathématiques mais aussi dans le *monde réel*, est `Math.random()`. Cela renvoie un nombre aléatoire⁶ entre 0 (inclus) et 1 (exclus).

► **Exercice 10** : Reprenez la galerie d'image créée dans les TDs précédents. Utilisez la méthode `Math.random()` pour donner à chaque image un angle de rotation (vraiment) aléatoire.

4. Ou bien ici : http://www.w3schools.com/css/css3_3dtransforms.asp.

5. Oups ! À nouveau ce gros mot.

6. Enfin... l'aléatoire est un concept très intéressant : philosophiquement et informatiquement !