

# Travaux dirigés n°4

## IMAC 1

---

### *Javascript & jQuery*

*Dans ce TD, découvrons le langage javascript et nous prendrons en main le framework jQuery.*

---

## Avant-propos

Javascript est un langage de script 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.

Dans ce TD, nous verrons :

- Comment l'encapsuler dans une page HTML.
- Comment réaliser des fonctions.
- Comment modifier le code HTML en Javascript.

Pour réaliser ce TD, vous allez dupliquer la page index.html de votre précédent TD (intégré), et la nommer javascript.html. Les exercices suivants seront à réaliser dans cette page, que vous pourrez ensuite relier au TD3 en y faisant un lien dans le menu.

## 1. Encapsulation

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

```
<script type="text/javascript">
  [...]
</script>
```

Voici un premier exemple de code écrit en Javascript :

```
<html>
<head>
  <title>Voici une page contenant du Javascript</title>
</head>
<body>
  <script type="text/javascript">
    alert("Voici un message d'alerte!");
  </script>
</body>
</html>
```

Dans cet exemple on appelle la méthode alert dans le corps du programme. Modifions l'exemple, pour qu'elle soit appelée dans une fonction :

```
<html>
<head>
  <title> Voici une page contenant du Javascript</title>
  <script type="text/javascript">

  fonction myalert() {
    alert("OK, ca marche")
    alert("un deuxieme pour la route")
  }
</script>
</head>
<body>
  <script type="text/javascript">
    myalert();
  </script>
</body>

</html>
```

On peut donc constater que du code Javascript peut s'insérer dans l'entête autant que dans le corps de la page. Enfin, comme dans la plupart des langages de programmation, nous pouvons utiliser des variables, et passer des arguments aux fonctions. En exemple, notre code encore une fois modifié.

```
<html>
<head>
  <title> Voici une page contenant du Javascript</title>

  <script type="text/javascript">
  fonction myalert(text) {
    alert(text)
    alert("OK...")
  }
  </script>
</head>
<body>
  <script type="text/javascript">
    var mytext = "c'est mon texte";
    myalert(mytext);
  </script>

</body>
</html>
```

Comme en CSS, vous pouvez également (et devez, généralement) externaliser le code Javascript, qui sera contenu dans un ou plusieurs fichiers (\*.js). Cela se fait de la façon suivante :

```
<html>
<head>
  [...]
  <script type="text/javascript" src="js/myjs.js"></script>
</head>
[...]
```

Et simplement, dans le fichier .js, vous pourrez utiliser la même syntaxe qu'au sein de la page :

```
alert('hello world') ;
```

Il faut retenir :

- que l'on déclare une variable grâce au mot clé var (pas obligatoire mais fortement recommandé).
- que l'affectation, ou initialisation se fait avec l'opérateur = (à ne pas confondre avec l'opérateur de comparaison ==).
- que Javascript comme le C ou Java, différencie les minuscules des majuscules.
- que Javascript utilise un typage faible pour ses variables. De plus il détermine automatique en fonction de l'initialisation le type de ses variables : chaîne de caractères, nombre ou indéterminé ("undefined").

## 2. Du déjà vu

La syntaxe de Javascript ressemble à s'y méprendre à celle du langage C. Ainsi tout ce qui est test et boucle est identique à ce langage.

En vous aidant de vos connaissances en C, modifiez votre page pour que celle-ci affiche 10 boîtes d'alerte pour les 10 premiers entiers (1, 2, ...) dans lequel on affiche si l'entier est pair ou impair.

## 3. jQuery

jQuery est une librairie Javascript (ou framework : boîte à outils) qui permet de simplifier la syntaxe et les commandes du langage. jQuery possède donc sa propre syntaxe. Une fois prise en main, cette dernière est puissante et s'avère être un gain de temps de développement énorme.

Tout d'abord, vous devez récupérer le fichier javascript de la librairie sur [le site de jQuery](#). Vous le placerez dans un répertoire « /js/ » et vous le joindrez à votre page de la façon suivante :

```
<script type="text/javascript" src="js/[nom du fichier .js]"></script>
```

Vous devrez ensuite créer un autre fichier Javascript où sera stocké votre code jQuery (script.js par exemple), et l'ajouter à votre page de la même façon. Celui-ci devra contenir les instructions suivantes, qui engloberont tout le reste du code :

```
$(document).ready(function() {
  ... // Le code jQuery à exécuter une fois le document chargé
});
```

jQuery permet de sélectionner ou de créer des entités CSS ou HTML d'une page « à la volée », puis de les modifier dynamiquement. Cette sélection se fait grâce à la syntaxe suivante :

```
$("#left") // sélectionne l'entité HTML dont l'id est « left »  
$(".item") // sélectionne les entités HTML dont la class est « item »  
$(".item p") // sélectionne tous les paragraphes contenus dans un « item »
```

Vous constaterez que ce mécanisme de sélection est similaire à la syntaxe du CSS. Beaucoup d'autres modes sont possibles, vous les trouverez ici : <http://docs.jquery.com/Selectors>

Une fois un – ou plusieurs – éléments sélectionnés, vous pouvez les modifier dynamiquement. Pour cela, nous allons utiliser la méthode « `addClass()` » qui permet d'ajouter une classe CSS à un élément.

```
$(".item").addClass("item2");
```

Sélectionnez tous les titres de la partie centrale de votre page (à côté des pictogrammes), et appliquez-leur une classe CSS (que vous devrez sûrement créer) afin de les changer de couleur.

## 4. Carrousel

En jQuery, vous allez maintenant modifier l'image de votre carrousel. D'autres images sont disponibles dans l'archive du TD.

- Grâce à la méthode « [attr\(\)](#) », modifiez la source de l'image par une image à votre disposition.
- Automatisez cette action toutes les 5 secondes grâce à un [timer javascript](#).

## 5. Aller plus loin

- Créez un effet de [fadeIn](#) / [fadeOut](#) lors de la transition d'images.