

# Travaux dirigés n°5

## IMAC 1

---

### *jQuery & AJAX*

*Dans ce TD, nous irons plus loin dans l'utilisation du framework Javascript jQuery. Nous verrons le concept de base d'AJAX.*

---

## Avant-propos

La semaine dernière, nous avons découvert le framework jQuery en utilisant les sélecteurs et les effets. Cette semaine, nous verrons :

- Les écouteurs d'évènements.
- Les premières requêtes AJAX.

Pour réaliser ce TD, vous allez dupliquer la page javascript.html de votre précédent TD, et la nommer jquery.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. Les évènements

En Javascript, un évènement et une action lancée par l'utilisateur (ou non) : une souris qui bouge, un survol, un clic... mais aussi le chargement d'un élément par exemple.

jQuery est également capable de tracker ces évènements grâce à des écouteurs. Un écouteur, c'est simplement une fonction qui est appelée automatiquement une fois qu'un évènement donné à eu lieu. C'est le cas du `$(document).ready(...)` que vous avez déjà utilisé, qui peut se traduire par : « Une fois que le code HTML du document a été chargé, on lance les instructions suivantes... ».

Nous allons commencer par écouter le clic sur certains éléments. Créez quatre liens (balises `<a>`) n'importe où sur votre page. Chaque lien devra modifier l'image affichée dans le carrousel. Vous pouvez englober ces liens au sein d'une liste qui portera un ID équivoque (*links* par exemple).

En vous servant du TD de la semaine dernière, sélectionnez ces liens grâce à une syntaxe jQuery. Ce sélecteur appellera une fonction, comme dans l'exemple ci-dessous.

```
$("....").click(function() {  
  // Bloc d'instructions  
});
```

Vous venez de créer votre premier écouteur. Dans ce bloc, vous pourrez accéder aux attributs du lien cliqué grâce au mot-clé `$(this)` et, par exemple, à la méthode `attr()` :

```
$("....").click(function() {  
  alert($(this).attr("href"));  
});
```

- Trouvez un moyen d'identifier le lien cliqué, et de modifier l'image du carrousel (comme la semaine dernière) en fonction.
- Adaptez le timer de la semaine dernière. Celui-ci devra être arrêté lors du clic sur l'image affichée et devra reprendre après un double-clic.

## 2. AJAX

AJAX (acronyme de Asynchronous Javascript and XML), contrairement aux idées reçues, n'est pas un langage. C'est simplement une façon de concevoir des applications web.

AJAX permet, en Javascript, d'appeler de façon asynchrone (c'est-à-dire indépendamment du chargement de la page) le serveur, qui renverra des informations qui pourront être traitées via Javascript. A la base, il devait s'agir de nœuds XML, mais on peut en fait récupérer n'importe quel type de données : code HTML, objets JSON, simple texte... Une fois les données récupérées, on peut les injecter dans une partie de la page web.

Ainsi, on peut envisager modifier le contenu d'une page HTML lors d'une interaction de l'utilisateur (lors d'un clic sur un bouton par exemple) sans avoir à la recharger entièrement. jQuery, en bon framework Javascript, permet d'implémenter facilement les requêtes AJAX. Ce qui prendrait une dizaine de lignes en Javascript « de base » est simplifié au maximum. Pour en savoir plus, vous pouvez consulter la [documentation](#). Voici un exemple :

```
$.ajax({
  type: "GET",
  url: "some.txt",
  success: function(msg){
    alert( "Contenu du fichier : " + msg );
  }
});
```

Pour tester AJAX, créez un dossier **ajax/** à la racine de votre projet. Placez-y un fichier texte (.txt) contenant un texte basique. Puis, ajoutez un lien (<a>) dans votre code HTML, en lui attribuant un ID (*link-ajax* par exemple). Vous devrez également créer une <div> vide, elle-aussi identifiée par un ID (*success-ajax* par exemple).

L'idée est de remplir cette <div> avec le contenu du fichier texte lors du clic de l'utilisateur sur le lien.

- Créer un écouteur sur votre lien *link-ajax* comme vu dans l'exercice précédent.
- Adaptez l'exemple ci-dessus pour récupérer les données du fichier texte.
- Injectez le contenu du fichier dans votre div *success-ajax* grâce à la méthode jQuery [html\(\)](#).

Bravo, vous venez de réaliser votre première requête AJAX ! Continuez sur votre lancée :

- Créez trois liens dans votre page HTML. Un clic sur chacun de ces liens appellera via AJAX un fichier différent, mais remplira la même div. Comme dans l'exercice précédent, vous devrez identifier le lien cliqué, et appeler le fichier en conséquence.

### 3. Aller plus loin

- Injectez du code HTML dans votre document (à la place du texte simple).
- Plutôt que de faire trois fichiers, créez un seul fichier PHP. Vous devrez ajouter un paramètre (POST) à votre requête AJAX. En fonction de ce paramètre le PHP renverra le bon contenu.
- Modifiez votre site du WEI (TD2) pour que tous les liens soient appellent en fait des requêtes AJAX, remplissant la partie contenu de votre page.