

# Programmation Web « Client Riche »

## M413 - TD n°2 (séance 3)

### 1 Objectifs

Ce TD illustre la partie du cours sur la programmation web et en particulier les concepts liés à la conception d'application riches coté client, c'est-à-dire dans votre navigateur.

Les principaux concepts abordés seront :

- Les principes avancés du langage JavaScript.
- L'exploitation du DOM.

Vos « **livrables** » qui seront à déposer dans **la boîte de dépôt ad-hoc sur le LMS Moodle d'UCA avant la date spécifiée**, seront basés sur l'arborescence fournie et constitués **a minima** des **dossiers *assets/*, *css/*, *js/*** et des **fichiers** suivants :

- Les réponses à chacune des questions posées devront être rédigées au format « **Markdown** » dans le fichier « **README.md** » fournie.
- Des fichiers *index.html*, *js/td{x}.js* et *css/td{x}.css*, et plus si nécessaire...

<https://fr.wikipedia.org/wiki/Markdown>

### 2 Documentation

Pour ce TD, mais également pour les suivants, sauf mention contraire dans l'énoncé ou si un lien vous est fourni, vous ne devez pas aller chercher des réponses sur internet !

**Seuls les sites suivants sont autorisés :**

- Le cours sur : <https://lms.univ-cotedazur.fr/>
- Le site de Mozilla : <https://developer.mozilla.org/fr/>
- Les sites du W3C : pour la validation [HTML5](#) ou du CSS  
Mais aussi pour la documentation sur le [DOM](#).

## 3 Prise en main de l'environnement

Pour réaliser ce TD, vous aurez également besoin d'un éditeur de texte pour écrire le code de vos pages. Vous pouvez par exemple utiliser un des logiciels **Brackets**, **Notepad++**, **Eclipse**, etc.

Pensez à sauvegarder régulièrement votre travail et à commenter votre code.

N'hésitez pas à sauvegarder les différentes versions, évolutions d'un même exercice de manière à pouvoir facilement réviser ou le réutiliser par la suite...

## 4 Les Outils pour parcourir le Document Object Model

De nos jours, des outils de débogage évolués sont intégrés aux différents navigateurs ( cf. cours). Ces outils vous seront très utiles pour explorer l'arbre DOM de votre page, vérifier les propriétés CSS ou encore connaître les erreurs de votre code JavaScript !

## 5 Rappel de cours

Lors de ce TD, l'ensemble des pages web écrites devra (sauf mention contraire) être constitué de pages XHTML5 valide ne contenant pas de mise en forme ( squelette index.html fournis).

La mise en forme sera dans un ou plusieurs fichiers CSS valides ( squelette css/td{x}.css fournis).

De même, le code JavaScript sera également dans des fichiers séparés ( squelette js/td{x}.js fournis).

## 6 Exercice 1 : Les évènements (obligatoire)

Pour cet exercice vous devez partir d'une page XHTML5 valide qui contiendra au moins :

- une balise **<h1>**,
- deux balises **<h2>**,
- quelques balises **<div>**, **<p>**, **<ul>** ou **<ol>**, **<li>**, **<a>** et **<span>**.

Un objet devra avoir une couleur d'arrière-plan spécifiée par son **id**, d'autres par des **classes CSS** et d'autres encore par leur **type de balises**.

Au besoin, vous pouvez utiliser « [Lorem Ipsum](#) » pour générer de faux contenu pour votre page.

### 6.1 Sélection d'un Objet

L'objectif final de notre exemple sera de pouvoir sélectionner un objet (i.e. `HTMLElement`) dans notre page via un simple clic souris.

La sélection de l'objet sera indiquée à l'utilisateur par un changement de la couleur de son arrière-plan (en rouge par exemple).

Une façon simple de faire cela est d'abonner l'élément de plus haut niveau de notre page à l'évènement « click ».

Ecrivez une fonction **initSelect()** qui aura pour but de faire cela.

L'ajout d'un écouteur d'évènements « click » qui déclenchera une fonction **select()** devra se faire en JavaScript.

Ajoutez un **id= « title »** à votre **<h1>**.

➤ Comment avez-vous ajouté l'écouteur d'évènement et sur quel objet ?

Maintenant on souhaite écrire la fonction **select()** qui va mettre en couleur le fond de notre lors du premier clic, et le supprimer lors du clic suivant.

Sur le paramètre de type évènement de votre fonction de « callback », la propriété **.target** vous indique l'objet qui a reçu l'évènement [[MDN](#)].

Attention, à ne pas confondre avec l'objet qui à l'écouteur d'évènement.

➤ Que se passe-t-il si vous utilisez *currentTarget* en lieu et place de *target* ?

Notes :

- Pour tester si une variable est déjà définie, on peut comparer son type avec la chaîne de caractères « **undefined** ».
- Pour supprimer une variable de la mémoire (elle sera alors de type non définie), on peut utiliser la fonction **delete(variable)**.

## 6.2 Insertion d'objets

Ajoutez la **<div>** suivante au début du **<body>** de votre page au chargement de votre page, donc dynamiquement en JavaScript.

```
<div id="insert-div">
  <select id="insert-type" name="type">
    <option value="div">div</option>
    <option value="p">p</option>
    <option value="span">span</option>
  </select>
  <input type="text" id="insert-text" value="My New Text Element">
</div>
```

Ecrivez une fonction **select2()**, qui fonctionne comme la fonction **select()** sauf que :

- **select2()** ne permet de sélectionner d'un seul élément à la fois.  
On mettra le fond en bleu pour différencier les 2 fonctions.
- **select2()** ne permet pas de sélectionner la **<div>** ci-dessus ainsi que les éléments qu'elle contient.

Ecrivez maintenant la fonction **insertElement(target)** qui insert avant l'élément sélectionné, c'est à dire l'élément « target » de la fonction **select2()**, un élément dont le type est donné par le **<select>** de la **<div>** « insert-div » et qui contient le texte fourni par la valeur de l'**<input />** de type « text » de la **<div>** « insert-div ».

Les éléments ainsi ajoutés à votre page, pourront ensuite eux-même être sélectionnés...

- Comment avez-vous ajouté l'élément ?
- Comment avez-vous fait pour que la fonction **select2()** ignore les évènements de la **<div>** donnée ci-dessus ?

## 7 Exercice 2 : parcours de l'objet document (obligatoire)

Revenons maintenant sur le dernier exercice du TD1.

Ajoutez y les balises ci-dessous à votre programme et essayez de rechercher le texte « **ornare** ».

Si l'ensemble des occurrences n'est pas bien sélectionné ( il y en a 8 dans les balises ci-dessous), corrigez votre programme !

```
<div>
Proin ornare nunc sodales <span>ornare</span> lorem rhoncus ornare fringilla
</div>
<div>
Proin ornare nunc sodales or<span>nare</span> lorem rhoncus ornare fringilla
</div>
<div>
Proin ornare nunc sodales ornare lorem rhoncus ornare fringilla
</div>
```

## 8 Exercice 3 : parcours (optionnel)

En repartant de l'exercice 1, ajoutez les fonctionnalités suivantes :

- Quand je sélectionne un HTML`Element` via un clic, l'arrière-plan de l'objet passe en rouge, celui de son père en orange.
- Si je sélectionne un HTML`Element` O1, puis son père PO1, O1 et PO1 ont un arrière-plan rouge et l'arrière-plan du père de PO1 est en orange.  
Maintenant, si je resélectionne PO1, son arrière-plan doit revenir orange.