

# JavaScript

## Présentation

Cette introduction à JavaScript est constituée de deux tds. Il ne s'agit pas de devenir un spécialiste de JavaScript mais d'apprendre à comprendre et réutiliser du code existant : en effet, le langage JavaScript est très utilisé, que ce soit du coté client, du côté serveur, dans le développement d'applications mobiles ou encore dans la gestion d'objets connectés. Elle sera noté par un qcm d'environ 30 minutes et la note sera comptabilisée dans la ressource R109.

## Pré-requis

On enregistrera tous les fichiers utiles aux exercices qui suivent dans le dossier « www/acces/R109-JS » sur « [isis.unice.fr](http://isis.unice.fr) ». Pour transférer les documents de votre poste sur « [isis.unice.fr](http://isis.unice.fr) », qui est le serveur web du département, vous pourrez utiliser l'outil « filezilla » (hôte : <sftp://isis.unice.fr> suivi de vos login et mot de passe habituels et du port « 22 »). Une fois les fichiers déposés, si votre nom d'utilisateur est « misterx », vous pourrez les visualiser en tapant l'adresse « <http://isis.unice.fr/~misterx/acces/monfichier.html> » (si c'est le fichier « monfichier.html » que vous voulez voir).

Les exercices seront faits sous Linux, avec Firefox comme navigateur web. Dans Firefox, les outils de développement sont accessibles dans le menu en haut à droite sous le titre « Outils supplémentaires » puis « Outils de développement web ».

C'est en choisissant ensuite l'onglet « Console » que l'on verra apparaître en bas d'écran une zone d'affichage qui donnera entre autre toutes les erreurs JavaScript. En cliquant sur l'onglet « Inspecteur », on pourra parcourir tous les éléments HTML de la page. Enfin en cliquant sur « Éditeur de Style », on aura accès à tout le css utilisé dans la page, en lecture mais aussi en écriture.

Les réponses aux différentes questions seront données dans un fichier texte, sur papier ou dans les divers fichiers de code demandés.

## Introduction

Lors de l'apparition du web dans les années 1990, seul le langage HTML, très petit sous-ensemble de celui que nous connaissons actuellement, était nécessaire pour construire un site web.

Aujourd'hui :

- à un HTML dans sa version 5.0 utilisé pour décrire la structure du document
- s'ajoute le CSS (Cascading Style Sheet) pour l'apparence
- et JavaScript pour la modification dynamique des structures et apparences et pour l'interactivité. JavaScript est également utilisé du coté serveur à travers « node.js ».

L'objectif de ces tds n'est pas de vous transformer en des professionnels du JavaScript, un module entier serait au moins nécessaire, mais de vous permettre de comprendre suffisamment le JavaScript

## Introduction à JavaScript

pour réutiliser des portions de code dans vos pages et d'acquérir les bases nécessaires à l'utilisation d'une bibliothèque très connue, JQuery.

Pour cela, on verra dans une première partie comment introduire du code JavaScript dans une page HTML, avant d'étudier dans une seconde partie le DOM (Document Object Model) d'une telle page et d'apprendre dans une troisième partie comment Jquery permet la gestion des événements (chargement, ouverture, clic, survol...). Une quatrième partie montrera comment modifier DOM et CSS. avec JQuery.

## JavaScript et HTML

Le code JavaScript utilisé dans une page HTML doit se placer entre les balises <script> et </script>.

Il est possible :

- soit d'avoir le code directement entre les balises <script> et </script> ;
- soit d'avoir le code dans un fichier externe :
  - <script src="nomDuFichier.js"></script>

L'adresse de la source peut être :

- l'adresse d'un fichier local ;
- l'adresse d'un fichier distant.

### Question

Quelle est l'extension que l'on doit donner à un fichier JavaScript ?

### Question

Selon vous, quels sont les avantages et les inconvénients du code directement intégré à la page web et du code placé dans un fichier externe ?

### Question

Selon vous, vaut-il mieux placer le code JavaScript en début ou en fin de page web ?

### Question

Faites référence au code JavaScript situé à l'adresse

<http://isis.unice.fr/~mgautero/ext/R109-JS/intro.js>

dans le code html situé à l'adresse

<http://isis.unice.fr/~mgautero/ext/R109-JS/debut.html>

que vous aurez préalablement copié chez vous. Vérifiez que votre solution fonctionne. Que s'affiche-t-il dans la page web récupérée ?

Pour comprendre ce qui se passe, il est nécessaire de savoir comment les pages web sont représentées dans le navigateur, de découvrir la structure du DOM ou « Document Object Model »

# Le DOM (« Document Object Model »)

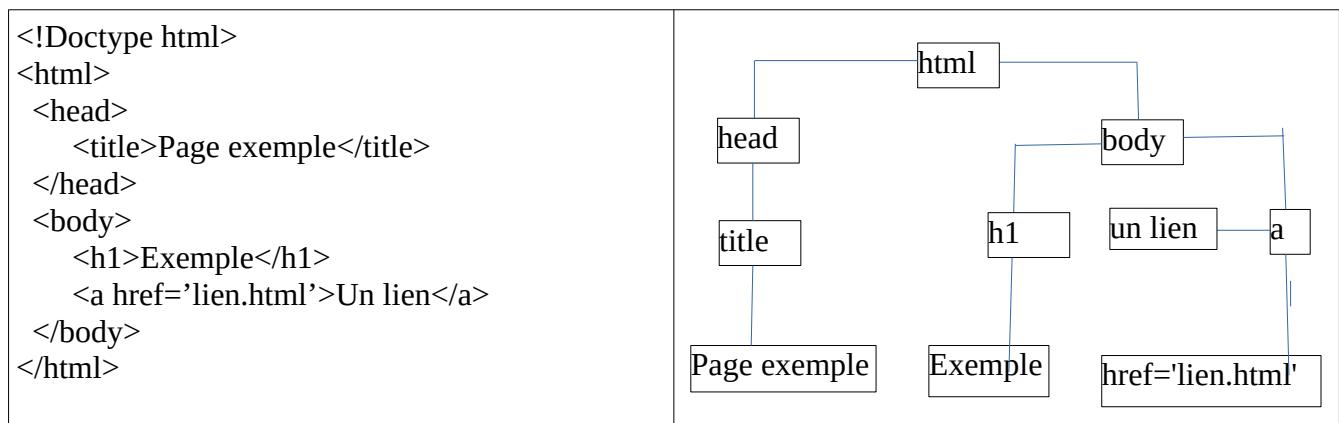
## Explications

Chaque page HTML est représentée par le navigateur web sous forme d'un arbre dont :

- La racine est le document.
- Chaque balise ouvrante est une nœud qui peut posséder des fils, que ce soit des balises ou des feuilles.
- Chaque texte et chaque attribut sont des feuilles.

Dans l'exemple ci-dessous, on voit à gauche la page HTML et à droite le DOM correspondant :

- La balise « html » a deux fils, « head » et « body ».
- La balise « head » a un fils, « title » qui a une feuille, son texte.
- La balise « body » a deux fils, « h1 » et « a »
- ...
- Les balises « h1 » et « a » sont frères.



## Exercices

JavaScript permet d'accéder au DOM par l'intermédiaire d'une API (« Application Programming Interface »). Celle-ci étant de bas niveau, nous utiliserons en priorité la bibliothèque JQuery qui offre une interface de plus haut niveau. Dans le code de la page « intro.html », on peut voir l'appel à la bibliothèque Jquery, indispensable avant tout appel à une fonction Jquery.

## Jquery

Si l'on examine le fichier JavaScript « intro.js » utilisé précédemment, on peut voir le code suivant :

```
$(function() {  
    // Code à exécuter  
});  
  
qui peut aussi s'écrire  
  
jQuery(document).ready(function() {  
    // Code à exécuter  
});
```

Le code qui se trouve dans la fonction sans nom et sans paramètre (« function () ») sera automatiquement exécuté une fois le document, le DOM, entièrement construit, quand le document sera « ready ».

Analysons ce code : « \$('#tblabla').append(' : texte inséré par Jquery.');" :

- La première partie, « \$('#tblabla') » agit comme un sélecteur : on recherche dans le DOM l'élément dont l'id (caractère « # ») est blabla.
- La seconde partie « append(' : texte inséré par Jquery.') » indique l'action que l'on veut faire sur l'élément sélectionné. Ici il s'agit d'ajouter au contenu de l'élément dont l'id est « blabla » le texte passé en paramètre, « : texte inséré par Jquery. »

### Question

Modifier le fichier JavaScript, préalablement recopié dans votre dossier, de telle sorte que le texte du second h1 soit également modifié lors du chargement de la page.

## Sélection d'élément avec Jquery

On a vu que le code « \$('#nomId') » permettait de sélectionner l'élément du DOM courant dont l'id est « nomId ».

De la même manière, il est possible avec Jquery de sélectionner :

- Les éléments appartenant à une classe donnée par le sélecteur « \$('.nomClasse') »
- Les éléments correspondants à une balise donnée par le sélecteur « \$('balise') ». Par exemple « \$('ul') » sélectionnera toutes les balises « ul » du document.

Comme en CSS, il est possible de combiner tout ceci :

- « \$('div p') » sélectionnera les balises « p » enfants d'une balise « div ».
- « \$('div.desc') » sélectionnera les balises « div » de classe « desc ».
- ...

Comme en CSS, il existe des sélecteurs particuliers, qui peuvent se combiner avec les autres :

- « \$('selecteur :last') » renvoie le dernier élément d'une liste d'éléments sélectionnés
- « \$('selecteur :first') » renvoie le premier élément d'une liste d'éléments sélectionnés
- « \$('selecteur :eq(n)') » renvoie le nième élément d'une liste d'éléments sélectionnés
- ...

### Question

Sachant que l'instruction `selecteur.css(« nomPropriétéCss », valeurPropriété)` appliquée à un ou plusieurs élément du DOM permet de donner à la propriété CSS de nom « `nomPropriétéCss` » la valeur « `valeurPropriété` », proposer le sélecteur à donner pour :

- Mettre à rouge le fond de page
- Écrire « `h1` » en bleu
- Avoir un fond vert pour les `li` des listes non ordonnées (`ul`)
- Mettre en gras tous les éléments de la classe « `important` »
- Mettre les éléments des listes contenus dans des tableaux en jaune.
- Mettre en italique le texte du premier et du dernier élément d'une liste d'identifiant « `liste` »

## Gestion des événements avec Jquery

Tout ceci est bien joli mais quel intérêt de modifier quoi que ce soit au moment du chargement de la page ? Pas très grand en effet, du moins tant que les modifications restent aussi simplistes. Par contre, il serait plus intéressant de pouvoir faire une modification quelconque :

- quand un élément perd (ou gagne) le rôle d'élément actif (on quitte ou on pénètre le champ d'un formulaire par exemple)
- quand un élément change de valeur (liste)
- quand un clic se produit, que ce soit avec le bouton gauche ou le bouton droit
- quand la souris survole un élément
- quand une touche est pressée
- quand un élément est chargé
- ...

bref quand le moindre événement se produit sur la page web.

Jquery possède des fonctions qui vont permettre d'intercepter chacun de ces événements : « `blur`, `click`, `dblclick...` » (voir la liste complète à l'adresse <http://api.jquery.com/category/events/> ).

Par exemple si l'on souhaite que la couleur de fond de toutes les balises « `h1` » passent au rouge quand on clique sur l'élément d'identifiant « `cliqueMoi` », on écrira :

```
$("#cliqueMoi").click(function () {  
    $("h1").css("background-color", "red");  
});
```

### Question

Écrire une page « testeEvent.html » qui vous permettra de tester les événements « click », « mouseover », « mouseout ».

Indications : l'instruction « window.open("URL") ; » où URL représente une URL ouvre une nouvelle fenêtre montrant cette URL.

L'instruction « window.alert("TEXTE") ; » ouvre une fenêtre d'alerte montrant le message TEXTE.

On pourra utiliser ces deux instructions pour voir facilement le bon fonctionnement de la gestion des différents événements.

On a parfois besoin de savoir quel est l'objet, l'élément du DOM, qui a généré l'événement. En Jquery, quand on est dans la fonction qui gère un événement, l'élément courant, celui qui a généré l'événement est accessible par « \$(this) ». Par exemple avec le code

```
« $('h1').click(function () {  
    $(this).css("background-color", "red");  
});
```

un click sur un élément h1 mettra le fond de cet h1 et de cet h1 uniquement en rouge.

### Question

Créer une page "tableau.html" qui contiendra au minimum un tableau de plusieurs lignes (4 ou 5 minimum) et y ajouter le code Jquery qui permettra de mettre en bleu la couleur des lignes quand le curseur de la souris les survole et revenir à la couleur de base quand le curseur les quitte.