
Initiation au JavaScript

I. Sommaire

II. JavaScript ?	1
III. Du JavaScript dans une page web	2
III.1. Cas du bloc entouré de balises.....	2
III.2. Cas du fichier externe.....	2
III.3. Exercice 11.....	2
IV. jQuery : un framework JavaScript	3
IV.1. jQuery ?.....	3
IV.2. Ajouter jQuery à ses pages.....	3
V. Premiers pas en jQuery	4
V.1. Un exemple.....	4
V.2. Le sélecteur de jQuery : \$().....	4
V.3. Les événements jQuery.....	4
V.4. Modifier le contenu d'une page en jQuery.....	5
V.5. Exercice 12.....	6
V.6. Collecter des informations sur le DOM avec jQuery.....	6
V.6.1. Exercice 13.....	7
V.7. Naviguer dans le DOM avec jQuery.....	7
V.7.1. Exercice 14.....	8
VI. Programmer en JavaScript	8
VI.1. Le langage JavaScript.....	8
VI.2. Exercice 15.....	9
VII. Des plugins et scripts jQuery	9
VII.1. Exercice 16.....	10

II. JavaScript ?

JavaScript est un langage de programmation de haut niveau, de type « script ». Les langages de script sont interprétés directement sans être transformés (par exemple par une compilation) et nécessitent donc un interpréteur (programme capable de les transformer à la volée en instructions-machine).

Il y a souvent confusion entre JavaScript et Java. Java n'est pas un langage de script et ne peut donc être interprété directement, il demande une phase de compilation avant d'être exécuté. JavaScript s'inspirait de Java à sa création, mais s'est différencié dès le départ notamment par le fait qu'en tant que langage de script il est interprété directement. Les navigateurs web (Mozilla Firefox, Google Chrome, Microsoft Internet Explorer, etc.) possèdent tous un interpréteur JavaScript. Ils sont ainsi capables d'exécuter du JavaScript.

Il faut donc prendre conscience que sur le web, les programmes en JavaScript sont exécutés par le client web (le navigateur, chez l'utilisateur) et non comme les programmes en PHP par le serveur web (ex. : Apache).

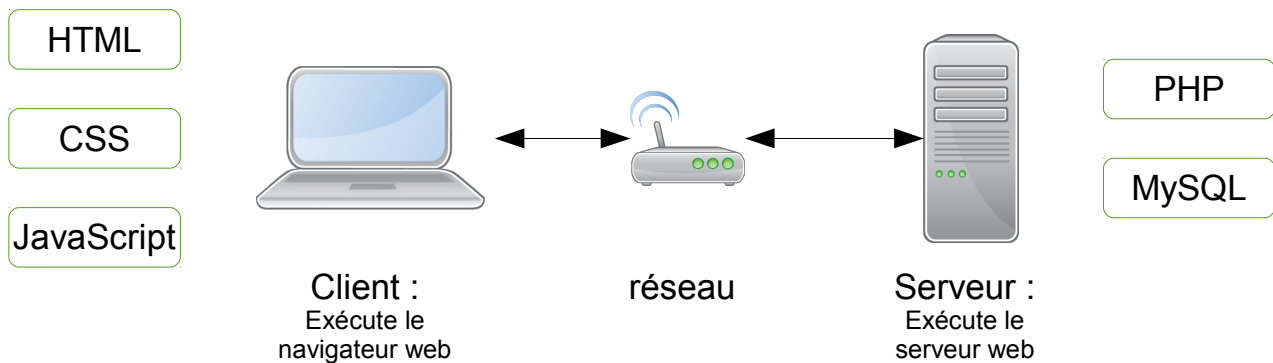


Figure 1 : Quelques langages interprétés par le client et le serveur web

voir aussi : https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide/JavaScript_Overview

III. Du JavaScript dans une page web

Comme pour les CSS, il y a deux façons d'intégrer du JavaScript dans une page web (HTML). Soit directement dans la page web en entourant le bloc contenant JavaScript par des balises `<script type="text/javascript"></script>`, soit en désignant un fichier externe.

III.1. Cas du bloc entouré de balises

```
<script type="text/javascript">
  alert("ceci est un programme en JavaScript");
</script>
```

Ce script ne contient qu'une seule instruction qui ouvre une boîte de dialogue d'information. Essayez-le dans jsfiddle (<http://jsfiddle.net/>).

III.2. Cas du fichier externe

```
<script type='text/javascript' src="/chemin_relatif_vers_fichier/nomfichier.js"></script>
```

On remarque dans ce cas que la balise `<script>` n'a pas de contenu. Le contenu est chargé à partir du fichier externe. Ces fichiers sont des fichiers textes contenant simplement du JavaScript. Par ex. :

```
// Exemple de contenu pour le fichier nomfichier.js
alert("ceci est un programme en JavaScript");
```

III.3. Exercice 11

Bootstrap utilise du JavaScript, d'une part pour rendre les anciennes versions d'Internet Explorer compatibles HTML5 et CSS 3, d'autre part pour certaines fonctionnalités (comme le menu « responsive »).

À partir de base de Bootstrap, repérez les programmes JavaScript qui sont chargés :

```

<!DOCTYPE html>
<html>
  <head>
    <title>Base de bootstrap</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
      <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Titre du contenu</h1>
    ... contenu...
    <script src="jquery/jquery-2.1.1.min.js"></script>
    <script src="bootstrap/js/bootstrap.min.js"></script>
  </body>
</html>

```

IV. jQuery : un framework JavaScript

IV.1. jQuery ?

Comme pour le HTML/CSS, il existe des frameworks JavaScript. Le but est toujours le même :

Un framework est un cadre pour organiser et faciliter le développement grâce à des concepts clairs, des méthodes efficaces, des morceaux de code, et des éléments graphiques préconçus qui fonctionnent partout.

[jQuery](#) est un framework JavaScript Libre et vraiment **très** répandu sur le web. Il facilite l'utilisation du JavaScript et permet très facilement d'améliorer ses pages en les rendant plus interactives.

Autres exemples de frameworks web : [Prototype](#), [MooTools](#), etc.

N'hésitez pas à consulter la documentation de jQuery : <https://learn.jquery.com/using-jquery-core/>

IV.2. Ajouter jQuery à ses pages

Bootstrap utilise jQuery pour ses fonctionnalités utilisant le JavaScript. Dans l'exercice 11, vous avez dû repérer la ligne `<script src="jquery/jquery-2.1.1.min.js"></script>` qui charge jQuery avant de charger le JavaScript spécifique de Bootstrap `<script src="bootstrap/js/bootstrap.min.js"></script>`.

Donc, pour pouvoir utiliser jQuery, il faut aller chercher le fichier sur le site de jQuery : <https://jquery.com/download/> (on remarque ici aussi qu'il y a des fichiers « *min* » minifiés et des fichiers complets pour le développement). Il faut ensuite l'inclure dans sa page HTML avant ses propres scripts grâce à l'utilisation de `<script src="chemin_vers_jquery/fichier_jquery.js"></script>`.

Si vous utilisez déjà la base de Bootstrap ces étapes ne sont pas nécessaires puisque jQuery en fait déjà partie.

V. Premiers pas en jQuery

V.1. Un exemple

Dans la page de votre CV, après la ligne qui ajoute jQuery dans votre page, ajoutez le script suivant :

```
<script type='text/javascript'>
// On indique à jQuery de réagir à l'événement "scroll"
$(document).scroll(function(){
// On indique à jQuery de cibler les éléments de classe "jumbotron"
// et d'y ajouter (append) le texte ' je défile '
$(".jumbotron").append(' je défile ');
});
</script>
```

Chargez le CV dans le navigateur et défilez la page. Vous constatez que le « *jumbotron* » se remplit de texte à chaque fois que vous défilez la page. Nous avons donc pu, avec seulement ces quelques lignes de jQuery : détecter un défilement et quand c'est le cas, modifier le contenu de la page. Analysons le contenu du script qui permet cela.

V.2. Le sélecteur de jQuery : \$()

L'élément le plus important en jQuery est \$(**sélecteur**) qui permet de sélectionner des éléments du DOM de la page pour soit :

- les surveiller afin d'y détecter un événement comme le défilement. Ex. : Le défilement (« *scroll* »)
- les modifier, par exemple en ajoutant de nouveaux contenus (« *append* »)
- les interroger (pour connaître leur contenu ou leurs propriétés)

Exemples de sélections avec jQuery :

- \$(document) sélectionne tout le document HTML (l'ensemble du DOM).
- \$(".jumbotron") sélectionne les éléments de la page qui portent la classe CSS « *jumbotron* ». On remarque que l'écriture est la même que pour les sélecteurs CSS : **.jumbotron**
- \$("#menu ul") sélectionne les listes non ordonnées (**ul**) présentes dans l'élément d'identifiant « *menu* ». Là encore l'écriture est la même que les sélecteurs CSS. Essayez dans le CV de modifier \$(".jumbotron") par \$("#menu ul").

V.3. Les événements jQuery

En JavaScript, il existe le concept d'événements. Il s'agit de **méthodes** (autrement dit de *fonctions*) qui surveillent certaines actions de l'utilisateur et du navigateur web. Ces événements permettent par exemple de déclencher des actions à certaines occasions. jQuery permet de simplifier l'utilisation des événements de JavaScript. Ainsi, dans notre exemple, l'instruction \$(document).scroll() permet de surveiller si le document défile (*scroll* en anglais).

Il existe de nombreux événements qui peuvent être surveillés. Voici quelques exemples :

- \$(**sélecteur**).click() permet de surveiller que les éléments sélectionnés sont cliqués.
- \$(**sélecteur**).hover() permet de surveiller que les éléments sélectionnés sont survolés à la souris.

- `$(sélecteur).ready()` permet de surveiller que les éléments sélectionnés sont bien chargés et insérés dans le DOM par le navigateur web (très utile pour vérifier que tout est prêt avant de modifier une page en JavaScript).
- `$(sélecteur).resize()` permet de surveiller que les éléments sélectionnés sont redimensionnés.
- `$(sélecteur).submit()` permet de surveiller si le formulaire ciblé par le sélecteur a été soumis (validé).

Vous trouverez la liste exhaustive sur le site de jQuery : <https://api.jquery.com/category/events/>

C'est de qui est passé en paramètre à la méthode qui est déclenché quand l'événement se produit. Ex. : pour provoquer quelque chose lorsqu'une page web a été entièrement chargée par le navigateur on va écrire :

```
<script type='text/javascript'>
// On indique à jQuery de réagir lorsque la page est complètement chargée par le navigateur
$(document).ready(function(){
// On place ici les instructions JavaScript à exécuter quand le DOM est prêt
});
</script>
```

On remarquera qu'en JavaScript les « // » précèdent des commentaires, c'est-à-dire des choses qui ne sont pas interprétées (comme en PHP).

Sans rentrer dans les détails, le `function(){}` permet de déclarer une nouvelle fonction (comme PHP, JavaScript permet d'écrire des fonctions), avec la particularité qu'elle n'a pas de nom et n'est donc appelée que lors de l'événement. Comme en PHP, le contenu de la fonction sera placé entre les accolades. Pour ceux qui veulent des détails sur les fonctions anonymes de JavaScript: [https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Fonctions et portée des fonctions #The function declaration .29function statement.29](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Fonctions_et_portee_des_fonctions#The_function_declaration_statement.29) et pour les plus curieux : <http://www.wewereweb.be/javascript-les-iife-ou-comprendre-les-fonction/2012/10/13/>

Exemple :

```
<script type='text/javascript'>
// On indique à jQuery de réagir lorsque la page est complètement chargée par le navigateur
$(document).ready(function(){
// On affiche une boîte d'information quand le DOM est prêt
alert('je suis prêt');
});
</script>
```

Le script ci-dessus affiche une boîte d'information lorsque le DOM a été entièrement chargé par le navigateur grâce à l'instruction : `alert('je suis prêt')`. Si votre machine n'est pas très rapide, vous remarquerez qu'il y a un délai entre le moment où le navigateur charge le page et où la boîte d'information s'ouvre. Tant que la boîte d'information ne s'ouvre pas, toute la page n'est pas chargée. C'est pourquoi on essaiera toujours de placer son code JavaScript dans une fonction anonyme déclenchée par l'événement « ready » sur « document », comme pour la boîte d'information dans l'exemple ci-dessus, afin d'attendre la que la page soit prête à être manipulée avant de lancer nos scripts.

V.4. Modifier le contenu d'une page en jQuery

Dans l'exemple initial, nous utilisons l'instruction `$(".jumbotron").append(' je défile ')` pour ajouter le texte « je défile » au « jumbotron » à chaque fois que la page défile. Cela se fait grâce à la méthode « append » qui rajoute un contenu à l'élément ciblé par le sélecteur jQuery. jQuery propose beaucoup d'autres méthodes pour modifier des contenus de la page.

Exemples de méthodes de modifications du DOM :

- `$(sélectionneur).before()` permet d'ajouter un contenu *avant* des éléments sélectionnés (alors que « append » permet de les ajouter *à la fin*).
- `$(sélectionneur).clone()` permet de cloner les éléments sélectionnés.
- `$(sélectionneur).after()` permet d'ajouter de nouveaux éléments *après* les éléments sélectionnés (alors que « append » ajoute des éléments *à la fin*, et donc à l'intérieur, de l'élément sélectionné).
- `$(sélectionneur).remove()` permet de supprimer du DOM les éléments sélectionnés.

Exemples de méthodes de modification de l'aspect de la page (souvent une action sur les CSS) :

- `$(sélectionneur).addClass()` permet d'ajouter une classe CSS aux éléments sélectionnés.
- `$(sélectionneur).hide()` permet de cacher les éléments sélectionnés (ils sont bien présents dans le HTML et dans le DOM, mais ne sont plus affichés).
- `$(sélectionneur).show()` permet de montrer les éléments sélectionnés qui sont cachés (soit grâce à « hide » soit grâce à une propriété CSS comme « display : none »).
- `$(sélectionneur).fadeIn()` permet de montrer les éléments sélectionnés qui sont cachés avec un effet de fondu progressif (il existe aussi « fadeOut » pour cacher, et « fadeToggle » pour passer d'un état à l'autre).
- `$(sélectionneur).slideDown()` permet de montrer les éléments sélectionnés qui sont cachés avec un effet de défilement progressif (il existe aussi « slideOut » et « slideToggle »).

Vous trouverez la liste exhaustive sur le site de jQuery : <https://api.jquery.com/category/manipulation/> et <https://api.jquery.com/category/effects/>

V.5. Exercice 12

Écrivez un script utilisant jQuery pour le CV permutant l'ouverture et la fermeture (par fondu ou défilement) des coordonnées au survol de « Me contacter ». Attention, les coordonnées doivent être cachées en premier lieu.

Attention : nous utilisons le survol pour cet exercice. Mais, pour vos sites, pensez aux mobiles et souvenez-vous que pour la plupart des appareils tactiles la notion de survol n'existe pas. Vous ne devez donc pas réserver des interactions à ce type d'événement (survol) et au moins les faire doubler par l'événement du clic.

V.6. Collecter des informations sur le DOM avec jQuery

jQuery permet aussi de collecter des informations sur le DOM du navigateur web. Le framework propose pour cela plusieurs méthodes. Voici quelques exemples :

- `$(sélectionneur).attr('nom_attribut')` retourne le contenu de l'attribut nommé `nom_attribut` (ex. de noms d'attributs : « href », « id », « class », etc.) du premier des éléments sélectionnés. Attention, `$(sélectionneur).attr('nom_attribut','valeur')` aura pour effet de changer la valeur de l'attribut.
- `$(sélectionneur).hasClass('nom_classe')` retourne true si les éléments sélectionnés ont la classe passée en paramètre.
- `$(sélectionneur).is(':visible')` retourne true si les éléments sélectionnés sont visibles dans la page web. Cette méthode « is » est un peu particulière : elle vérifie que les éléments sélectionnés

correspondent aussi aux sélecteurs qui lui sont passés en paramètres. **:visible** est aussi un **sélecteur particulier qui ne correspond pas à un sélecteur CSS**, qui permet de sélectionner les éléments visibles. Il peut aussi s'utiliser ainsi : `$('.section:visible')` pour sélectionner par exemple toutes les sections visibles. Il existe d'autres sélecteurs intéressants qui sont décrits dans la documentation de jQuery : <https://api.jquery.com/category/selectors/>

- `$(sélecteur).html()` retourne le contenu des éléments sélectionnés. Attention, `$(sélecteur).html('autre contenu')` aura pour effet de changer le contenu des éléments ciblés par **autre contenu**.
- `$(sélecteur).height()` retourne la hauteur d'affichage des éléments sélectionnés. Attention, `$(sélecteur).height(autre valeur)` aura pour effet de changer la hauteur des éléments ciblés par **autre valeur**. Il existe l'équivalent pour la largeur : `$(sélecteur).width()`
- `$(sélecteur).position()` retourne l'objet décrivant la position d'affichage des éléments sélectionnés. `$(sélecteur).position().top` est la méthode qui retourne la position par rapport au haut du document. `$(sélecteur).position().left` par rapport à la gauche.

Vous trouverez la liste exhaustive sur le site de jQuery : <https://api.jquery.com/category/manipulation/>

V.6.1. Exercice 13

Ajoutez au CV un script utilisant jQuery qui retourne la classe CSS d'un élément sur lequel on a cliqué dans une boîte d'information (`alert()`).

Pour faire cet exercice, vous allez devoir utiliser l'objet « *this* ». À l'intérieur d'une méthode jQuery « *this* » sert à désigner l'objet qui a été sélectionné. Ex. :

```
<script type='text/javascript'>
// quand le jumbotron est survolé
$('.jumbotron').hover(function(){
// On y ajoute ' survol ' grâce à this qui désigne le jumbotron sélectionné
$(this).append(' survol ');
});
</script>
```

V.7. Naviguer dans le DOM avec jQuery

jQuery propose des méthodes pour naviguer dans l'arbre du DOM. Elles sont rassemblées sur une page de la documentation : <https://api.jquery.com/category/traversing/>. On y trouve par exemple les méthodes suivantes :

- `$(sélecteur).children('sélecteurs des enfants (optionnels)')` permet de sélectionner les enfants des éléments sélectionnés, en les filtrant éventuellement par les sélecteurs passés en paramètre.
- `$(sélecteur).each()` joue le même rôle que le `foreach` du PHP et permet d'exécuter des fonctions pour chacun des éléments sélectionnés.
- `$(sélecteur).is('sélecteurs')` [voir ci-dessus](#).
- `$(sélecteur).parent('sélecteurs du parent (optionnels)')` permet de sélectionner le parent des éléments sélectionnés, en le filtrant éventuellement par les sélecteurs passés en paramètre.
- `$(sélecteur).siblings('sélecteurs des frères (optionnels)')` permet de sélectionner les éléments frères (c'est à dire ayant le même parent) des éléments sélectionnés, en les filtrant éventuellement par les sélecteurs passés en paramètre.

N'hésitez pas à consulter la documentation, car il y en a beaucoup d'autres qui sont intéressants.

V.7.1. Exercice 14

Remplacez le script de l'exercice 13 par un script utilisant jQuery et qui au clic sur un élément y ajoute dans une liste (.........) les classes CSS et les id de ses frères (éléments qui ont le même parent).

Pour faire cet exercice, vous pourriez avoir besoin d'utiliser des variables pour désambigüiser le « this » quand vous l'utilisez dans des méthodes imbriquées. En JavaScript, contrairement au PHP, il faut déclarer les variables avant de les utiliser. Cela se fait avec l'instruction **var**. De plus en JavaScript, les noms de variables ne commencent pas par le caractère « \$ » comme en PHP¹. Voici un exemple :

```
// On stocke la valeur de this dans la variable mon_element
// (les noms de variables ne commencent pas par $)
var mon_element = this;
// cela permet d'utiliser mon_element à la place de this
$(mon_element).hide();
```

Vous pourriez aussi avoir besoin de concaténer des chaînes de caractères. En JavaScript on ne fait pas comme en PHP (en PHP on utilise le caractère point : \$texte = "chaîne 1 ".\$variable."chaîne 2"); et l'on utilise le caractère « + » (plus). Ex. : var texte = "chaîne 1 "+variable+"chaîne 2";

En JavaScript le caractère « . », que l'on a déjà beaucoup utilisé, sert à désigner la méthode d'un objet (c'est-à-dire une fonction liée à cet objet).

VI. Programmer en JavaScript

VI.1. Le langage JavaScript

JavaScript est avant tout un langage de programmation. Nous avons vu qu'il possède des fonctions (ou méthodes), des variables. Bien sûr, il possède aussi toutes les autres caractéristiques des langages de programmation. Voici deux exemples importants :

- JavaScript a des structures de test (le couple « if else » par exemple)
- JavaScript a des boucles (par exemple : « while », « for »)

Plutôt que de rentrer dans les détails, reportez-vous à ces trois liens :

1. Une liste (non exhaustive) des différences entre PHP et JavaScript qui vous sera très utile : <https://www.lullabot.com/blog/article/learning-javascript-php-comparison>
2. Une documentation simplifiée de JavaScript proposée par les auteurs de jQuery : <https://learn.jquery.com/javascript-101/>
3. La documentation officielle de JavaScript: <https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide>

¹ D'ailleurs, le \$() du sélecteur est en fait le nom d'une variable déclarée par jQuery : la variable s'appelle \$, tout court. Cette variable \$ est un alias de la variable jQuery que l'on peut utiliser à sa place (ex. : jQuery(mon_element).hide();). Le framework JavaScript MooTools utilise une variable nommée \$\$ pour le même usage.

VI.2. Exercice 15

Revenons sur l'exercice 12 dans lequel nous faisons apparaître et disparaître le contenu de « Me contacter » lors du survol. Le but de cet exercice est de compléter le script en le rendant compatible avec les appareils tactiles (sans survol). Il faut donc prévoir le même comportement au survol et au clic :

- quand la souris est sur l'élément et qu'il est caché on affiche le contenu,
- quand la souris quitte l'élément on le cache le contenu,
- quand on clique sur l'élément, s'il était caché, il apparaît
- quand on clique sur l'élément et qu'il est visible, il disparaît
- il doit être possible de combiner survol et clic : si je survole « Me contacter », le contenu apparaît (s'il était caché), et quand je clique il disparaît.

VII. Des plugins et scripts jQuery

jQuery facilitant grandement le travail des développeurs, de nombreux outils reposent sur jQuery. Voici une liste non exhaustive de quelques scripts fondés sur jQuery (ils ne fonctionnent que s'il est présent) qui pourront vous aider :

- Pour manipuler les grilles CSS :
 - Les plugins sont nombreux, mais le plus utilisé est : <http://masonry.desandro.com/>
- Pour rendre les tableaux de données plus dynamiques pour l'utilisateur :
 - Il y a aussi beaucoup de plugins qui font ça, mais je vous conseille de commencer par essayer : <http://www.jquery-bootgrid.com/>
- Pour faire des effets visuels ou des animations :
 - jQuery UI est la référence : <http://jqueryui.com/> (d'autant qu'il permet d'enrichir les interactions)
 - Beaucoup de webdesigners considèrent que l'interaction principale d'un utilisateur de site web est le défilement. Ainsi, <http://scrollme.nckprsn.com/> permet d'associer des effets au défilement très simplement.
 - Dans la même veine : <http://alvarotrigo.com/pagePiling/>
- Afficher des carrousels de documents :
 - Bootstrap embarque un carrousel : <http://getbootstrap.com/examples/carousel/>
 - Un autre classique : <http://jquery.malsup.com/cycle/>
- Ouvrir une fenêtre modale ;
 - Il y a de très nombreux plugins pour ça. Je vous conseille de commencer par : <http://www.jacklmoore.com/colorbox/>
- Bulles d'aide (infobulles ou tooltips) :
 - <https://stevenbenner.github.io/jquery-powertip/>
 - <http://onehackoranother.com/projects/jquery/tipsy/>

- Afficher des diagrammes interactifs :
 - <http://www.chartjs.org/>
 - <http://omnipotent.net/jquery.sparkline/#s-about>
 - <http://anthonyterrien.com/knob/>
- etc.

VII.1. Exercice 16

Mettez en œuvre l'un de ces plugins/scripts jQuery dans le CV.