

Programmation Web

TP n° 2a : JavaScript et jQuery

Le but de ce TP est d'apprendre le langage JavaScript et de découvrir la bibliothèque JQuery. Les exercices proposent de l'utiliser coté client pour dynamiser des pages Web. Pour déboguer vos programmes, vous aurez besoin d'utiliser la console JavaScript. On y accède dans le navigateur par le menu des outils de développement. Elle apparaît généralement comme un volet sur la partie inférieure de la fenêtre, ou dans une fenêtre séparée. La méthode `console.log()`, appelée dans un script, permet d'afficher une chaîne de caractères dans la console. De plus la console fournit un interprète JavaScript. Testez par exemple la méthode `console.log()` directement dans l'interprète de la console.

1 JQuery

JQuery est une bibliothèque JavaScript d'usage simple, stockée dans un unique fichier `jquery.js` téléchargeable sur le Web (<http://jquery.com/download/>). La bibliothèque peut ensuite être chargée dans le *head* d'une page web :

```
<script src="chemin/vers/jquery.js" type="text/javascript"></script>
```

Elle est aussi hébergée par des *content delivery networks*, il est donc possible d'inclure JQuery dans ses pages Web depuis eux :

```
<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
```

Cette méthode délègue l'hébergement de la bibliothèque à un serveur tiers, ce qui peut avoir des bénéfices en termes de coûts d'hébergement ou de vitesse d'affichage au détriment d'une perte de contrôle : le fichier peut être modifié sans votre connaissance, et sa disponibilité n'est plus assurée par votre serveur.

Tous les fichiers (contenant des scripts) des exercices qui suivent sont à écrire sur ce modèle :

```
$(document).ready (function () {
    // corps du programme.
});
```

1. **CSS pur.** Cet exercice peut être réalisé en CSS pur. Dans une page Web, créez un bloc (div) contenant deux `span`¹. Exprimez dans une feuille de style que l'un est visible, et que l'autre ne l'est pas (display). Puis que lorsqu'on survole le bloc (:hover), son premier fils devient invisible (none) et que le second devient visible (inline). Rq vous pouvez aussi répondre à cette question en utilisant l'attribut `visibility`, qui peut être soit visible soit hidden
2. **Premiers pas avec JQuery.** Cet exercice ne peut pas être réalisé en CSS pur. Créez une page Web contenant un paragraphe « ceci est un paragraphe ». En utilisant JQuery, faites en sorte que lorsque la souris survole le paragraphe, son texte soit modifié en « ceci n'est pas un paragraphe » (`mouseenter`, `text`).
Modifier le code pour que le texte soit effectivement modifié comme précédemment lorsque la souris survole le paragraphe (sans artifice hidden ou display), et revienne à son contenu initial lorsque la souris ne le survole plus (`mouseleave`).
3. **Blocs et texte.** Méthodes utiles : `mouseover`, `mouseup`, `mousedown`, `text`, `click`, `append`, `hide` et `console.log`. Créez une page web avec un header, un footer, et un bloc avec une bordure rouge. Ce bloc sera centré, occupera 50% de la largeur de la page, et affichera en son centre le message « Cliquez ici - Décompte (0,0) ». Complétez cette page par un script externe de manière à obtenir le comportement suivant :
 - a. Lorsqu'on passe la souris sur le bloc rouge, un message « Attention, vous entrez dans la zone rouge » est affiché dans la console.

1. `span` est une balise qui n'implique pas un retour à la ligne (contrairement à `div`) ; on ne peut pas non plus lui appliquer certains positionnement (`margin ...`)

- b. Lorsqu'on presse le bouton de la souris sur le bloc rouge, la bordure du bloc devient verte et un message vert « up *xx* » apparaît dans le header html, à la suite de ce que contient déjà ce header.
Indications. La fonction `$` de jQuery permet aussi de créer de nouveaux objets jQuery en écrivant littéralement leur contenu HTML. Servez-vous de `append` pour ajouter l'élément créé à un autre élément. Le texte d'un élément textuel peut être mis à jour via la méthode `text`.
 - c. Lorsqu'on relâche le bouton de la souris sur le bloc, la bordure du bloc redevient rouge et un message rouge « down *xx* » apparaît dans le footer html, à la suite de ce que contient déjà ce footer.
 - d. Faites en sorte qu'à leur création les « *xx* » soient le numéro correspondant à l'ordre dans lequel ils ont été créés.
 - e. Lorsqu'on clique sur un mot du header ou du footer, « up » ou « down », ce mot disparaît, et le nombre de mots encore présent dans le header et dans le footer sont mis à jour dans le message du bloc central "décompte (x,y)".
Indications. Les éléments créés à la question **b** peuvent être munis d'un écouteur. Plutôt que de les supprimer, rendez-les invisibles via la méthode `hide`.
 - f. (Question supplémentaire) À la question **c**, si la souris n'est pas relâchée sur le bloc, aucun message ne sera ajouté au footer. Comment faire pour que cet ajout ait lieu si la souris est relâchée n'importe où sur le page, et seulement si elle a été pressée sur le bloc ?
4. **Effets.** Méthodes utiles : `hide`, `slideToggle`, et `text` (sans et avec argument).
Créez un document HTML ayant un bouton « Afficher ». puis une liste à trois éléments. Attachez au document HTML un script permettant d'obtenir le comportement suivant.
- a. La liste n'est pas visible au chargement. Lorsque l'on clique sur le bouton, la liste se déroule en douceur.
 - b. Lorsqu'on clique sur le bouton et que liste commence à se dérouler, le texte du bouton devient « Masquer ». Si l'on clique à nouveau sur le bouton, la liste se renroule en douceur, et le texte du bouton redevient « Afficher ».
 - c. Ajouter un paragraphe « Bonjour » de grande taille et invisible, et un bouton « Bienvenue ». À l'aide de la méthode `animate`, faire en sorte que lorsqu'on clique sur ce bouton le message « Bonjour » apparaisse à gauche de la page puis se déplace lentement de gauche à droite pour se figer au centre de la page.
 - d. Modifier l'animation pour que le message traverse toute la page de gauche à droite, puis disparaisse en semblant traverser l'extrémité droite de la page.
5. **Faire défiler des images** Créer une page Web qui comporte un bloc d'identifiant « defile », contenant une suite de blocs, chacun contenant une image (vous pouvez prendre les images dans le dossier *images*). Les images seront centrées dans la page, mais initialement non affichées. Ajouter un bouton « suivant ». Le code suivant sera écrit dans une fichier `js` attaché à la page.
- a. À l'aide d'un sélecteur JQuery, sélectionner tous les éléments `img` contenus dans le bloc « defile »—sauvegarder la sélection dans une variable pour plus d'efficacité. Sélectionner (à l'aide de `eq`) la première image (d'index 0) et faire en sorte que seule cette image apparaisse à l'écran.
 - b. Faire en sorte que lorsqu'on clique sur le bouton, l'image courante disparaisse et la suivante soit affichée, de façon circulaire.
 - c. Masquer le bouton. Utiliser la fonction `setInterval` de JavaScript pour faire défiler automatiquement les images toutes les 3 secondes.
 - d. Remplacer les changements d'images (un peu tristes) par les effets suivants :
 - L'image précédente disparaît lentement (effet *fade*), la suivante apparaît lentement (`fadeIn`, `fadeOut` ou `fadeToggle`).
 - L'image suivante se déroule vers le bas (effet *slide*), et la précédente disparaît avec l'effet inverse(`slideDown`, `slideUp` ou `slideToggle`).
 Dans les deux cas, veillez à ce que l'image suivante n'apparaisse pas avant que la première ait complètement disparu (dans le cas contraire, quel serait le problème?).
 - e. Remplacer les changements d'images par un effet de « rotation » : l'image courante disparaît en se déplaçant vers la gauche jusqu'à sortir de la page, l'image suivante en traversant le bord droit de la page et se déplaçant vers le centre de la page (`animate` et les positions relatives de CSS).