

Programmation Web

TP n° 2 : 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 du navigateur, 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 un CDN :

```
<script src="https://code.jquery.com/jquery-3.3.1.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 contenant deux span. 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`).
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 modifié comme précédemment lorsque la souris survole le paragraphe, mais 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 et 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.
- 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, 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, 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 décompte du nombre de mots dans le header et dans le footer sont mis à jour dans le message du bloc central.

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 affichant 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 ce 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).

2 JQuery et formulaires

Eléments utiles :

- pour sélectionner un élément de nom de balise *balise* et de nom *nom* :
`$("balise[name=nom]")` ;
- `val` : récupérer (sans argument) ou modifier (avec argument) la valeur d'un champ d'un formulaire.
- `change` : en argument, une fonction à exécuter si la valeur d'un `input` est modifiée.
- `keyup` : en argument, une fonction à exécuter si une touche est relâchée dans un champ.
- `prop` : récupérer la valeur d'une propriété d'un élément (*e.g.* `checked` pour une `checkbox`).
- `attr` : modifier la valeur d'un attribut (*e.g.* `attr('type', 'text')`), ou ajouter un attribut en spécifiant sa valeur (*e.g.* `attr('disabled', true)` pour un bouton).
- `each` : en argument, une fonction à exécuter sur chacun des éléments d'un groupe d'éléments sélectionnés (dans la fonction : `this` \equiv l'élément courant).
- `prev` : récupérer l'élément qui précède immédiatement un élément donné dans une page web.
- `css` : récupérer ou modifier la valeur d'une propriété css d'un élément, *e.g.* `.css("color":"red");`.

6. Formulaire d'inscription

- a. Créer une page html avec une feuille de style associée.
- b. Créer un formulaire avec les champs suivants et les labels associés :
 - état civil (M. ou Mme) `select`,
 - nom (`input`, `type="text"`),
 - prénom,
 - nom complet (désactivé, `disabled`),
 - date de naissance, (`type="date"`),
 - niveau en JavaScript (`type="range"`),
 - couleur préférée (`type="color"`),
 - mot de passe (`type="password"`)
 - mot de passe (confirmation),
 - afficher le mot de passe (`type="checkbox"`),
 - description (`textarea`),
 - bouton réinitialiser (pour réinitialiser le formulaire, `type="reset"`),
 - bouton envoyer (pour envoyer le formulaire, `type="submit"`).

Formulaire d'inscription

| | |
|-----------------------------|---|
| Etat civil | <input type="text" value="M."/> |
| Nom | <input type="text"/> |
| Prénom | <input type="text"/> |
| Nom complet | <input type="text"/> |
| Date de naissance | <input type="text" value="dd/mm/yyyy"/> |
| Niveau en javascript | <input type="range"/> |
| Couleur preferee | <input type="color" value="#000000"/> |
| Mot de passe | <input type="password"/> |
| Mot de passe (confirmation) | <input type="password"/> |
| Afficher mot de passe | <input type="checkbox"/> |
| Description | <input type="text"/> |

- c. En HTML5 seul (attributs `min` et `max`, à spécifier à la main pour l'instant), faire en sorte que le champ « date de naissance » ne puisse contenir que des dates entre aujourd'hui et il y a 100 ans.
- d. À l'aide du CSS, créer une mise en page similaire à celle de la figure ci-dessus.
- e. À partir de maintenant, on utilisera jQuery pour les questions qui suivent.
Le champ « nom complet » doit changer de valeur en fonction de l'état civil, du nom et du prénom (p. ex. si l'utilisateur a entré respectivement « M. », « Dupont » et « Toto », il faudra afficher « M. Toto Dupont ») (`val`, `change`, voir ci-dessus, ou mieux, `keyup` pour une modification en continu).
- f. La valeur numérique du champ « niveau en JavaScript » doit apparaître à côté de celui-ci (`val`, `change`, ou mieux, `mousemove` pour une modification en continu).
- g. Lorsqu'on sélectionne l'option « afficher le mot de passe », les deux champs précédents affichent le mot de passe en clair (`prop`, `attr`, voir ci-dessus).
- h. Le bouton « envoyer » est désactivé si l'un des champs « nom », « prénom », « date de naissance », « mot de passe » ou « mot de passe (confirmation) » n'est pas rempli. Chacun de ces champs, s'il est vide, doit apparaître en rouge. (`prev`, `css`, voir ci-dessus).
- i. Le champ « mot de passe (confirmation) » apparaît en rouge si que son contenu n'est pas égal à celui du champ « mot de passe ». Le bouton « envoyer » est désactivé si les deux champs « mot de passe » ne coïncident pas (`val`, `attr`).
- j. Le champ « date de naissance » ne peut contenir que des dates entre aujourd'hui et il y a 100 ans, en générant la date du jour avec JavaScript (`attr` avec `'min'` et `'max'`).
- k. Lorsqu'on soumet le formulaire, celui-ci disparaît au profit d'un texte contenant toutes les informations (du type : « Bonjour M. Toto Dupont, vous êtes né le 03/02/2001 ») (`val`, `html`).

- 7. QCM coté client (exercice supplémentaire).** Dans cet exercice, pour les modifications des couleurs, servez-vous de `prev` et de `css`, comme dans l'exercice précédent.
- a. Créer une page contenant un QCM avec 5 questions qui auront chacune 3 réponses possibles.
 - b. Ajouter un bouton « Corriger » qui, au lieu d'envoyer les réponses à un serveur, les vérifie coté client : pour chaque question, si la bonne réponse est sélectionnée, la question apparaît en vert, sinon elle apparaît en rouge (les “bonnes” réponses seront choisies arbitrairement, et écrites en dur dans le code).
 - c. De plus (toujours après appui sur le bouton « Corriger »), lorsqu'une mauvaise réponse a été sélectionnée, la réponse sélectionnée apparaît en rouge et la bonne réponse en vert. Lorsque la bonne réponse a été sélectionnée, elle apparaît en vert.
 - d. Lorsque l'utilisateur clique sur « Corriger » alors que certaines questions n'ont pas de réponse sélectionnée, alors la correction n'a pas lieu et ces questions apparaissent en orange.
 - e. Après la correction, un message apparaît sous le QCM avec la note obtenue.