

Travaux dirigés n°6

Plug-ins jQuery

Dans ce TD, nous comprendrons toute la puissance de jQuery en utilisant des plug-ins, qui nous permettront des résultats impressionnants en quelques lignes de code.

Avant-propos

Lors des deux précédents TDs, nous avons créé à la main un carrousel grâce à jQuery. Le résultat est certes simple, mais fonctionnel. Seulement, vous vous en doutez, vous n'êtes pas les seuls à développer des sites et applications avec jQuery. Et si quelqu'un avait déjà réalisé un carrousel plus évolué que le nôtre ?

Le gros avantage de cette bibliothèque, c'est qu'elle nous permet d'utiliser autant de composants externes que l'on souhaite, pour nous éviter de coder quelque chose que quelqu'un aurait déjà fait ailleurs (et sûrement en mieux, sans offense). N'importe quel développeur peut proposer ses modules à la communauté : c'est ce qu'on appelle un plug-in. Outre le fait que les plug-ins nous permettent de créer des fonctionnalités complètes en quelques lignes, ceux-ci sont en général bien documentés et entièrement personnalisables.

Dans ce TD, nous verrons :

- Un plug-in de carrousel : nivoSlider.
- jQuery UI et ses plug-ins principaux : autocompletion, sélecteur de date...
- Les widgets appliqués à un formulaire.

Pour réaliser ce TD, comme les précédents, vous allez partir de votre code HTML de base (index.html), dans une nouvelle page que vous nommerez **plugins.html**. Les exercices suivants seront à réaliser dans cette page, que vous pourrez ensuite relier au TD3 en y faisant un lien dans le menu.

1. Et le carrousel devint un slider...

Le slider jQuery est un des plug-ins les plus courants. Il permet d'afficher plusieurs images à la suite sur une page web, avec une transition automatique ou lors d'un clic de souris... En bref, exactement ce que nous avons fait lors des deux dernières séances. Sauf que son implémentation est beaucoup plus simple, puisqu'il s'agit d'un plug-in. Il suffit de d'inclure un fichier javascript supplémentaire, et de l'appliquer à un élément.

Dans cet exercice, nous allons implémenter le plug-in **nivoSlider**. Téléchargez la source et placez :

- Le dossier **demo/images/** dans **img/slider/**.
- Le fichier **jquery.nivo.slider.pack.js** dans votre répertoire **js/**.
- Le fichier **nivo-slider.css** dans votre répertoire **css/**.

Vous allez devoir inclure ces deux derniers fichiers dans le <head> votre page HTML, aux côtés des JS et CSS déjà liés. Attention, le nouveau fichier javascript devra être placé après l'inclusion de jQuery (puisque c'est un composant utilisant la bibliothèque) mais avant votre fichier JS contenant votre propre code javascript (script.js).

Vous allez devoir modifier légèrement votre code HTML et CSS pour que celui-ci soit compatible avec le plug-in. Vous pouvez constater que dans la **documentation**, les images à afficher doivent être placées à la suite dans la même <div>. C'est sur cette <div> que le slider est ensuite appliqué.

- Placez les quatre images dans la <div> carrousel.
- Comme dans la documentation, ajoutez le fichier CSS par défaut du slider (ils sont gérés sous forme de thèmes).
- Videz votre fichier javascript qui accueillait le code jQuery du carrousel de la séance précédente (en ayant fait une sauvegarde préalablement).
- Lancez le slider en vous aidant de la documentation avec les paramètres de bases. Ajoutez quelques légendes aux images.
- Modifiez les effets de transition, la vitesse de l'animation, les différents modes de navigation...
- Ajoutez les vignettes correspondant aux images du carrousel.

Comme vous pouvez le constater, vous venez de réaliser en une ligne ce qui vous aviez mis en place en deux semaines. Vous ne ferez plus jamais votre lessive à la main, et vous allez adorer les plug-ins jQuery.

2. jQuery UI

Grâce à une communauté de développeurs très active, un ensemble de plug-ins orientés « graphique » a été créé. Il s'agit de **jQuery UI** (pour « User Interface »). Vous pourrez y trouver des fonctionnalités de drag & drop, des nouveaux effets de transitions ou encore des widgets divers et variés (barre de progression, onglets, sélecteur de couleur...).

Commencez par télécharger jQuery UI. Son contenu étant varié, il est possible de le télécharger module par module, afin d'éviter de charger trop de données inutiles sur sa page web. Ici, nous n'aurons par exemple pas besoin de tous les nouveaux « effets » proposés dans UI. Décochez-les tous, et téléchargez l'archive proposée. (Note : vous pouvez également choisir un style pour tous les widgets : plusieurs CSS sont disponibles). Parcourez les fichiers téléchargés. En plus des CSS, JS et images nécessaires au fonctionnement des plug-ins, vous avez à votre disposition une documentation complète.

Comme dans l'exercice précédent, copiez les fichiers suivants :

- jquery-ui.js dans votre répertoire **js/**.
- jquery-ui.theme dans votre répertoire **css/**.

Joignez-les à votre page HTML (attention encore une fois à l'ordre d'inclusion). Vous êtes maintenant prêt(e) à utiliser tous les plug-ins UI à votre disposition !

3. Formulaires

Bon nombre des plug-ins UI sont utilisables au sein de formulaires. Ils permettent d'ajouter des composants simplifiant la saisie à l'utilisateur, ou d'en valider le contenu. Afin de les tester, nous allons donc en créer un sur notre page HTML.

3.1 - Structure HTML

(Re ?)venons rapidement sur la structure d'un formulaire en HTML. Celui-ci est caractérisé par une balise `<form>`, possédant un certain nombre de composants (`<input>` pour les champs de saisie, `<select>` pour les listes déroulantes, etc.).

Pour une question d'accessibilité, chaque champ doit être accompagné d'un label, qui correspond à une zone textuelle : c'est la description du champ, décrivant à l'utilisateur la saisie attendue. Afin de lier le label avec son champ, le premier prend comme attribut « for » la valeur de l'id du second. Voici un exemple concret avec un champ de texte ayant comme id « name » :

```
<form method="post" action="">  
  <label for="name">Votre nom</label><input type="text" name="name" id="name" />  
</form>
```

- En vous aidant de la [documentation](#) des différents types de composants HTML d'un formulaire, créez-en un sur votre page. Celui-ci devra permettre à l'utilisateur de saisir :
 - Son nom.
 - Son prénom.
 - Son adresse e-mail.
 - Sa date de naissance.
 - Son métier.
 - La façon dont il a entendu parler du site, lui proposant un certain nombre de réponses possibles (`<select>`).
 - Une case à cocher lui demandant s'il désire recevoir une newsletter.
- Mettez en page convenablement en CSS votre formulaire.

3.2 - Autocomplétion

Qui n'a jamais joué avec Google Suggests ? Vous tapez quelques lettres dans le champ de recherche, et plusieurs termes vous sont proposés. Ce procédé s'appelle l'auto-complétion. Il est assez utilisé dans les formulaires... et est possible grâce à jQuery UI !

Dans notre formulaire, nous voulons simplifier la saisie du métier de l'utilisateur. Pour cela, nous allons établir une petite « base de données » des métiers les plus courants, et lui proposer de compléter le champ de texte s'il commence à écrire le nom d'un métier prédéfini.

Commencez par parcourir la documentation du **plug-in jQuery UI autocomplete**. En vous inspirant du code source de l'exemple proposé de la fonctionnalité basique, vous allez devoir :

- Stocker 15 métiers différents dans un tableau dans votre code jQuery.
- Permettre l'autocomplétion sur le champ « métier » de votre formulaire.

3.3 - Sélecteur de date

Pour ce qui est de la date de naissance, jQuery UI permet d'afficher un calendrier qui évite à l'utilisateur de saisir sa date entièrement.

En vous aidant de la documentation de **DatePicker**, affichez ce calendrier lors du clic sur le champ date. Vous prendrez soin que cette date soit au format français : jour/mois/année.

3.4 - Interface d'onglets

En utilisant le **plug-in Tabs**, remplacez l'affichage côte-à-côte des éléments (<div> internes) de la <div> "content", par une interface d'onglets. Chaque <div> interne sera un onglet du nouvel affichage.

4. Aller plus loin

- Utilisez le **plug-in validation** de jQuery (hélas absent de jQuery UI) pour valider votre formulaire. Tous les champs devront être obligatoires, sauf la case à cocher permettant la réception de la newsletter.
- Implémentez l'autocomplétion AJAX (« remote »).