

JQuery ++

Dans ce document, nous allons voir les notions avancées de JQuery. Mon travail va porter sur les plugins JQuery.

Lien avec RIA

Description de la problématique et du contexte

Pourquoi faire un plugin ?

L'utilisation d'un plugin a pour but de permettre une portabilité de notre fonction et réduire les conflits avec notre code à nous. Le code doit être fait pour être utilisé sur différents sites et avoir le moins de modifications possible à faire. Notre code devrait donc fonctionner sur tout les sites et nous évitons de recréer notre fonction.

Pourquoi un slider ?

En effet, un slider n'est pas "révolutionnaire" mais sur la grande majorité des sites pour des clients, il est demandé d'avoir des interactions pour montrer nos réalisations, travaux, produits. Il me semblait donc utile de comprendre un peu comment tout cela fonctionne.

Qu'est-ce qu'un plugin JQuery ?

Un plugin JQuery est simplement une nouvelle methode qui utilise des objets JQuery. Pour créer un nouvel objet JQuery, nous devons appeler

```
JQuery()
```

Qui va hériter des méthodes JQuery. L'idée d'un plugin est de faire quelque chose avec une collection d'éléments. Nous pourrions considerer les méthodes existantes comme des plugins

```
.fadeOut()
```

ou

```
.addClass()
```

Créer son propre plugin

Pour créer un plugin, nous devons ajouter une fonction au nom de notre plugin.

```
$.fn.greenify = function() {
    this.css( "color", "green" );
}
$( "a" ).greenify();
```

Ensuite, tous nos liens deviendront vert.

\$.fn	.greenify	this.css("color", "green");	\$("a").greenify();
instanciation plugin	nom de fonction	actions du plugin	appel du plugin

Slider pas à pas

Comment effectuer le slider ?

Le slider est composé en 2 parties:

- la création des boutons/liens
- l'affichage des images avec les boutons

Créer les buttons

Pour créer les buttons, il faut compter le nombre d'image que possède notre page, la fonction **.each** vous permettra de les compter. Ensuite, j'attribue un id pour chacune de mes images et je rajoute un bouton contenant le même id. ainsi, les images sont liées au boutons

Ajouter un evenement

Oui, nos boutons sont générés mais ils ne font rien, nous allons ajouter un événement **.click** sur lequel nous effectuerons plusieurs actions:

- Récupérer le lien de l'image que nous voulons afficher
- Afficher l'image choisie
- Masquer l'image actuelle

Pour le faire, je me suis basé sur une class css qui affiche les images et par défaut, je masque les autres.

Voici quelques fonctions que j'ai utilisé : **.addClass()** **.removeClass()** **.fadeIn()** **.fadeOut()**

Si vous n'êtes pas à l'aise avec ces fonctions, je vous convie d'aller lire la documentation à ce sujet sur le site officiel de JQuery

Timer sur les images

Pour changer l'image de façon automatique, Il faut utiliser une fonction qui va avoir lieu tout les n milliseconde, il faudra à nouveau avoir le nombre d'images totale et au moment ou on a attend le nombre d'image maximum, on remet la variable à 0 et la boucle continuera.

Si il faut quelques indications, voici quelque chose qui aidera ceux avec un peu plus de difficulté: `setInterval`

Références théoriques

- [Comment créer un plugin JQuery](#)
- [Pourquoi les plugins JQuery](#)

Résumé

Cet exercice propose la création d'un plugin basique pour créer votre propre plugins pour différents sites web. Il est aussi possible de le faire en vanillajs mais ce n'est pas le but de cet exercice.

Références supplémentaires

[Aller plus loin](#)