

Vous trouverez sur le Moodle, une archive qui contient quelques images pour ce projet. Vous trouverez également un fichier swf que vous pouvez visualiser en l'ouvrant avec un navigateur équipé de flash ou bien dans le gestionnaire de fichier en cliquant – tirant – déposant ce fichier dans une fenêtre de navigateur. Cette video montre le résultat à obtenir par votre programme. Faites en sorte de rendre possible le rajout d'images et texte de manière simple sans modifier tout le code js.

Remarque : l'énoncé du TD vous aide à réaliser ce travail. En condition réelle, vous devez pouvoir réaliser ce site sans indication particulière. Il s'agit plus à proprement parler d'un TD sur HTML, CSS avec un peu de jQuery qu'un TD 'lourd' sur jQuery. Vous devez donc déjà être au point. Vous n'utiliserez que jQuery de base (pas de jQuery UI).

Mini site de présentation à volets animés

Vous allez réaliser une page de présentation d'images avec légendes. Les images se trouvent dans les ressources à télécharger et vous avez accès à une vidéo flash qui montre les fonctionnalités du site.



Commencez par définir les styles des différentes parties dans un fichier **css**.

L'ensemble de la page sera contenue dans une **div** de style **.scene**.

Utilisez l'attribut **overflow :hidden** pour vous assurer que le contenu ne débordera pas de la boîte que vous allez définir pour le contenu (utilisez **position, width, height, margin-left, top, left** etc). Choisissez un fond de couleur (vert par exemple) pour voir la boîte de contenu.

Jonglez avec **position :relative** et **position :absolu**.

Fabriquez une **div** pour la pochette centrale (avec le style **.central**). Idem positionnez la et choisissez une taille compatible avec votre image de fond. Rajoutez-y l'image de départ dans une balise ****.

Commencez à attaquer les volets de gauche avec un premier volet. Fabriquez une classe **imgG** avec un attribut **float :left**.

Complétez cette classe avec les infos sur les tailles images (cf image centrale).

Réalisez également un autre volet en réutilisant la classe **imgG** : notez que l'attribut **float** permet d'empiler (ici à gauche) les **divs**.

Pour permettre de retrouver les volets, utilisez la classe "**imgG volet**" au lieu de la classe "**imgG**" seule (jQuery pourra ainsi retrouver les volets, qu'ils soient à droite ou à gauche en cherchant la classe volet – même si elle n'existe pas dans la description CSS).

Faites la même chose à droite. N'oubliez pas d'utiliser **overflow** pour vous assurer que rien ne dépasse des **divs**.

Commencez par initialiser votre code avec un **\$(document).ready()**. Vérifiez avec un alert que le code est bien exécuté au lancement.

On souhaite associer une fonction d'affichage à chaque volet. Comme chaque volet fait partie de la classe volet, vous pouvez associer une fonction par

```
$(".volet").click(function() {  
    ?  
})
```

Vous pouvez évidemment également utiliser **.on()**.

Vérifiez que vous pouvez utiliser un alert (ou une sortie console.log) pour tester cette fonction.

A la place du alert, utilisez un **animate** sur le **width** (jusqu'à **470px**) pour ouvrir le volet. Mettez un temps assez long pour voir l'animation graphique. Utilisez **\$(this)** pour avoir accès au contexte du volet courant.

Pour refermer les autres volets avant d'ouvrir le volet courant, faites d'abord un **animate** sur tous les volets pour réduire le **width** à une bande verticale de **100px**.

Rajoutez un if/else pour gérer le volet courant (qui se ferme s'il était ouvert, et s'ouvre s'il était fermé) – ou bien un **toggle**.

Gérez le clic sur la pochette centrale : ce clic doit fermer tous les volets.

Reste maintenant à gérer les textes sur les pochettes. Commencez par créer un **style txtbulle** pour le texte. Rajoutez également une classe bidon **panneauTxt** qui vous servira à cacher tous les textes en utilisant le sélecteur CSS avec le **hide** de JQuery.

Le placement de chaque div de texte dépend du volet. Utilisez donc des attributs **posLeft** et **posTop** dans vos div de texte pour permettre au programmeur Web de placer son texte où il le souhaite.

Pour permettre de placer ce texte où il le faut, vous devez donc rajouter un traitement pour chaque zone de texte qui lit les valeurs de **posLeft** et **posRight** (par exemple dans les variables a et b) puis modifie ces valeurs par **\$(this).css({'top' :a,'left' :b})**. Utilisez un **.each** pour faire le travail au moment de l'initialisation.

Il ne reste plus qu'à gérer ensuite la disparition progressive ou l'apparition progressive des textes après l'apparition du volet sélectionné.

Testez et vérifiez le comportement de vos pages.