


Créez un tableau de bord avec jQuery

par Michael Wilson ([Auteur](#)) Didier Mouronval ([Traducteur](#))

Date de publication : 07 février 2009

Dernière mise à jour :

(1)

Cet article est la traduction de  **Creating a jQuery Dashboard** du site **Design Shack**. La fonctionnalité "Tableau de bord" sur le système d'exploitation des Macintosh est très agréable et le menu permettant d'ajouter ou retirer des outils joliment fait. Un effet similaire permettrait d'ajouter un ensemble complet de fonctionnalités à un site web avec un simple menu déroulant en haut d'une page. Dans cet article, nous allons détailler comment réaliser cela.

Avant de commencer, vous voudrez probablement voir [un exemple](#) ou [télécharger les sources](#).

Commencer avec Photoshop.....	3
Créer une grille à dupliquer pour le fond.....	3
Sélectionner un pack d'icônes.....	3
Créer une zone de texte pour afficher les informations.....	3
HTML et CSS.....	3
jQuery.....	6
Conclusion et exemple.....	7

Commencer avec Photoshop.

Créer une grille à dupliquer pour le fond.

Commencez par créer une image dans Photoshop de dimension 56x64 pixels. Appliquez ensuite un dégradé de trois couleurs de #111111 à #313131 et de nouveau à #111111. Voici ce que vous devriez obtenir :

Maintenant, créez vos motifs.

Sélectionnez l'outil "Rectangle" (U) et choisissez une forme carrée de 2px et de couleur #000000. Créez un nouveau calque (et vérifiez bien qu'il soit sélectionné !) et commencez par mettre un point n'importe où sur ce calque. Déplacez le point à 3 pixels du bord haut et 1 pixel du bord droit. Dupliquez ce calque 11 fois (afin d'en obtenir 12 au total) et sur chaque calque, déplacez le point de 5 pixels vers le bas.

Vous pouvez maintenant fusionner ces 12 calques. Dupliquez cette colonne de points 6 fois et déplacez la colonne de 8 pixels à chaque fois.

En utilisant la première colonne créée, dupliquez-la encore une fois et déplacez-la de 4 pixels à droite et 2 pixels vers le bas.

Cette nouvelle colonne doit elle aussi être répétée 6 fois et à chaque fois déplacée de 8 pixels à droite. Vous devriez désormais avoir le fond du tableau de bord.

Sélectionner un pack d'icônes.

Pour cet article, j'ai choisi des icônes provenant de  **Smashing Magazine**. A vous de choisir les icônes qui vous conviennent.

Créer une zone de texte pour afficher les informations.

Créez une nouvelle image de 292 pixels de large pour 34 pixels de haut et ajoutez un nouveau calque. Supprimez le fond du calque. Dessinez un rectangle à bords arrondis de proportions fixes 290px par 32px, créez une sélection, appliquez une couleur de fond blanche et ajustez le fond du calque à 10%. Il ne vous reste plus qu'à ajouter un contour blanc.

Voici ce que vous devriez obtenir (j'ai volontairement mis un fond noir pour que vous puissiez voir ce que cela donne) :

Voilà, vous êtes prêts pour ajouter du HTML et du CSS et obtenir ce que nous souhaitons !

HTML et CSS

Pour réaliser cet exemple, vous aurez besoin de la dernière version de jQuery **disponible ici**.

La première des choses à faire est de préparer la page HTML. Vous aurez besoin d'éléments `<div>`, le premier servira de conteneur global, ce qui assurera au tableau de bord de prendre toute la largeur de la page. Le suivant contiendra les icônes, d'une largeur fixe de 954px afin de s'assurer qu'elles seront toutes visibles sur les résolutions d'écrans les plus répandues.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>Tableau de bord</title>
<link href="index.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="jquery-1.2.6.js"></script>
<body>
  <div id="advanceMenu">
    <div id="advanceContainer">
    </div>
  </div>
</body>
</html>
```

Comme vous pouvez le voir, la feuille de style *index.css* est attachée à la page, elle contient quelques styles permettant d'avoir l'aspect souhaité.

```
#advanceMenu {
  background:url(images/gratingRepeat.png) repeat-x;
  height:64px;
  display:none;
}

#advanceContainer {
  width:945px;
  margin:auto;
}
```

La propriété *repeat-x* est appliquée à l'image de fond pour qu'elle s'applique à toute la largeur. La hauteur sera celle de l'image *gratingRepeat.png* créée précédemment.

La *div advanceContainer* a une largeur de 945px pour pouvoir être vue entièrement sur les définitions d'écrans les plus répandues. Actuellement, la résolution minimale à prendre en compte me semble être 1024x768. De plus, le *margin: auto* va contraindre le navigateur à centrer les icônes quelle que soit la taille de la fenêtre. Mais pour que cela fonctionne, la propriété *width* de l'élément doit être définie.

L'étape suivante va être d'ajouter des balises ** qui contiendront les icônes.

```
<ul id="advanceList">
  <li class="home"></li>
  <li class="monitor"></li>
  <li class="green"></li>
  <li class="interact"></li>
  <li class="people"></li>
  <li class="phonebook"></li>
  <li class="school"></li>
  <li class="vault"></li>
  <li class="world"></li>
  <li class="plus"></li>
</ul>
```

Vous pouvez constater que chaque ** possède une classe différente. C'est comme cela que vous pourrez attribuer des propriétés spécifiques à chaque icône. Faites attention de nommer vos classes de façon à pouvoir vous rappeler à quelle catégorie elle correspond.

D'abord, définissez les styles principaux des balises.

```
ul {
  list-style:none;
  margin:0; padding:0; /*défini le padding et le margin des ul */
}
```

```

}

ul#advanceList {
  float:left;
  width:645px;
  margin-top:6px;
}

#advanceList li {
  height:48px;
  width:48px;
  float:left;
  margin-left:15px;
}
    
```

Concernant le *ul*.

Le *list-style* permet de supprimer les puces pour la liste.

Le *padding* et le *margin* sont fixés à 0 pour éviter les différences d'interprétation des navigateurs et assurer un affichage homogène. J'ai affecté ces valeurs à l'ensemble des *ul* de la page car nous allons en utiliser d'autres.

Concernant le *ul advanceList*.

J'ai choisi de positionner les icônes en utilisant la propriété *float*. Je considère que c'est le meilleur moyen de bien contrôler le positionnement. La propriété *width* est très importante avec ce type de positionnement. J'ai choisi 645px pour préserver de la place pour la zone de texte qui sera placée plus tard.

Concernant les *li* de ce *ul*.

Je leur ai affecté un *width* et un *height* (les mêmes dimensions que les icônes) et un flottement à gauche. Cela permet de disposer les *li* horizontalement plutôt que verticalement (style par défaut). La propriété *margin-left* mettra un espacement de 15px entre les icônes.

La partie suivante du CSS répètera (en modifiant un peu) les mêmes propriétés pour chaque classe. J'en expliquerai une pour la compréhension globale. C'est en fait très simple, il s'agit de fixer le *background* pour chaque image et de prévoir de changer de couleur au survol (ce qui ne fonctionnera pas avec IE6).

```

.info {
  background:url(images/icon.png);
}

.info:hover {
  background:url(images/iconHover.png);
}
    
```

Assurez-vous de répéter cela pour chaque classe et chaque image du tableau de bord.

Enfin, vous allez ajouter la zone de texte et lui appliquer des styles. Le HTML ressemble beaucoup à celui des icônes, en utilisant les balises `` et ``, mais cette fois, il n'y aura qu'un élément dans la liste avec tout une série de `<div>` avec des classes spécifiques.

```

<ul id="contactList">
  <li class="advanceTip">
    <div class="plusTip">Plus</div>
    <div class="worldTip">Global</div>
    <div class="vaultTip">Privé</div>
    <div class="schoolTip">Bloc note</div>
    <div class="phonebookTip">Contacts</div>
    <div class="peopleTip">Amis</div>
    <div class="interactTip">Connexion</div>
    <div class="greenTip">Communauté</div>
    <div class="monitorTip">Affichages</div>
    <div class="homeTip">Accueil</div>
  </li>
</ul>
    
```

Même si cela ressemble à du code un peu décourageant, c'est en fait très simple. Chaque `<div>` se rapporte à une icône, ainsi, en utilisant les CSS, une seule `div` sera affichée ou aucune. Assurez-vous d'utiliser des noms de classes en rapport avec l'icône correspondante. Cela sera utile par la suite.

```

ul#contactList {
  margin-top:16px;
  float:right;
  width:292px;
}

.advanceTip {
  background:url(images/textArea.png) no-repeat;
  height:34px;
}

.plusTip, .worldTip, .vaultTip, .schoolTip, .phonebookTip, .peopleTip, .interactTip, .greenTip, .monitorTip, .homeTip {
  display:none;
  height:34px;
  width:292px;
  text-align:center;
  line-height:35px;
  color:white;
  font-family:Verdana, Arial, Helvetica, sans-serif;
}
    
```

Pour commencer, j'ai positionné le `ul` en flottant à droite et lui ai attribué une largeur. Cela placera la zone de texte à droite des icônes puisque la largeur du conteneur d'icônes est attribué pour permettre l'affichage de l'ensemble sur une seule ligne.

J'ai ensuite affecté un `background` pour afficher l'image de la zone de texte créée tout à l'heure, en n'oubliant pas de fixer une taille pour que l'image soit visible.

Toutes les classes des `div` sont à `display: none` car elles doivent être cachées par défaut. Vous les ferez apparaître avec JavaScript.

Chaque `div` possède les mêmes dimensions que l'image de fond, cela permet de positionner le texte avec précision. La propriété `text-align: center` va placer le texte au milieu du champ. Enfin, `line-height: 35px` permet le centrage vertical. Cependant, cette valeur dépendra de la police utilisée, vous devrez donc tâtonner un peu avant de trouver la bonne valeur.

Pour finir, vous aurez besoin d'un bouton pour permettre d'afficher le tableau de bord. Vous devrez donc placer un lien en dehors des blocs précédents, en lui attribuant une classe spécifique qui vous sera utile dans la partie JavaScript.

```

</div>
<a href="#" class="advanceMenuBtn"></a>
<a href="#" class="closeMenuBtn"></a>
</body>
</html>
    
```

Si vous regardez le résultat dans votre navigateur, vous verrez le tableau de bord avec les images qui changent au survol.

C'est un bon début !

jQuery

Il est maintenant temps d'ajouter les fonctions JavaScript.

Faites bien attention de placer le code JavaScript entre les balises `<head>` de votre page.

```
<script
$(document).ready(function(){
  $("a.advanceMenuBtn").click(function () {
    $("a.advanceMenuBtn").toggle();
    $("a.closeMenuBtn").toggle();
    $("div#advanceMenu").slideToggle("slow");
  });

  $("a.closeMenuBtn").click(function () {
    $("a.advanceMenuBtn").toggle();
    $("a.closeMenuBtn").toggle();
    $("div#advanceMenu").slideToggle("slow");
  });
});
</script>
```

Cette première partie du code permet de basculer l'affichage du tableau de bord lorsque l'on clique sur le bouton. Pour être sûr que celui-ci sera masqué au chargement de la page, ajoutez un `display: none` à la `div advanceMenu` dans le CSS.

La toute première ligne demande d'appliquer ce code lorsque la page est chargée, ainsi le code fonctionnera dès que la page sera visible à l'écran. La partie suivante référence le bouton et indique une fonction à lancer lorsque celui-ci est cliqué. Avec jQuery, la syntaxe pour désigner un élément est comparable à celle des CSS. Vous indiquez d'abord la balise (ici 'a') puis sa classe ('`advanceMenuBtn`').

La ligne suivante pointe sur le tableau de bord et indique la fonction à appeler ainsi que les paramètres. `slideToggle` est une fonction interne de jQuery avec différentes vitesses à lui appliquer. Vous pouvez choisir `slow`, `normal` ou `fast`. Le tableau de bord est maintenant caché au chargement de la page puis affiché si l'on clique sur le bouton.

Une dernière partie de JavaScript va permettre d'afficher le texte dans la zone prévue lorsque l'on survole une icône. Encore une fois, le script est répétitif, je ne vous détaillerai qu'un exemple, qui devra être ajusté pour les autres icônes.

```
<script type="text/javascript">
$(document).ready(function(){
  $("li.home").mouseover(function () {
    $("li.advanceTip .homeTip").show();
  });
  $("li.home").mouseout(function () {
    $("li.advanceTip .homeTip").hide();
  });
});
</script>
```

Ce script est très simple. Encore une fois, il est lancé dès que la page est prête et affichée. La ligne suivante appelle l'icône ayant la classe `info` et lance une fonction lorsque la souris passe dessus.

Ensuite, la fonction indique à la classe de texte associée de s'afficher (`show()`). Il faudra annuler cette action pour que les autres textes puissent s'afficher correctement.

C'est précisément ce que font les lignes suivantes, en remplaçant 'mouseover' par 'mouseout' et '.show()' par '.hide()'

Vous devez répéter ces deux fonctions pour chaque icône, en modifiant juste les noms de classes correspondants.


Conclusion et exemple

Voilà, l'ensemble du code est écrit et vous pouvez le voir fonctionner entièrement.

Vous pouvez voir l'exemple sur lequel cet article est basé **sur cette page**. Vous pouvez aussi télécharger l'archive de l'exemple **ici**.



Attention, dans la mesure où IE6 n'accepte le `:hover` que sur les liens, cet exemple fonctionnera mal avec ce navigateur. Toutefois, il est possible de contourner ce problème en utilisant un peu plus de JavaScript.

1 : Cet article est la traduction de  **Creating a jQuery Dashboard** écrit par **Michael Wilson** et publié sur **Design Shack**.