

Créez votre menu interactif en images en HTML/CSS !

Par Dwayn ,
Cyrille Griboval (gcyrillus)
et minato13



www.openclassrooms.com

Sommaire

Sommaire	2
Lire aussi	1
Créez votre menu interactif en images en HTML/CSS !	3
La structure HTML	3
Notre corps HTML5	4
CSS 1/2 : Mise en forme du menu avec les images	5
La mise en forme	5
CSS 2/2 : Mise en place du texte et des finitions	11
Mise en place du texte dans les liens-images	11
Transitions CSS3	14
Le poids d'un sélecteur	15
Amélioration du temps au survol	17
Compléments et variations	18
Même menu, mais de plus petite épaisseur	18
Menu vertical :	20
Effets de zoom au survol et focus	21
Le dernier défi !	25
Q.C.M.	27
Partager	29



Créez votre menu interactif en images en HTML/CSS !



Par

Cyrille Griboval (gcyrrillus) et



Dwayn et



minato13

Mise à jour : 24/11/2012

Difficulté : Intermédiaire



Durée d'étude : 3 heures



Bienvenue dans ce tuto qui vous permettra de créer un menu interactif uniquement codé en HTML/CSS !

Dans ce mini-tuto, je vais - enfin nous allons - comme je l'ai dit au-dessus, vous apprendre à créer un menu, avec de beaux effets, interactif, uniquement en HTML5 / CSS3, sans même avoir besoin de Javascript !

Voici un aperçu de ce que vous saurez faire à la fin de ce tuto :



Nous pourrions naviguer dans les images soit avec Tabulation soit avec la souris 🐭.

Le menu en live 🧙♂️

Si vous êtes sage (🧙♂️), vous aurez droit à quelques variantes de ce menu...



Les seules connaissances requises sont, bien sûr, l'HTML5 et le CSS3, d'ailleurs, le tuto de M@teo21 est parfait pour les apprendre. 😊

Alors, envie ? Eh bien, allons-y !! 🧙♂️

Sommaire du tutoriel :



- La structure HTML
- CSS 1/2 : Mise en forme du menu avec les images
- CSS 2/2 : Mise en place du texte et des finitions
- Compléments et variations
- Q.C.M.

La structure HTML

Pour le tuto, nous utiliserons des images depuis lorempixel.com, celles-ci sont soumises aux droits d'auteurs. Elles peuvent servir à tester vos mises en pages, mais ne peuvent être réutilisées sur votre site, vous devrez les remplacer par les vôtres.

Nous allons donc construire notre menu en les intégrant à nos liens.



Que va contenir notre page HTML ?

Nous aurons des liens qui contiendront des images, ils auront un titre. Et pour finir, nous définirons l'ordre d'accès aux images avec la touche Tabulation.

Le reste se fera avec le CSS 😊.

Notre corps HTML5

Code : HTML- Structure du menu

```
<!DOCTYPE html>
<html lang="fr">

  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css" /> <!-- Fichier CSS -->
    <!--[if lt IE 9]>
<script
src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]>-->
    <title>Mon joli menu en HTML/CSS !</title>
  </head>

  <body>
    <div id="galerie-menu">
      <!-- Notre navigation -->

      <nav>
        <a href="#" tabindex="0" title="Allez, vas y!" ></a>
        <a href="#" tabindex="0" title="J'y vais" ></a>
        <a href="#" tabindex="0" title="Tenir la distance"></a>
        <a href="#" tabindex="0" title="Surf" ></a>
        <a href="#" tabindex="0" title="Sur du velours "></a>
        <a href="#" tabindex="0" title="Coucher de soleil"></a>
        <a href="#" tabindex="0" title="Plongez!" ></a>
        <a href="#" tabindex="0" title="Je vole !" ></a>
        <a href="#" tabindex="0" title="Iler jusqu'au bout"></a>
        <a href="#" tabindex="0" title="Prêt, contact !" ></a>
      </nav>
    </div>
  </body>

</html>
```

Si vous avez bien lu le tuto de M@teo21, cette page ne vous semblera pas difficile.

Trop facile pour nous, grands *WebDevelopeppers* 😊 !! Bon, passons à la suite.

CSS 1/2 : Mise en forme du menu avec les images

La mise en forme

Voici la liste des sélecteurs CSS dont nous aurons besoin pour cibler les éléments de notre menu :

Types :	Lesquels :	Cible :
Les identifiants uniques (aussi appelés id) :	#galerie-menu	La balise qui possèdera l'attribut <code>id="galerie-menu"</code> . En gros, la balise <code><div></code> .
Nom(s) de balise(s) :	La balise <code>a</code> (pour les liens)	<code>Lien</code>
Les pseudos-éléments :	<code>:hover</code> , <code>:focus</code> (Permettra de modifier tout élément HTML sélectionné par la touche Tabulation ou par la souris), <code>:first-child</code> (Premier élément du menu), <code>:before</code> (Permet de générer du contenu avant le contenu d'un élément !)	Une interaction ponctuelle (survol), permanente (focus), une position (first-child), complément de contenu (:before avec content:"").

Avec ces sélecteurs, nous pouvons déjà construire le squelette de notre feuille de style qui ne ciblera que cette boîte et ses éléments :

Code : CSS - style.css

```
#galerie-menu
{
}
#galerie-menu a, #galerie-menu:hover a:focus
{
}
#galerie-menu a:first-child
{
}
#galerie-menu a:hover, #galerie-menu a:focus
{
}
#galerie-menu a:before
{
}
```

Eh bien, il ne reste plus qu'à créer nos règles de style. En avant ! 😊

Notre premier objectif :

Nous allons faire un menu ouvert sur un "*lien par défaut*" tandis que les autres seront rétractés. L'idée est d'avoir un menu qui élargit un lien lorsqu'il est survolé et qui referme les autres afin que le menu conserve toujours la même largeur :



Il va falloir superposer les images de droite à gauche.

C'est beau, hein ? 😊

Le conteneur général

On veut une boîte rectangulaire montrant une image dépliée et les autres repliées. La hauteur et la largeur seront adaptées en fonction de la taille des images et du nombre de liens que l'on veut afficher.

Ici nous avons 10 images de 400 pixels de largeur sur 300 pixels de hauteur (400*300 px) :

Une ouverte et neuf autres visibles sur 30 pixels, largeur suffisante pour y afficher un texte vertical bien lisible.

Ce qui nous fait une boîte de :

1. 300 pixels de hauteur.
2. 670 pixels de largeur (1x400) + (9x30).

Une bordure, un léger effet d'arrondi et un ombrage suffiront à mettre en valeur notre boîte. 😊

Le premier style `#galerie-menu` ressemblera donc à ceci :

Code : CSS - style.css => `#galerie-menu`

```
#galerie-menu
{
  height: 300px;
  width: 670px;
  margin: 1em auto;
  border: solid;
  border-radius: 3px;
  box-shadow: 1px 1px 3px gray;
}
```

Mise en forme des liens

Nous allons tout d'abord faire en sorte que les liens, quand ils ne sont ni sélectionnés, ni survolés, aient 30 px de largeur pour laisser voir une partie de l'image.

Dans une autre étape, ces 30 pixels nous permettront d'afficher le titre des liens.

Par défaut, ce sont des éléments de type *inline* et ils doivent se rétracter sur 30px de largeur.

Il faut modifier leur formatage pour appliquer **height** et **width** et nous avons 2 pistes pour qu'il reste côte à côte : **float** ou **display**.

- **float:right**, est une option séduisante, mais qui a plusieurs défauts :

- Inverser visuellement le sens de lecture des liens et leur ordre de tabulation.
- On va devoir gérer les défauts des flottants qui sortent du flux.
- **display:inline-block**, conservera le caractère **inline** des liens tout en nous donnant la possibilité de nous servir des pseudo-éléments et de leur donner une taille.
Le seul inconvénient est qu'il y aura un espace résiduel entre chacun d'eux.

Nous allons opter pour **display:inline-block** et nous corrigerons plus tard ce petit défaut visuel. 😊

Donc voici notre `#galerie-menu a :`

Code : CSS

```
#galerie-menu a
{
  display:inline-block;
  width:30px;
  height:30px;
}
```

Mise en place du comportement au survol à la souris ou avec Tab des images du menu pour leur défilement

Il faut que l'image soit affichée entièrement lorsque nous passerons la souris ou lorsque que la touche Tab donne le focus. Il suffira juste de modifier la largeur de 400 px :

Code : CSS

```
#galerie-menu a:hover, #galerie-menu a:focus
{
  width:400px; /* Aussi simple que ça ! */
}
```

Mais maintenant il faut définir un lien :

- Toujours entièrement visible par défaut ;
- Qui ne gêne pas les autres ;
- Pour conserver un aspect visuel cohérent au menu au repos.

Nous choisirons le premier lien avec le pseudo-élément `:first-child`.

Cette chose sera possible grâce à **position: absolute** :

Code : CSS

```
#galerie-menu a:first-child
{
  width: 400px;
  position: absolute;
}
```

Petite problématique...



Comment peut-il rester ouvert quand, précédemment, on met tous les liens en 30 pixels de largeur ?

Il y a 2 raisons :

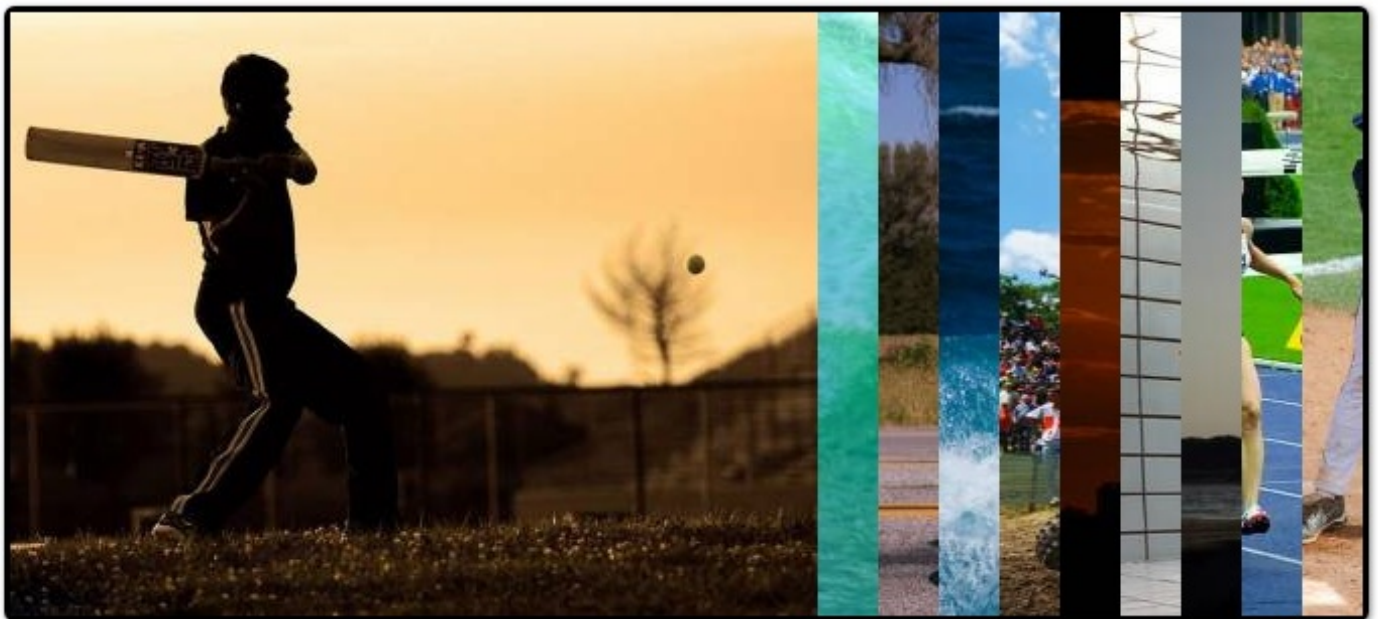
1. De façon générale, la dernière déclaration écrase les précédentes.
2. Il y a la notion de "poids des sélecteurs" qui est importante à connaître.

Et puis, c'est le CSS qui fait ça (*Cascading Style Sheets* ou effets en cascade si vous voyez ce que je veux dire 😊).



Le poids du sélecteur : `#galerie-menu` `a` est plus faible que `#galerie-menu a:first-child`, et si nous le plaçons plus tôt dans le code, il sera appliqué.
Lisez cet article pour davantage d'informations. Je vous conseille vivement de le lire, car il vous permettra de mieux comprendre comment fonctionnent les styles du menu.

Notre menu devrait donc, pour le moment, ressembler à ça :



Voici un lien pour tester, ce sera mieux qu'une image; 😊



Ce n'est pas ça du tout ! Les images ne sont pas toutes visibles et il y a un grand espace blanc vide à droite dans le menu 😞 !!

... qui trouve sa solution.

Nous allons régler ces quelques défauts, sinon nous n'aurions pas fait ce tuto !
Donc vous allez reprendre votre feuille de style, et y ajouter les quelques propriétés CSS utiles qui finaliseront cette première étape.
Si vous voulez, vous pouvez essayer de chercher. 😊

Voilà le tableau :

Propriétés CSS:	A mettre sur le(s) sélecteur(s) :	Pourquoi :
<code>text-align:right;</code> <code>white-space:nowrap;</code> <code>overflow:hidden;</code>	<code>#galerie-menu</code>	Pour remplir cette espace blanc à droite. Pour garder toutes les images sur une seule ligne. Pour cacher ce qui déborderait.

<code>position:relative;</code>	<code>#galerie-menu</code> et <code>#galerie-menu a</code>	Cloisonne la boîte qui va servir de référence de positionnement et activer le z-index .
<code>left:0</code> et <code>top:0</code>	<code>#galerie-menu a:first-child</code>	Pour que le premier lien reste à gauche.
<code>overflow:hidden;</code> <code>vertical-align:top;</code>	<code>#galerie-menu a</code>	Pour que les liens repliés cachent nos belles images qui débordent. Pour que tous nos liens s'alignent bien horizontalement.
<code>z-index:1;</code>	<code>#galerie-menu a</code>	Pour mettre les liens en avant-plan.
<code>z-index:0;</code>	<code>#galerie-menu a:first-child</code>	Pour s'assurer que le premier lien-image reste sur un plan inférieur aux autres. Bah oui car vous pourriez remarquer que les images se superposent en partant de la droite jusqu'à la gauche. 😊

Avec toutes ces propriétés, nous commençons à bien contrôler le comportement de notre menu.



Il reste une espace vide entre les liens !

Nous pourrions utiliser ou modifier d'autres propriétés pour enlever ces espaces, mais selon les navigateurs les techniques CSS à utiliser sont différentes... Nous allons donc simplement adapter notre manière d'écrire et d'indenter notre code HTML en s'attaquant à la véritable source de cet effet :

Ce que nous avons là :

Code : HTML

```
<nav>
  <a href="#" tabindex="0" title="Allez, vas y!"    ></a>
  <a href="#" tabindex="0" title="J'y vais"          ></a>
  <a href="#" tabindex="0" title="Tenir la distance"></a>
  <a href="#" tabindex="0" title="Surf"              ></a>
  <a href="#" tabindex="0" title="Sur du velours " ></a>
  <a href="#" tabindex="0" title="Coucher de soleil"></a>
  <a href="#" tabindex="0" title="Plongez!"          ></a>
  <a href="#" tabindex="0" title="Je vole !"         ></a>
  <a href="#" tabindex="0" title="1er jusqu'au bout"></a>
  <a href="#" tabindex="0" title="Prêt, contact !"  ></a>
</nav>
```

Ce que nous allons en faire :

Secret (cliquez pour afficher)

Code : HTML

```
<nav><a href="#" tabindex="0" title="Allez, vas y!"    >
</a><a href="#" tabindex="0" title="J'y vais"          >
</a><a href="#" tabindex="0" title="Tenir la distance">
</a><a href="#" tabindex="0" title="Surf" >
</a><a href="#" tabindex="0" title="Sur du velours ">
</a><a href="#" tabindex="0" title="Coucher de soleil">
</a><a href="#" tabindex="0" title="Plongez!" >
</a><a href="#" tabindex="0" title="Je vole !" >
</a><a href="#" tabindex="0" title="1er jusqu'au bout">
</a><a href="#" tabindex="0" title="Prêt, contact !" >
</a></nav>

```

En éliminant l'espace dans le code HTML entre la balise fermante et ouvrante de chaque lien, celui-ci disparaît aussi à l'affichage, l'effet est le même lorsque l'on accole les lettres d'un mot les unes aux autres. Rappelez-vous que c'est une particularité des éléments de type **inline** ou **inline-block**.

Ce n'est pas fini !

Il reste à améliorer l'ergonomie du menu en fluidifiant son comportement, afficher le texte des attributs **title** et à l'embellir.

Ça doit faire ~~un peu~~ beaucoup non ? Allez, voici un petit récap' de tout ce que l'on a fait à présent !!

Code : HTML- menu.html

```

<!DOCTYPE html>
<html lang="fr">

  <head>
    <meta charset="UTF-8">
    <title>Mon joli menu en HTML/CSS !</title>
    <link rel="stylesheet" href="style.css" /> <!-- Fichier CSS -
  ->

    <!--[if lt IE 9]>
  <script
src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
  </head>

  <body>
    <div id="galerie-menu">
      <nav><a href="#" tabindex="0" title="Allez, vas y!" >
      </a><a href="#" tabindex="0" title="J'y vais" >
      </a><a href="#" tabindex="0" title="Tenir la distance">
      </a><a href="#" tabindex="0" title="Surf" >
      </a><a href="#" tabindex="0" title="Sur du velours ">
      </a><a href="#" tabindex="0" title="Couché de soleil" >
      </a><a href="#" tabindex="0" title="Plongez!" >
      </a><a href="#" tabindex="0" title="Je vole !" >
      </a><a href="#" tabindex="0" title="1er jusqu'au bout">
      </a><a href="#" tabindex="0" title="Prêt, contact !" >
      </a></nav>
    </div>

```

```
</body>

</html>
```

Code : CSS - style.css

```
#galerie-menu
{
  height:300px;
  width:670px;
  margin:1em auto;
  border:solid;
  border-radius:5px;
  box-shadow:1px 1px 3px gray;
  text-align:right;
  position:relative;
  white-space:nowrap;
  overflow:hidden;
}
#galerie-menu a
{
  display:inline-block;
  width:30px;
  height:300px;
  overflow:hidden;
  position:relative;
  z-index:1;
  vertical-align:top;
}
#galerie-menu a:hover, #galerie-menu a:focus
{
  width:400px;
}
#galerie-menu a:first-child
{
  width:400px;
  position:absolute;
  top:0;
  left:0;
  z-index:0;
}
```

Testez : [Test-1](#)

CSS 2/2 : Mise en place du texte et des finitions

Mise en place du texte dans les liens-images

Nous allons réaliser ceci grâce à des boîtes virtuelles, une fois de plus, grâce aux pseudo-éléments.



Des boîtes virtuelles 🤖 ? Ça sert à quoi ?

Cette boîte virtuelle servira de conteneur et comme tout conteneur nous pourrons stocker des choses dedans. Nous pourrons donc mettre notre texte de présentation d'onglets. Un lien de type **inline-block** permettra l'usage des pseudo-éléments suivants : **:before** et **:after**. Ensuite, nous afficherons notre texte (génération de contenu via la feuille de style) avec la propriété **content: ""**.



Ces balises et leurs contenus sont dits virtuels car ils ne sont pas visibles dans le code source HTML.

Nous allons donc faire une boîte virtuelle de hauteur 30 px et de 300 px de largeur.

Code : CSS

```
#galerie-menu a:before
{
  content: "";
  height: 30px;
  width: 300px;
  position: absolute;
}
```

La propriété `content: ""` permet de placer du texte ou des images dans la boîte virtuelle.

Au lieu d'écrire un texte pour chaque boîte virtuelle des liens, nous allons l'extraire des attributs `title` de chaque lien. Pour récupérer le texte, nous allons mettre `content: attr(title)` ; . Pour finir, nous allons centrer tout ça ! 😊

Donc, voici notre propriété corrigée :

Code : CSS

```
#galerie-menu a:before
{
  content: attr(title);
  text-align: center;
  height: 30px;
  width: 300px;
  position: absolute;
}
```



Par défaut, cette boîte est horizontale, mais nous avons réservé un espace vertical sur nos liens repliés 😊, nous allons changer ça en plus de ce que je vais vous demander en-dessous. Vous ferez mieux de chercher, ça vous fera travailler le CSS. 😊

Voici les critères :

- La boîte sera placée en bas à gauche du lien ;
- Avec un fond translucide (ou transparent si vous voulez) ;
- Avec une bordure (ce sont les contours de la boîte) ;
- Élargie de 100 pixels avec une marge interne droite de façon à centrer le texte sur les 300 premiers pixels ;
- Le texte sera agrandi et coloré en blanc ;
- Le texte sera recentré horizontalement, je vous conseille `line-height` ;

Bon, allez, je vous laisse chercher ! Je reprends les copies dans 5 minutes ! 🐱



N'oubliez pas les propriétés avec IE ! Testez avec Firefox, Safari ou Chrome aussi.

Vous pouvez comparer avec ma solution :

Secret (cliquez pour afficher)

Code : CSS

```
#galerie-menu a:before
{
  content: attr(title);
  text-align: center;
  height: 30px;
  width: 300px;
  position: absolute;
  bottom: 0;
  left: 0;
}
```

```
background:rgba(0,0,0,0.3);
border:1px solid gray;
padding-right:100px;
font-size:30px;
color:white;
line-height:22px;
}
```

Regardez sur [ce lien](#) ce qu'on obtient 😎.

Pour voir le texte sur les liens repliés, nous allons effectuer une rotation des boîtes.

Nous utiliserons la propriété **transform** et une des fonctions qu'elle contient : **rotate (Xdeg)** . (**Xdeg**) étant donné en degrés et non en radians étant donné qu'il est plus facile d'utiliser des degrés par rapport aux radians. 😊

Enfin, on indique le point d'origine pour appliquer les transformations.

Code : CSS

```
#galerie-menu a:before
{
  content:attr(title);
  text-align:center;
  height:30px;
  width:300px;
  position:absolute;
  bottom:0;
  left:0;
  background:rgba(0,0,0,0.3);
  border:1px solid gray;
  padding-right:100px;
  font-size:30px;
  color:white;
  line-height:22px;
  transform:rotate(-90deg);
  transform-origin:17px;
}
```



Oh, dis donc ! Ma boîte ne tourne pas ! 😞

Cela peut être dû à :

- Votre navigateur qui n'a pas encore finalisé et validé l'implémentation de **transform**.
La solution est d'écrire ces règles pour chacun d'eux en ajoutant le préfixe du moteur de rendu.
-moz- pour Firefox, **-webkit-** pour Chrome et Safari, **-o-** pour Opéra et **-ms-** pour IE (mais pour IE, ce préfixe ne fait rien, car il a une autre utilité propre à IE selon les versions).
Sans oublier de laisser en dernier la règle sans préfixe pour les navigateurs qui en ont finalisé l'implémentation.
- Votre navigateur est **trop vieux**, il est probable que la boîte virtuelle ne soit pas générée (en gros, on fait rien et vous mettez à jour votre navigateur si vraiment ça vous manque).
- Votre navigateur est une version de IE qui a besoin d'un *activeX*.
Options :
 - On inclut l'appel de l'activeX spécifique pour Internet Explorer dans la feuille de style (avec ou sans préfixe selon les versions)
 - Rien de grave, on ne fait rien ou on met le navigateur à jour vers la IE 10.

Le conseil de Dueros du pro :

Pour conserver un code propre, facile à maintenir et laisser de côté les quelques navigateurs obsolètes sans que cela ne nuise à la navigation, il y a la solution d'utiliser un script qui va automatiser l'ajout des préfixes lorsque ceux-ci sont nécessaires.

Ce script est fiable et vous le trouverez [là](#). Il y a aussi une version pour jQuery. 😊

Pour ajouter ce script : `<script src="https://raw.githubusercontent.com/LeaVerou/prefixfree/gh-pages/prefixfree.min.js"></script>` après les feuilles de style ou en dessous de la balise `<link>`. Les feuilles de styles appelées avec la règle `@import`, ne sont pas prises en compte.

Transitions CSS3

Afin de créer une ouverture fluide, nous allons définir un laps de temps pour l'ouverture de l'onglet avec la propriété `transition` qui va prendre comme valeur `1s` soit 1 seconde.

Le problème est que nous ne savons pas où mettre ceci, eh bien ce sera aux sélecteurs `a` et `a:hover`. Et donc, au final, notre feuille style.css aura cette tête-là :

Code : CSS - style.css

```
#galerie-menu
{
  height:300px;
  width:670px;
  margin:3em auto;
  border:solid;
  border-radius:5px;
  box-shadow:1px 1px 3px gray;
  text-align:right;
  position:relative;
  white-space:nowrap;
  overflow:hidden;
}
#galerie-menu a
{
  display:inline-block;
  vertical-align:top;
  width:30px;
  height:300px;
  overflow:hidden;
  position:relative;
  z-index:1;
  box-shadow:0 0 15px white;
  transition:1s;
}
#galerie-menu a:hover , #galerie-menu a:focus
{
  width:400px;
  transition:1s;
}
#galerie-menu a:first-child
{
  width:400px;
  position:absolute;
  top:0;
  left:0;
  z-index:0;
}
#galerie-menu a:before
{
  content:attr(title);
  text-align:center;
  height:30px;
  width:300px;
  position:absolute;
  bottom:0;
  left:0;
  padding-right:100px;
  font-size:30px;
  color:white;
  border:1px solid gray;
  box-shadow:0 0 15px white;
  background:rgba(0,0,0,0.3);
  transform:rotate(-90deg);
  transform-origin:17px;
```



```
line-height:22px;
}
```

Le poids d'un sélecteur



Eh ! Si j'ai un lien qui a déjà le focus grâce à la tabulation et que j'en survole un autre avec la souris, mon menu tente de garder 2 liens ouverts, tout est cassé.
Alors, comment je corrige ce bug ?

Cela est dû au poids d'un sélecteur qui détermine quel sélecteur sera prioritaire. J'explique :

Le sélecteur `a:focus` est plus précis que `a`, il reste donc prioritaire.

Pour que notre lien actif se rétracte et laisse de la place aux autres au moment du survol de la souris sur la boîte, il suffit de le lui dire via la feuille de style avec un sélecteur encore plus précis.

La propriété `width:30px` est déjà appliquée au lien par défaut avec un simple sélecteur.

Ajoutons un sélecteur de poids supérieur à `#galerie-menu a:focus` pour ce cas de figure.

Comment ?

Simplement en précisant la situation dans laquelle il doit se refermer `#galerie-menu:hover a:focus` et en accolant ce sélecteur à celui déjà existant `#galerie-menu a` :

Code : CSS

```
#galerie-menu a , #galerie-menu:hover a:focus
{
  display:inline-block;
  width:30px;
  height:30px;
  overflow:hidden;
  position:relative;
  z-index:1;
  box-shadow:0 0 15px white;
  transition:1s;
}
```

En quittant la zone de survol, le lien actif s'ouvrira à nouveau.

Nous pouvons maintenant faire un récap' des codes...

Code : HTML- menu.html

```
<!DOCTYPE html>
<html lang="fr">

  <head>
    <meta charset="UTF-8">
    <title>galerie menu</title>
    <link rel="stylesheet" href="style-css3-A.css" />
    <!--[if lt IE 9]>
<script
src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]>-->
    <script src="https://raw.githubusercontent.com/LeaVerou/prefixfree/gh-
pages/prefixfree.min.js"></script> <!-- Fichier JavaScript assure
support CSS3 -->
  </head>

  <body>
    <div id="galerie-menu">
      <nav><a href="#" tabindex="0" title="Allez, vas y!"
>
```



```

    </a><a href="#" tabindex="0" title="J'y vais" >
    </a><a href="#" tabindex="0" title="Tenir la distance">
    </a><a href="#" tabindex="0" title="Surf" >
    </a><a href="#" tabindex="0" title="Sur du velours ">
    </a><a href="#" tabindex="0" title="Coucher de soleil">
    </a><a href="#" tabindex="0" title="Plongez!" >
    </a><a href="#" tabindex="0" title="Je vole !" >
    </a><a href="#" tabindex="0" title="1er jusqu'au bout">
    </a><a href="#" tabindex="0" title="Prêt, contact !" >
    </a></nav>
  </div>
</body>

</html>

```

Code : CSS - style.css

```

#galerie-menu
{
  height:300px;
  width:670px;
  margin:3em auto;
  border:solid;
  border-radius:5px;
  box-shadow:1px 1px 3px gray;
  text-align:right;
  position:relative;
  white-space:nowrap;
  overflow:hidden;
}
#galerie-menu a, #galerie-menu:hover a:focus
{
  display:inline-block;
  vertical-align:top;
  width:30px;
  height:300px;
  overflow:hidden;
  position:relative;
  z-index:1;
  box-shadow:0 0 15px white;
  transition:1s;
}
#galerie-menu a:hover , #galerie-menu a:focus
{
  width:400px;
  transition:1s;
}
#galerie-menu a:first-child
{
  width:400px;
  position:absolute;
  top:0;
  left:0;
  z-index:0;
}
#galerie-menu a:before
{
  content:attr(title);
  text-align:center;
}

```

```

height:30px;
width:300px;
position:absolute;
bottom:0;
left:0;
padding-right:100px;
font-size:30px;
color:white;
border:1px solid gray;
box-shadow:0 0 15px white;
background:rgba(0,0,0,0.3);
transform:rotate(-90deg);
transform-origin:17px;
line-height:22px;
}

```

Téléchargeable ici : [http://doc-css.com/tutoriel/sdz/menu-i \[...\] f/etape4.html](http://doc-css.com/tutoriel/sdz/menu-i [...] f/etape4.html)

Amélioration du temps au survol

Il est beau, il est bien notre menu ! Il est même un peu trop... réactif ! 😊

Nous allons tempérer un peu ça !

Nous allons ralentir le début de l'effet de transition en utilisant une courbe de Bézier et allonger le temps de la transition à 2 secondes. Comme ceci :

```
transition-timing-function: cubic-bezier(0.00,0.01,0.00,1.00);
```

Pourquoi ?

Tout simplement que la courbe de Bézier va nous permettre de contrôler la vitesse de transition pendant le temps de la transition, par défaut, la courbe de transition est linéaire.

La feuille de style rectifiée :

Code : CSS - style.css

```

#galerie-menu
{
  height:300px;
  width:670px;
  margin:3em auto;
  border:solid;
  border-radius:5px;
  box-shadow:1px 1px 3px gray;
  text-align:right;
  position:relative;
  white-space:nowrap;
  overflow:hidden;
}
#galerie-menu a, #galerie-menu:hover a:focus
{
  display:inline-block;
  vertical-align:top;
  width:30px;
  height:300px;
  overflow:hidden;
  position:relative;
  z-index:1;
  box-shadow:0 0 15px white;
  transition:2s;
  transition-timing-function: cubic-bezier(0.00,0.01,0.00,1.00);
}
#galerie-menu a:hover , #galerie-menu a:focus
{
  width:400px;
}

```

```

    transition:2s;
    transition-timing-function: cubic-bezier(0.00,0.01,0.00,1.00);
  }
  #galerie-menu a:first-child , #galerie-menu:hover a:focus:first-
  child
  {
    width:400px;
    position:absolute;
    top:0;
    left:0;
    z-index:0;
  }
  #galerie-menu a:before
  {
    content:attr(title);
    text-align:center;
    height:30px;
    width:300px;
    position:absolute;
    bottom:0;
    left:0;
    padding-right:100px;
    font-size:30px;
    color:white;
    border:1px solid gray;
    box-shadow:0 0 15px white;
    background:rgba(0,0,0,0.3);
    transform:rotate(-90deg);
    transform-origin:17px;
    line-height:22px;
  }

```

Pour adapter la courbe de Bézier à votre goût : <http://cssglue.com/cubic>.

Voici le menu enfin fini.

Compléments et variations

Même menu, mais de plus petite épaisseur



On utilise le même principe que pour la largeur:
Deux états :

- Survolé
- Au repos

On ajoute donc :

Code : CSS - style.css

```

/* Réduction du menu et des liens à 30px de hauteur et transition */
#galerie-menu , #galerie-menu a
{
  height:30px;
  transitions:1s;
}
/* Ouverture en hauteur du menu au survol et transition */
#galerie-menu:hover , #galerie-menu:hover a
{
  height:300px;
}

```

```
transition:1s;
}
```

Et on ajuste `#galerie-menu a:before` :

Pour la rotation du texte, on aura juste à retirer `transform:rotate(-90deg)` et `transform-origin:17px`.

On aura plus qu'à réajuster le `line-height` : `line-height:28px`.

Voilà la feuille complète :

Code : CSS - style.css

```
#galerie-menu
{
  height:300px;
  width:670px;
  margin:3em auto;
  border:solid;
  border-radius:5px;
  box-shadow:1px 1px 3px gray;
  text-align:right;
  position:relative;
  white-space:nowrap;
  overflow:hidden;
}
#galerie-menu a, #galerie-menu:hover a:focus
{
  display:inline-block;
  vertical-align:top;
  width:30px;
  height:300px;
  overflow:hidden;
  position:relative;
  z-index:1;
  box-shadow:0 0 15px white;
  transition:2s;
  transition-timing-function: cubic-bezier(0.00,0.01,0.00,1.00)
}
#galerie-menu a:hover , #galerie-menu a:focus
{
  width:400px;
  transition:2s;
  transition-timing-function: cubic-bezier(0.00,0.01,0.00,1.00)
}
#galerie-menu a:first-child, #galerie-menu:hover a:focus:first-child
{
  width:400px;
  position:absolute;
  top:0;
  left:0;
  z-index:0;
}
#galerie-menu a:before
{
  content:attr(title);
  text-align:center;
  height:30px;
  width:300px;
  position:absolute;
  bottom:0;
  left:0;
  padding-right:100px;
  font-size:30px;
  color:white;
  border:1px solid gray;
  box-shadow:0 0 15px white;
  background:rgba(0,0,0,0.3);
  line-height:28px;
}
```

```
#galerie-menu , #galerie-menu a
{
    height:30px;
    transitions:1s;
}
#galerie-menu:hover , #galerie-menu:hover a
{
    height:300px;
    transition:1s;
}
```

Menu vertical :



Pour cette variante, la feuille de style sera réduite. Nous aurons 3 états :

- Survolé ;
- Actif, via Tab ou passage de la souris ;
- Ni survolé ni actif ;

Il n'y a plus de positionnement absolu, excepté pour le texte réinjecté via **content**: "..."; , un positionnement relatif reste

cependant nécessaire.

La diminution du nombre de sélecteurs est conséquente du fait du peu de style à mettre en œuvre pour ce type de menu.

Bonne chance !

La feuille de style raccourcie et modifiée :

Code : CSS - style.css

```
#galerie-menu
{
  width:400px;
  margin:3em auto;
  border:solid;
  border-radius:5px;
  box-shadow:1px 1px 3px gray;
  text-align:right;
}
#galerie-menu a, #galerie-menu:hover a:focus
{
  display:block;
  height:30px;
  position:relative;
  overflow:hidden;
  box-shadow:0 0 15px white;
  transition:2s;
  transition-timing-function: cubic-bezier(0.00,0.01,0.00,1.00);
}
#galerie-menu a:hover , #galerie-menu a:focus
{
  transition:2s;
  transition-timing-function: cubic-bezier(0.00,0.01,0.00,1.00);
  height:300px;
}
#galerie-menu a:before
{
  content:attr(title);
  text-align:center;
  width:400px;
  position:absolute;
  top:0;
  left:0;
  font-size:30px;
  color:white;
  border:1px solid gray;
  box-shadow:0 0 15px white;
  background:rgba(0,0,0,0.3);
  line-height:28px;
}
```

Effets de zoom au survol et focus



Pour cette série de versions, la première différence est le format de l'image qui est plus petite et carrée. Il faudra donc éliminer dans la feuille de style tout ce qui vise à modifier la hauteur et la largeur des liens, ainsi que les sélecteurs inutiles.

La première partie de la mise en forme consiste à styler liens et images, en carré de 120px à 150px de côté avec une bordure. On garde le **title** affiché au-dessus sans autre mise en forme particulière. Nous garderons aussi la transition sur **a** et **a:hover**, **a:focus**.

Nous redimensionnerons les boîtes pour qu'elles s'affichent sur 2 lignes, 5 images par ligne.



Pour les vieux navigateurs, il se peut que qu'ils n'affichent pas ce que vous voulez. Il est grandement temps de changer de navigateur !!! 😞

Donc nous avons 2 échelles de zoom à répartir sur 3 états :

1. Survolé
2. Actif
3. Par défaut

Afin de donner un bel effet de zoom sur notre menu ainsi que de l'éclaircir, nous allons d'abord effectuer un **zoom** de 0.75 par défaut sur les liens et un **zoom** de 2.1 pour le survol et focus. 😊



Je ne comprends pas ! La propriété CSS **zoom** est invalide et compatible uniquement avec des versions d'Internet Explorer.

Comment je fais avec les autres navigateurs 😞 ?

En CSS3, les fonctions de modifications visuelles en 2D et 3D sont introduites avec la propriété **transform**.

Donc au lieu d'utiliser **zoom** nous utiliserons **transform:scale()**.

Ce qui donne :

Code : CSS - style.css

```
#galerie-menu a
{
    transform:scale(0.75);
}
#galerie-menu a:hover, #galerie-menu a:focus
```



```
{  
    transform:scale(2.1);  
}
```

On définit le point de départ de la transformation avec **transform-origin: top center;** et on règle le **z-index** sur les différents états.

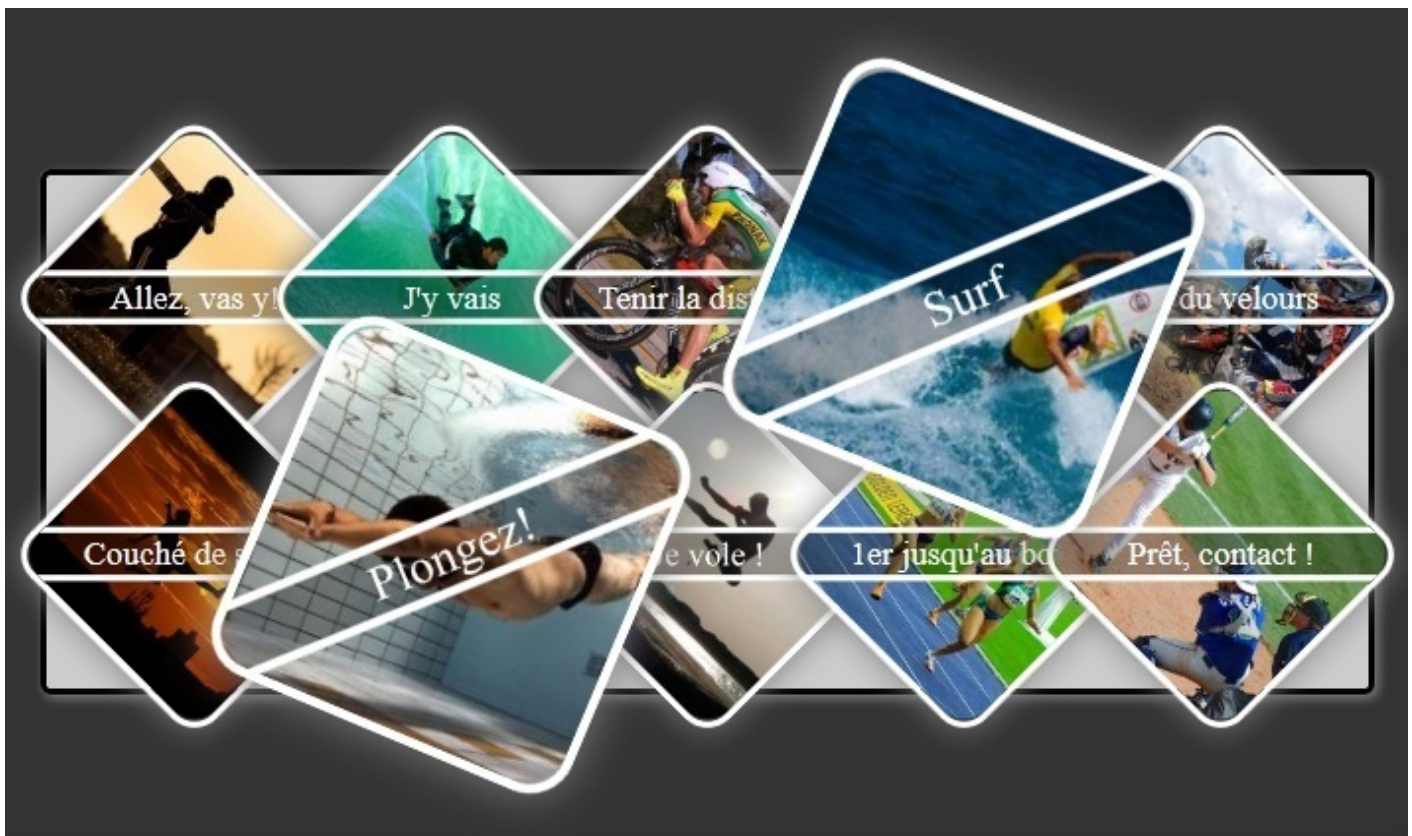
Notre feuille de style pour cette version de notre menu:

Code : CSS - style.css

```
#galerie-menu  
{  
    width:670px;  
    margin:auto;  
    border:solid;  
    border-radius:5px;  
    box-shadow:2px 2px 5px gray;  
    background:#ccc;  
    text-align:center;  
}  
#galerie-menu a img  
{  
    height:120px;  
    width:120px;  
    border:none;  
    vertical-align:top;  
}  
#galerie-menu a  
{  
    border-radius:15px;  
    display:inline-block;  
    transform:scale(0.75);  
    overflow:hidden;  
    box-shadow: 0 0 15px gray;  
    border:solid white;  
    transition:1s;  
    z-index:1;  
}  
#galerie-menu a:hover,#galerie-menu a:focus  
{  
    transform:scale(2.07);  
    transform-origin:top center;  
    z-index:5;  
    background:white;  
    transition:1s;  
}  
#galerie-menu a:before  
{  
    content:attr(title);  
    position:absolute;  
    color:white;  
    bottom:0;  
    box-shadow:0 0 3px white;  
    display:block;  
    width:100%;  
    background:rgba(0,0,0,0.3);  
}
```

Continuons à modifier l'aspect général avec une rotation de 45 degrés des liens sans zoom négatif et avec le texte mis en travers.





La propriété **transform** peut prendre plusieurs valeurs ou fonctions.

Nous ajouterons **rotate (45deg)** et **scale (1)** pour les valeurs par défaut et pour l'état survolé ou actif: **rotate (0)** et **scale (2.1)**.

L'élément virtuel **:before** avec le texte de l'attribut **title**, va lui aussi être mis en rotation avec une bordure. Ce sont à peu près les seules choses à faire.

Code : CSS - style.css

```
#galerie-menu
{
  width:670px;
  margin:auto;
  border:solid;
  border-radius:5px;
  box-shadow:2px 2px 5px gray;
  background:#ccc;
  text-align:center;
}
a img
{
  height:120px;
  width:120px;
  border:none;
  vertical-align:top;
  border-radius:10px;
}
a
{
  border-radius:15px;
  display:inline-block;
  transform: scale(1) rotate(45deg);
  overflow:hidden;
  position:relative;
  box-shadow: 0 0 15px gray;
  border:solid white;
```

```

background:white;
transition:1s;
z-index:1;
height:120px;
width:120px;
}
a:hover,a:focus
{
transform: scale(2.07) rotate(0);
transform-origin:top center;
z-index:5;
background:white;
transition:1s;
}
a:before
{
content:attr(title);
position:absolute;
transform:rotate(-45deg);
transform-origin: 130px 64px;
border:solid;
color:white;
display:block;
width:160px;
text-align:center;border-radius:15px;
background:rgba(0,0,0,0.3);
}

```

Le dernier défi !

Bon là vous allez réaliser un menu de qualité ! C'est le dernier exo, et puis il sera temps de nous quitter 😊.



Quelques pistes :

- Ombres multiples avec box-shadow ;
- Border-style ;
- Background ;
- Optionnel, le centrage vertical du menu.

Vous donnez votre langue à la souris ? 😊

Secret (cliquez pour afficher)

Code : CSS - style.css

```
#galerie-menu
{
    width:670px;
    margin:auto;
    border:dashed red 15px;
    border-radius:15px;
    box-shadow: 0 0 1px black, inset 0px 0px 5px black, 0 0 50px 50px
    white ;
    background:yellow;
    text-align:center;
}
a img
{
    height:120px;
    width:120px;
    border:none;
    vertical-align:top;
    border-radius:10px;
}
a
{
    border-radius:15px;
    display:inline-block;
    transform: scale(1.1) rotate(45deg) ;
    overflow:hidden;
    position:relative;
    box-shadow: 0 0 15px gray;
    border:solid white;
    background:white;
    transition:1s;
    z-index:1;
    height:120px;
    width:120px;
}
a:hover,a:focus
{
    transform: scale(2.07) rotate(0deg) ;
    transform-origin:top center;
    z-index:5;
    background:white;
    transition:1s;
}
a:before
{
    content:attr(title);
    position:absolute;
    transform:rotate(-45deg);
    transform-origin: 130px 64px;
    border:solid;
    color:white;
    display:block;
    width:160px;
    text-align:center;border-radius:15px;
    background:rgba(0,0,0,0.3);
}

/* En option : */
html
{
    display:table;
    height:100%;
    width:100%;
    background:#00664E
}
```

```
body
{
  display:table-cell;
  vertical-align:middle;
}
```



Petite variante... 🤖

Pour faire les ronds, il suffira de caler le border-radius de **a**, **img** à au moins la moitié de leur hauteur/largeur.

Voilà pour ces exercices ! 😊

Q.C.M.

Le premier QCM de ce cours vous est offert en libre accès.
Pour accéder aux suivants

[Connectez-vous](#) [Inscrivez-vous](#)

Qu'est-ce-que **tabindex** ?

- ☐ Une balise HTML
- ☐ Un attribut HTML
- ☐ Une propriété CSS
- ☐ Un attribut HTML dédié à l'accessibilité
- ☐ Une propriété CSS spéciale IE.

Quelles **propriété** et **valeur** permettent tout ceci :

- Déplacer visuellement un élément de sa position.
- Déterminer une zone d'affichage comme référence.
- De mettre en œuvre la propriété z-index.
- De cloisonner une zone.
- Provoquer un reflow sur l'affichage de l'élément.
- ☐ **float:left;**
- ☐ **position:absolute;**

- ☐ `position: relative;`
- ☐ `display: inline-block;`
- ☐ Aucune
- ☐ Les deux premières réponses
- ☐ `box-shadow: 0 0 0 transparent;`

Pour un élément positionné en relatif, l'effet des règles

- ☐ `- left: 50px;`
- ☐ `- margin-left: 50px;`
- ☐ Sont identiques
- ☐ S'annulent mutuellement
- ☐ Se cumulent visuellement
- ☐ Sont identiques à `margin-left: 100px`
- ☐ Sont identiques à `left: 100px`

Quel est la particularité de la valeur `inline-block` pour la propriété `display` ?

- ☐ Elle remplace la règle `float: left;`
- ☐ Elle remplace la règle `float: right;`
- ☐ Elle pose des problèmes de lecture
- ☐ Elle modifie le contexte de formatage d'un élément qui hérite alors d'un comportement similaire a celui du texte ou d'une image.
- ☐ Elle permet de centrer une boîte

Qu'est ce que `:before` ?

- ☐ C'est un attribut.
- ☐ C'est un pseudo-élément.
- ☐ C'est une balise.

Qu'est-ce qu'une "boîte virtuelle" ?

- ☐ Une boîte informatique !
- ☐ Une boîte en carton avec du papier binaire collé dessus.
- ☐ Une boîte imaginaire en informatique.
- ☐ Une boîte qui contient des propriétés.
- ☐ Un conteneur généré via CSS

Quel effet à la règle CSS : `content: attr (title)` ?

- ☐ A injecter le contenu de l'attribut `title` de la balise ciblée.
- ☐ Ce n'est pas une règle CSS, mais du JavaScript !
- ☐ A contenir quelque chose.

Comment enlever le soulignement des textes et les bordures des images ?

- ☐ `a { text-decoration: none; border: none; outline: none }`
- ☐ `a { text-decoration: none; outline: none } a img { border: none; }`
- ☐ `a, img { text-decoration: none; border: none; outline: none }`
- ☐ `#menu { text-decoration: none; border: none; outline: none }`

Correction !

Statistiques de réponses au QCM

Avec ce tuto, nous venons de découvrir quelques aspects de CSS3 en produisant un menu réactif du plus bel effet.

Ces menus sont généralement construits et rendus dynamiques à l'aide de JavaScript, ce tuto est une parfaite occasion de créer des belles choses et d'approfondir vos connaissances en HTML/CSS.

Pour ceux qui veulent aller plus loin, un de nos auteurs vous a fait un [fichier à télécharger](#) comprenant tout les fichiers du tuto

que vous pourrez réutiliser 😊 .

Faites-nous de beaux menus et postez-les dans les commentaires ! 😊



La totalité des caractéristiques des styles du menu ne fonctionnent pas encore sous tous les navigateurs actuels (11/2012) ! Soyez indulgents avec vos navigateurs, le menu, lui, est et restera **fonctionnel**.

Partager

