
Un peu de CSS3

Il est possible d'utiliser certaines propriétés CSS3 dès aujourd'hui¹. CSS3 permet entre autre de gérer certains effets interactifs à la place de technologies plus lourdes telles que Flash et JavaScript. Dans cette étude, nous ciblerons en particulier les propriétés suivantes qui peuvent s'avérer être très utiles :

- `border-radius` : arrondit les coins d'un élément selon un rayon spécifié ;
- `text-shadow` : ajoute une ombre à du texte ;
- `box-shadow` : ajoute une ombre à un élément ;
- Images d'arrière-plan multiples avec `background` ;
- `opacity` : définit l'opacité d'un élément ; 1 pour complètement opaque et 0 pour complètement transparent ;
- `rgba` : nouveau modèle de couleur permettant de spécifier un niveau d'opacité ;
- `transition` : permet au changement d'une propriété CSS de se produire progressivement sur une durée spécifiée ;
- `transform` : permet de modifier certaines caractéristiques des éléments avec `scale`, `rotate`, `skew` et `translate`.

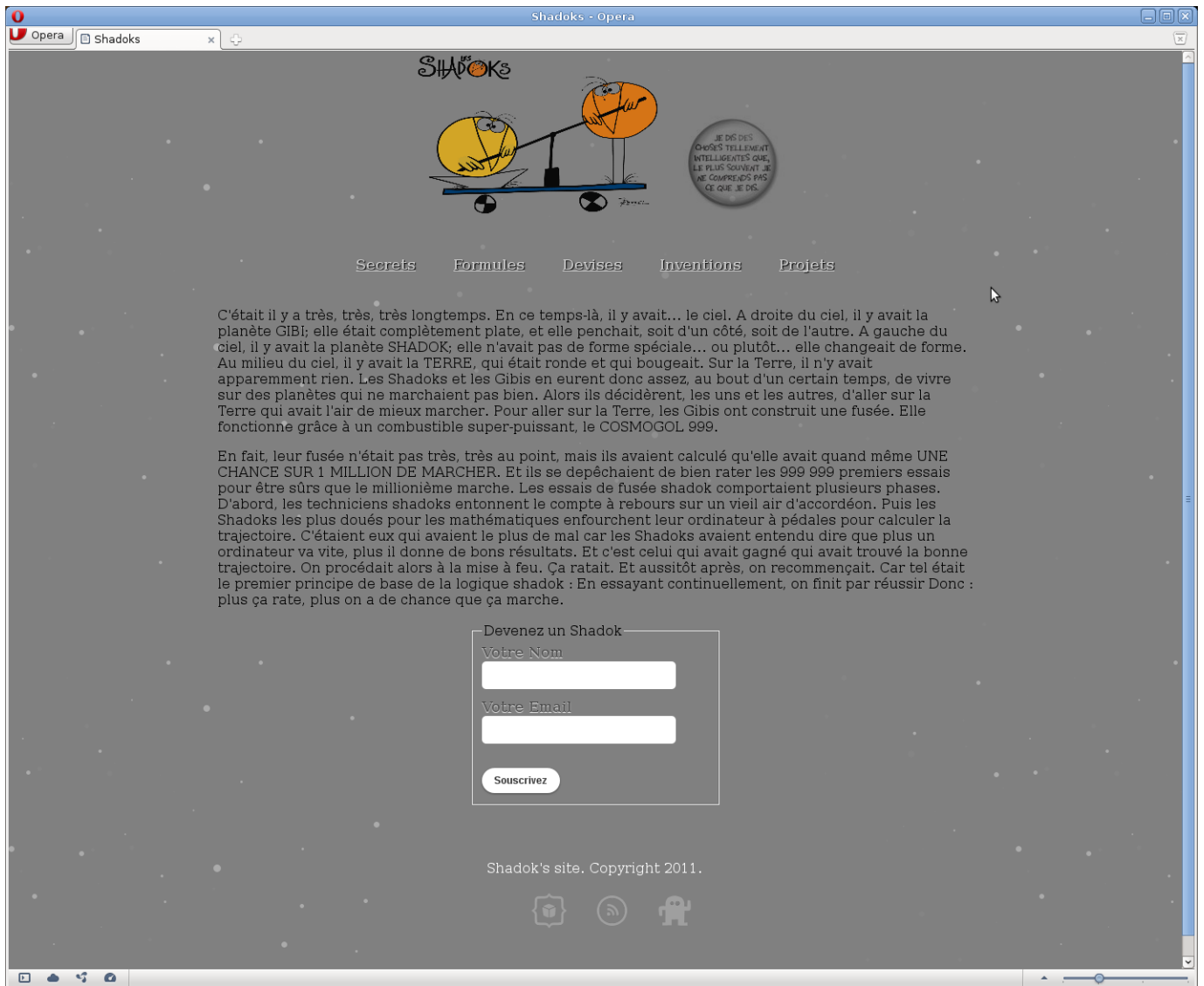
La plupart de ces propriétés seront reconnues par :



Pour en savoir plus sur ces propriétés et leurs arguments, voir par exemple <http://www.w3schools.com/css3/default.asp>

Nous allons expérimenter avec une page web telle que celle affichée sur la page suivante, et structurée comme indiquée par le code source HTML qui suit.

1. Le sujet de ce TP est basé sur les éléments du livre "CSS3 pour les Web designers" par Dan Cederholm chez Eyrolles (12 euros).



```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD_XHTML_1.0_Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    <title>Shadoks</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link type="text/css" rel="stylesheet" href="shadoks.css" />
  </head>

  <body>
    <div id="header">
      
      
      <ul id="nav">
        <li><a href="#">Secrets</a></li>
        <li><a href="#">Formules</a></li>
        <li><a href="#">Devises</a></li>
        <li><a href="#">Inventions</a></li>
        <li><a href="#">Projets</a></li>
      </ul>
    </div>
  </body>
</html>

```

```

</ul>
</div>

<p>
C'était il y a très, très, très longtemps. En ce temps-là, il y avait... le
ciel. A droite du ciel, il y avait la planète GIBI; elle était
complètement plate, et elle penchait, soit d'un côté, soit de l'autre. A
gauche du ciel, il y avait la planète SHADOK; elle n'avait pas de forme
spéciale... ou plutôt... elle changeait de forme. Au milieu du ciel, il y
avait la TERRE, qui était ronde et qui bougeait. Sur la Terre, il n'y
avait apparemment rien. Les Shadoks et les Gibis en eurent donc assez, au
bout d'un certain temps, de vivre sur des planètes qui ne marchaient pas
bien. Alors ils décidèrent, les uns et les autres, d'aller sur la Terre
qui avait l'air de mieux marcher. Pour aller sur la Terre, les Gibis ont
construit une fusée. Elle fonctionne grâce à un combustible super-puissant
, le COSMOGOL 999.

</p>
<p>
En fait, leur fusée n'était pas très, très au point, mais ils avaient calculé
qu'elle avait quand même UNE CHANCE SUR 1 MILLION DE MARCHER. Et ils se
depêchaient de bien rater les 999 999 premiers essais pour être sûrs que
le millionième marche. Les essais de fusée shadok comportaient plusieurs
phases. D'abord, les techniciens shadoks entonnent le compte à rebours sur
un vieil air d'accordéon. Puis les Shadoks les plus doués pour les
mathématiques enfourchent leur ordinateur à pédales pour calculer la
trajectoire. C'étaient eux qui avaient le plus de mal car les Shadoks
avaient entendu dire que plus un ordinateur va vite, plus il donne de bons
résultats. Et c'est celui qui avait gagné qui avait trouvé la bonne
trajectoire. On procédait alors à la mise à feu. Ça ratait. Et aussitôt
après, on recommençait. Car tel était le premier principe de base de la
logique shadok : En essayant continuellement, on finit par réussir. Donc :
plus ça rate, plus on a de chance que ça marche.

</p>

<form action="/" method="post" id="idForm">
  <fieldset>
    <legend>Devenez un Shadok</legend>
    <label for="idNom">Votre Nom</label>
    <input type="text" id="idNom" />
    <label for="idEmail">Votre Email</label>
    <input type="text" id="idEmail" /> <br /> <br />
    <input type="submit" value="Souscrivez" />
  </fieldset>
</form>

<div id="footer">
  <p>Shadok's site. Copyright 2011.</p>
  <ul>
    <li><a href="#"></a></li>
    <li><a href="#"></a></li>
    <li><a href="#"></a></li>
  </ul>
</div> <!-- /footer -->
</body>
</html>

```

1 Exercice

1. Créer le fichier `menu.html` avec le contenu donné ci-dessus.
2. Créer le fichier `menu.css` avec 3 règles telles que :

- pour le sélecteur `body`, on fixe la largeur à 900px, la marge à "0 auto", la couleur de fond à gris.
- pour le sélecteur `#nav`, on fixe le padding à 25px et l'alignement du texte à centré
- pour le sélecteur `#nav li`, on change le mode d'affichage à "inline" et la marge avec des valeurs satisfaisantes

3. Ajouter les règles

```
#nav li a {
  padding: 5px 10px;
  font-weight: bold;
  color: rgba(255,255,255,0.7);
}
#nav li a:hover {
  background: rgba(255,255,255,0.15);
}
```

Essayer différentes valeurs pour l'opacité (quatrième paramètre compris entre 0 et 1) Et si le navigateur ne supporte pas `rgba`, que doit-on prévoir ?

4. Ajouter au sélecteur `#nav li a`

```
text-shadow: 1px 1px 1px black;
```

Essayer différentes valeurs.

5. Ajouter au sélecteur `#nav li a`

```
-webkit-border-radius : 14px;
-moz-border-radius : 14px;
-o-border-radius : 14px;
border-radius : 14px;
```

Essayer différentes valeurs.

6. Ajouter au sélecteur `#nav li a`

```
-webkit-transition: all 0.3s ease-in-out;
-moz-transition: all 0.3s ease-in-out;
-o-transition: all 0.3s ease-in-out;
transition: all 0.3s ease-in-out;
```

Essayer différentes valeurs.

7. Pourquoi y-a-t-il plusieurs propriétés de transition ? Que signifie `all` ? Y-a-t-il moyen de remplacer `all` par quelque chose d'autre ici ?

8. Ajouter au fichier `menu.css` 2 règles telles que :

- pour le sélecteur `#footer`, on fixe le padding à 25px, l'alignement du texte à centré et la couleur à blanche.
- pour le sélecteur `#footer li`, on change le mode d'affichage à "inline" et la marge avec des valeurs satisfaisantes

9. Ajouter le sélecteur

```
#footer a img {
  opacity:0.25;
}
```

Essayer différentes valeurs.

10. Ajouter la règle :

```
#footer a:hover img {
    opacity:0.75;
}
```

Tester et noter l'intérêt de la propriété : nous n'avons besoin que d'un seul jeu d'images

11. Pour avoir un fondu, ajouter ce qui suit au sélecteur `#footer a img`

```
-webkit-transition: all 0.5s ease-in-out;
-moz-transition: all 0.5s ease-in-out;
-o-transition: all 0.5s ease-in-out;
transition: all 0.5s ease-in-out;
```

12. Maintenant, essayons de jouer avec les ombres en ajoutant au sélecteur `#footer a:hover img` ce qui suit :

```
-webkit-box-shadow:4px 4px 10px black;
-moz-box-shadow:4px 4px 10px black);
-o-box-shadow:4px 4px 10px black;
box-shadow:4px 4px 10px black;
```

13. Et maintenant ajouter au sélecteur `#footer a:hover img` ce qui suit :

```
-webkit-transform:scale(1.5);
-moz-transform:scale(1.5);
-o-transform:scale(1.5);
transform:scale(1.5);
```

Tester. Le changement d'échelle est judicieux si la qualité de l'image affichée ne s'en ressent pas. Avez-vous noté l'intérêt ici : un seul jeu d'images est nécessaire.

14. Modifier le code précédent en ajoutant une instruction de rotation comme par exemple :

```
-webkit-transform:scale(1.5) rotate(-10deg) ;
-moz-transform:scale(1.5) rotate(-10deg) ;
-o-transform:scale(1.5) rotate(-10deg);
transform:scale(1.5) rotate(-10deg);
```

Tester. Essayer également de jouer avec `skew` et `translate`

15. Maintenant, essayer d'ajouter au sélecteur `body` ceci :

```
background:
    url(img/stars-1.png) repeat fixed -130% 0,
    url(img/stars-2.png) repeat-x fixed 40% 0,
    url(img/space-bg.png) repeat-x fixed -80% 0,
    url(img/clouds.png) repeat-x fixed 100% 0;
```

Il s'agit ici de 4 images (PNG semi-transparentes) qui vont se superposer. La première image est celle la plus proche de l'utilisateur. Essayer différentes combinaisons d'images en arrière-plan (par exemple, en n'utilisant que les deux premières ci-dessus).

16. On ajoute les 3 règles suivantes par rapport au formulaire :

```
#idForm {
    width: 300px;
    margin: 0 auto;
}
#idForm input {
    margin: 0 0 10px 0;
}
#idForm label {
```

```

display: block;
font-weight: bold;
line-height: 1.4;
color: #666;
color: rgba(0,0,0,.6);
text-shadow: 0 1px 1px #fff;
}

```

17. On ne souhaite pas cette marge de 10px après le dernier élément `input`. Aussi ajoute-t-on ceci en utilisant à la nouvelle pseudo-classe `:last-child`

```

#idForm input:last-child {
    margin: 0px;
}

```

18. Ajouter la règle suivante et déterminez le rôle de chaque propriété.

```

#idForm input[type="text"] {
    width: 215px;
    padding: 5px 8px;
    font-family: Helvetica, sans-serif;
    font-size: 1.2em;
    color: #666;
    border: none;
    background-color: #fff;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    -o-border-radius: 5px;
    border-radius: 5px;
}

```

19. Pour jouer avec des dégradés, ajouter à la dernière règle :

```

background-image: -webkit-gradient(linear, 0% 0%, 0% 12%, from(#999), to(#fff));
background-image: -moz-linear-gradient(0% 12% 90deg, #fff, #999);

```

20. Pour affiner le rendu du bouton, on ajoute la règle suivante :

```

#idForm input[type="submit"] {
    padding: 8px 15px;
    font-family: Helvetica, Arial, sans-serif;
    font-weight: bold;
    line-height: 1;
    color: #444;
    border: none;
    background-color: #fff;
    -webkit-border-radius: 23px;
    -moz-border-radius: 23px;
    -o-border-radius: 23px;
    border-radius: 23px;
    text-shadow: 0 1px 1px rgba(255,255,255,.85);
    background-image: -webkit-gradient(linear, 0% 0%, 0% 100%, from(#fff), to(#bbb));
    background-image: -moz-linear-gradient(0% 100% 90deg, #bbb, #fff);
    -webkit-box-shadow: 0 1px 2px rgba(0,0,0,.5);
    -moz-box-shadow: 0 1px 2px rgba(0,0,0,.5);
    -o-box-shadow: 0 1px 2px rgba(0,0,0,.5);
}

```

```
    box-shadow: 0 1px 2px rgba(0,0,0,.5);  
}
```

Déterminer le rôle de chaque propriété.

21. Pour finir, un petit effet lorsque qu'un élément prend le focus :

```
#idForm input[type="text"]:focus {  
  -webkit-box-shadow: 0 0 12px rgba(51,204,255,0.5);  
  -moz-box-shadow: 0 0 12px rgba(51,204,255,0.5);  
  -o-box-shadow: 0 0 12px rgba(51,204,255,0.5);  
  box-shadow: 0 0 12px rgba(51,204,255,0.5);  
}
```