

LYDIE TREMBLET

Initiation au développement web

Isfates SG-LOG 2013

Mise en forme d'un site Internet

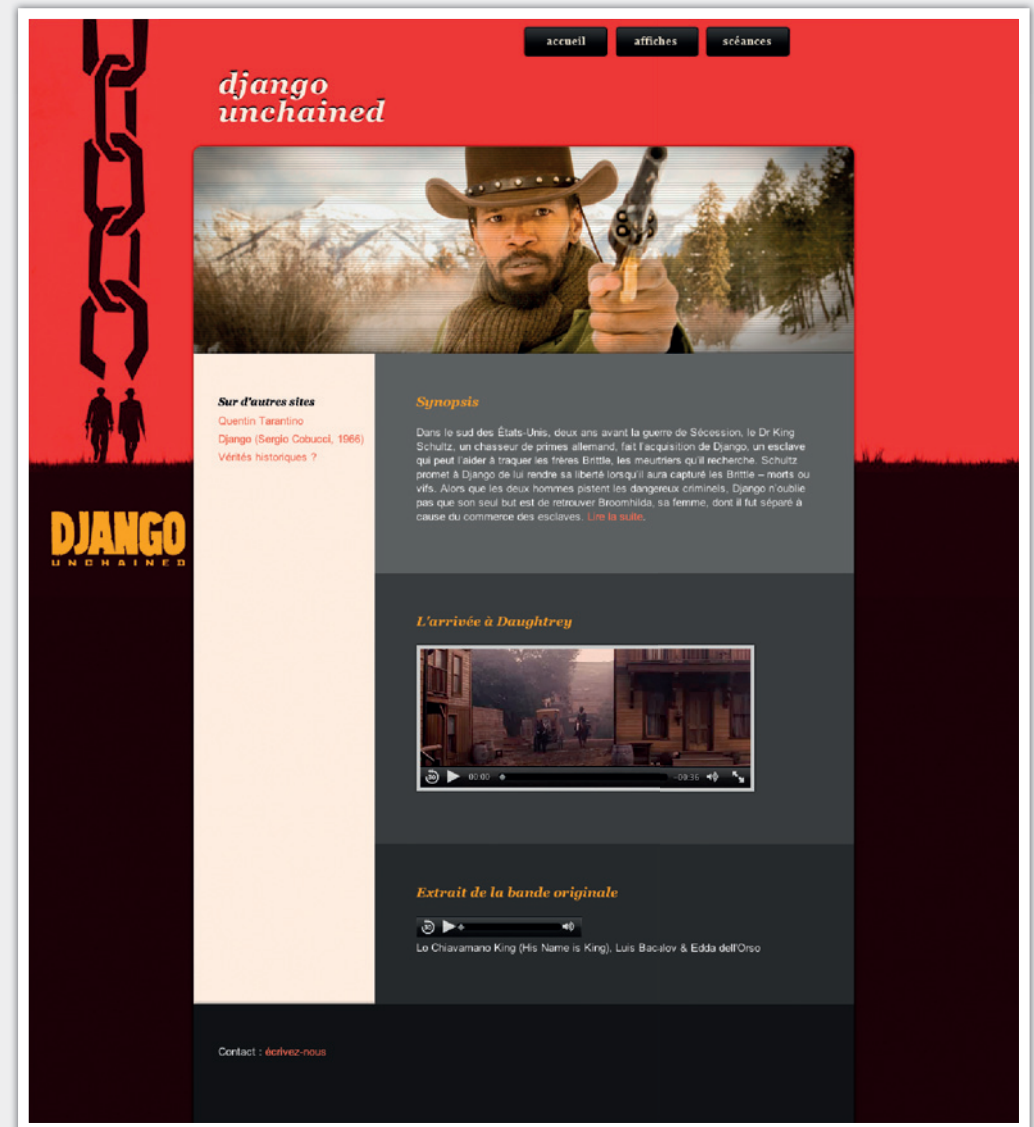
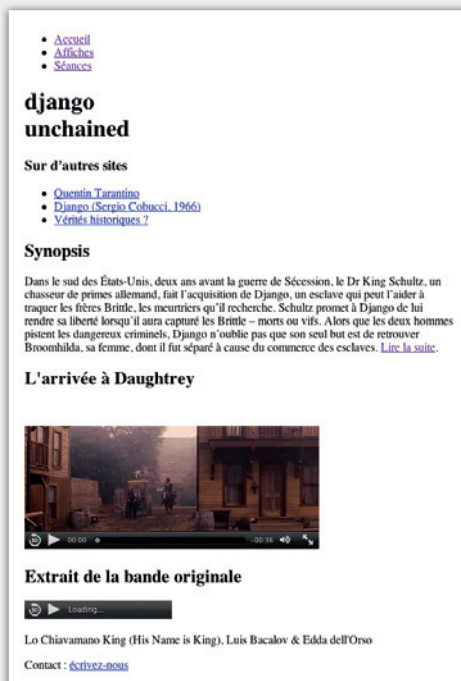
HTML Blocs conteneurs

CSS Le positionnement

CSS Quelques commandes CSS3

+ Notions de design graphique

Pour aborder les points traités dans ce cours magistral, nous allons travailler à partir d'un exemple de page web, que nous mettrons en forme.



1. Blocs conteneurs

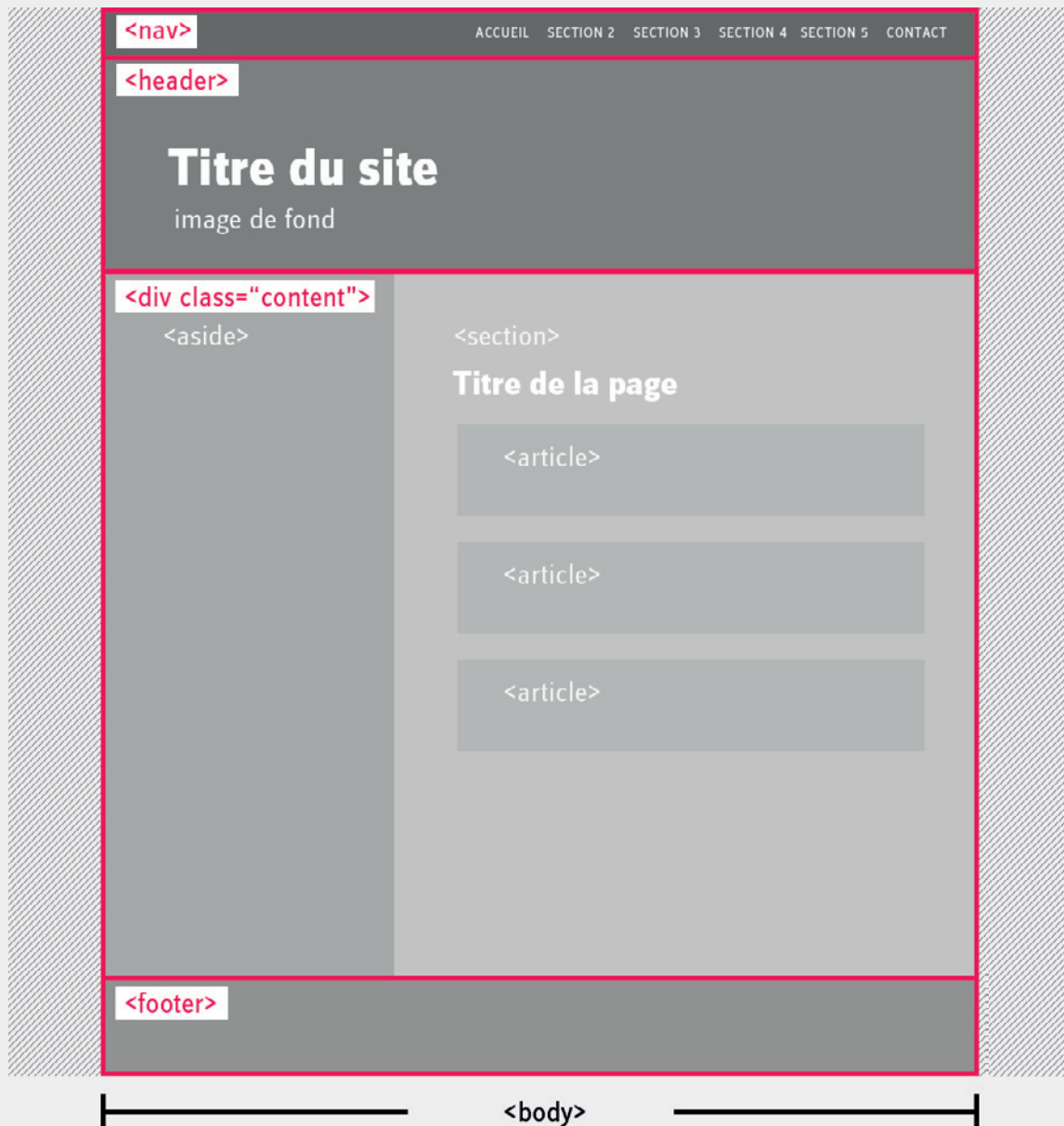
Nous avons vu précédemment des documents HTML composés de blocs (éléments) disposés les uns au-dessus des autres. Nous cherchons à organiser notre page pour créer des colonnes, des boutons de navigation ou encore des paragraphes bien distincts.

Pour cela, nous allons utiliser des **blocs conteneurs**, qui nous permettent de regrouper les informations de même nature.

Nous pourrons alors construire notre page de manière logique, puis y appliquer un design graphique.

a. Balises

<code><body></code>	<i>Body</i>	Corps de la page
<code><header></code>	<i>Header</i>	En-tête de page
<code><nav></code>	<i>Navigation</i>	Menu
<code><section></code>	<i>Section</i>	Chapitre, section
<code><article></code>	<i>Article</i>	Paragraphe, article
<code><aside></code>	<i>Aside</i>	Contenu en rapport indirect
<code><footer></code>	<i>Footer</i>	Pied de page
<code><div></code>	<i>Division</i>	Conteneur polyvalent



On retrouve les blocs conteneurs dans les pages de sites Internet, comme sur les gabarits proposés.

Pour rendre plus lisible la structure des documents, on utilise les **commentaires**. Ceux-ci ne sont pas visibles dans les navigateurs. Les commentaires servent aussi à rendre inactives des lignes de code.

```
<!-- header de la page -->
```

Commentaires CSS :

```
/* header */
```



Créer les blocs conteneurs / commenter

b. Dimensions

Nous exprimerons les dimensions des blocs en **pixels** (px). D'autres unités existent, telles que les pourcentages.



```
div {  
  width: 50px;  
  height: 50px;  
}
```



```
div {  
  width: 100px;  
  height: 200px;  
}
```



```
div {  
  width: 200px;  
  height: 150px;  
}
```



Ajouter les dimensions aux blocs conteneurs concernés
Auparavant : éliminer les marges internes et externes
par défaut

2. Juxtaposer les blocs

a. La propriété float

Pour juxtaposer des blocs, nous utiliserons principalement la technique des éléments flottants et la commande **float**. On peut aligner à gauche ou à droite.

- Aligner une image imbriquée dans un bloc de texte :



Aliquam lectus orci,
adipiscing et, sodales
ac, feugiat non, lacus.
Ut dictum velit nec est.
Quisque posuere, purus
sit amet malesuada
blandit, sapien sapien
auctor arcu, sed pulvi-
nar felis mi sollicitudin
tortor. Maecenas volut-
pat, nisl et dignissim
pharetra, urna lectus

ultrices est, vel pretium pede turpis id velit.
Aliquam sagittis magna in felis egestas rutrum.

```
<div>  
  ...  
</div>
```

```
img {  
  float: left;  
}
```

```
img {  
  float: right;  
}
```

Aliquam lectus orci,
adipiscing et, sodales
ac, feugiat non, lacus.

Ut dictum velit nec
est. Quisque posuere,
purus sit amet male-
suada blandit, sapien
sapien auctor arcu, sed
pulvinar felis mi sollici-
tudin tortor. Maecenas
volutpat, nisl et dignis-
sim pharetra, urna lec-

tus ultrices est, vel pretium pede turpis id velit.
Aliquam sagittis magna in felis egestas rutrum.



- Pour juxtaposer deux ou plusieurs blocs distincts, il est indispensable de spécifier leur largeur.

Aliquam
lectus orci,
adipiscing et,
sodales ac,
feugiat
non, lacus.
Ut dictum velit
nec est.

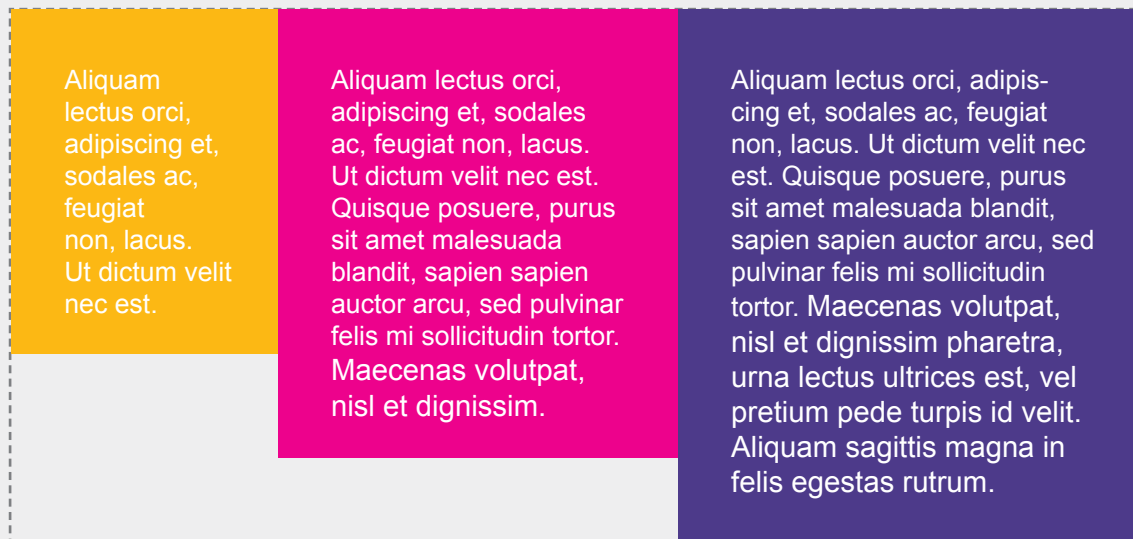
Aliquam lectus orci,
adipiscing et, sodales
ac, feugiat non, lacus.
Ut dictum velit nec est.
Quisque posuere, purus
sit amet malesuada
blandit, sapien sapien
auctor arcu, sed pulvinar
felis mi sollicitudin tortor.
Maecenas volutpat,
nisl et dignissim.

Aliquam lectus orci, adipis-
cing et, sodales ac, feugiat
non, lacus. Ut dictum velit nec
est. Quisque posuere, purus
sit amet malesuada blandit,
sapien sapien auctor arcu, sed
pulvinar felis mi sollicitudin
tortor. Maecenas volutpat,
nisl et dignissim pharetra,
urna lectus ultrices est, vel
pretium pede turpis id velit.
Aliquam sagittis magna in
felis egestas rutrum.

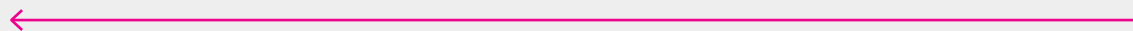
```
<div class="orange">...</div>  
<div class="pink">...</div>  
<div class="violet">...</div>
```

```
←  
div {  
  float: left;  
}  
  
.orange {  
  width: 100px;  
}  
  
.pink {  
  width: 150px;  
}  
  
.violet {  
  width: 175px;  
}
```

- Lorsque ces blocs sont imbriqués dans un bloc conteneur, le total de leurs largeurs additionnées ne doit pas être supérieur à la largeur du bloc parent.



```
.orange {  
  width: 100px;  
}  
.pink {  
  width: 150px;  
}  
.violet {  
  width: 175px;  
}
```



```
.main-content { width: 425px; }
```

```
<div class="main-content">  
  <div class="orange">...</div>  
  <div class="pink">...</div>  
  <div class="violet">...</div>  
</div>
```



```
.orange {  
  width: 100px;  
}
```

```
.pink {  
  width: 150px;  
}
```

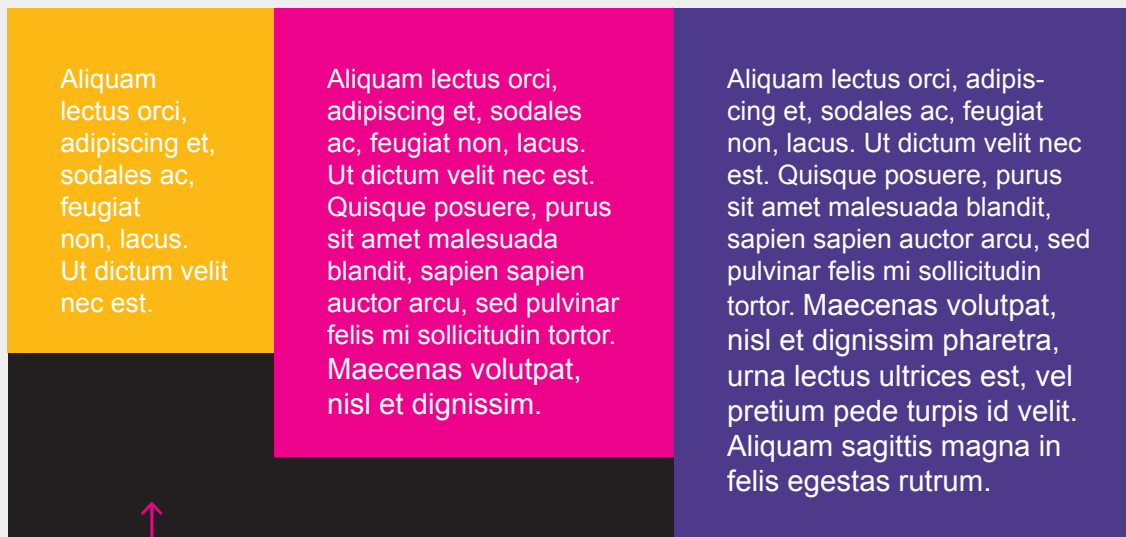
```
.violet {  
  width: 200px;  
}
```

total
450px

450px > 425px
Le dernier bloc
est déplacé en-dessous.

```
.main-content { width: 425px; }
```

- Pour être étendu sur toute la hauteur des blocs enfants, le bloc parent doit comporter la propriété **overflow: auto**. Cela permet notamment de voir le fond de couleur du bloc parent et de faire en sorte que les éléments suivants dans la page se disposent correctement.

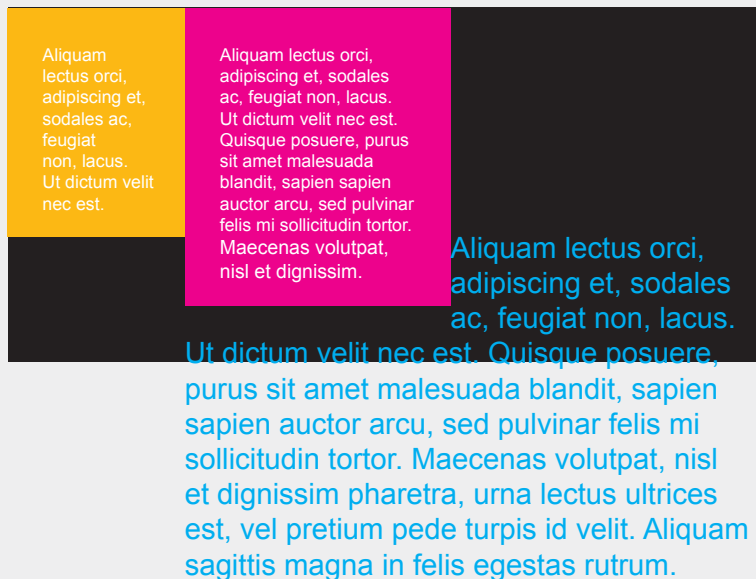


Le fond du bloc parent apparaît
grâce à la propriété **overflow: auto**.

```
<div class="main-content">  
  <div class="orange">...</div>  
  <div class="pink">...</div>  
  <div class="violet">...</div>  
</div>
```

```
main-content {  
  overflow: auto;  
  background-color: black;  
}
```

- Lorsque l'on a fini de juxtaposer des blocs dans une partie du site, on applique la propriété **clear: both** au prochain bloc de la page. Sinon, les éléments continuent d'être « aimantés ».

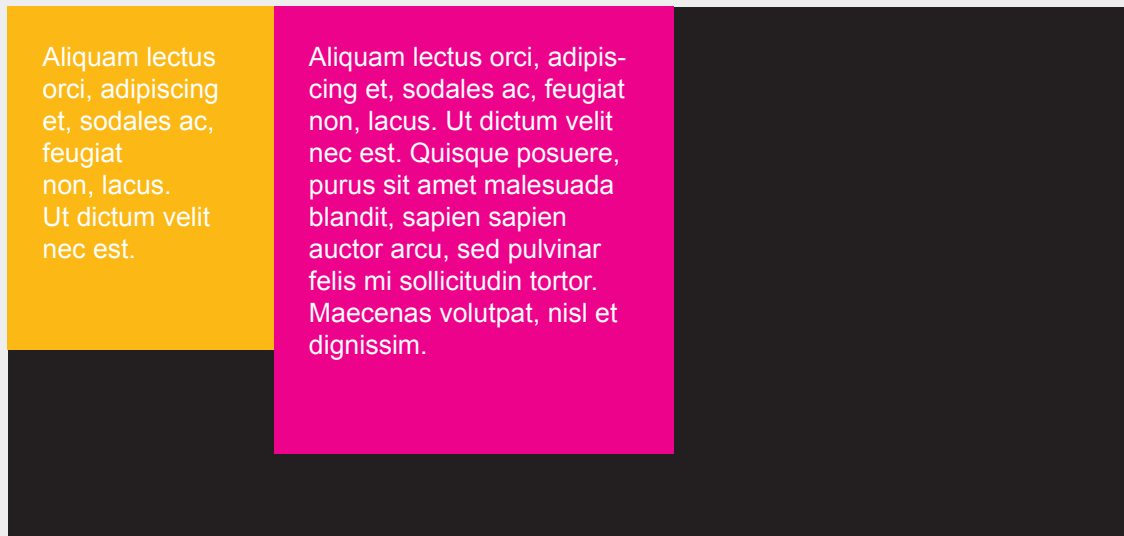


```
<div class="main-content">  
  <div class="orange">...</div>  
  <div class="pink">...</div>  
</div>
```

```
<div class="conclusion">...</div>
```

```
.main-content {  
  background-color: black;  
}
```

```
.orange, .pink {  
  float: left;  
}
```



Aliquam lectus orci, adipiscing et, sodales ac, feugiat non, lacus. Ut dictum velit nec est. Quisque posuere, purus sit amet malesuada blandit, sapien sapien auctor arcu, sed pulvinar felis mi sollicitudin tortor. Maecenas volutpat, nisl et dignissim pharetra, urna lectus ultrices est, vel pretium pede turpis id velit. Aliquam sagittis magna in felis egestas rutrum.

↑
L'élément suivant ainsi que le reste de la page
apparaissent correctement avec **clear: both**.

```
.main-content {  
  overflow: auto;  
}
```

```
.orange, .pink {  
  float: left;  
}
```

```
.conclusion {  
  clear: both;  
}
```



b. Menu de navigation horizontal

- D'une part, nous devons appliquer la propriété **float** aux éléments `<li>` pour les juxtaposer. On définit aussi les **marges externes** pour espacer les boutons.
- Par ailleurs, toutes les **options de dimensionnement** et de **fond** sont appliquées aux liens `<a>` imbriqués dans les balises `<li>` afin d'avoir un bouton entièrement clicable.
- Pour pouvoir appliquer une largeur et une hauteur à un lien `<a>`, il faut auparavant lui appliquer la propriété **display: block**.

accueil

vidéos

réserver



```
<ul class="menu">
  <li><a href="index.html">accueil</a></div>
  <li><a href="videos.html">vidéos</a></div>
  <li><a href="reserver.html">réserver</a></div>
</ul>
```

```
.menu li {
  float: left;
}
```

```
.menu li a {
  display: block;
  width: 100px;
  height: 35px;
  line-height: 35px;
  color: white;
  background-color: deepskyblue;
}
```

```
.menu li a:hover {
  color: lightskyblue;
  background-color: darkslateblue;
}
```



Terminer le menu

3. Quelques commandes CSS3

a. Les ombres de blocs › **box-shadow**

Nulla dui purus, eleifend vel,
consequat non, dictum porta, nulla.
Duis ante mi, laoreet ut, commodo
eleifend, cursus nec, lorem. Aenean
eu est. Etiam imperdiet turpis.
Praesent nec augue.

```
div {  
  width: 200px;  
  height: 100px;  
  background-color: white;  
  box-shadow: 2px 2px 5px lightblue;  
}
```

Décalage horizontal

Décalage vertical

Taille

Couleur

Les décalages verticaux et horizontaux peuvent prendre des valeurs négatives.



Appliquer ombres portées aux blocs concernés

b. Les ombres de textes › **text-shadow**



Décalage horizontal Décalage vertical Couleur

Taille

```
text-shadow: 0px -1px 1px black; › creusé
```

```
text-shadow: 0px 1px 1px black; › relief
```

```
text-shadow: 0px 0px 5px deeppink; › halo
```



Appliquer l'ombre de texte au main-title

c. Les coins arrondis › border-radius

Aliquam lectus
orci, adipiscing
et, sodales ac,
feugiat non,
lacus. Ut dictum
velit nec est.

```
border-radius: 0px 15px 0px 15px;
```

↑ ↑ ↑ ↑
haut haut bas bas
gauche droite droite gauche

Aliquam lectus
orci, adipiscing
et, sodales ac,
feugiat non,
lacus. Ut dictum
velit nec est.

```
border-radius: 0px;
```

↑
le même arrondi aux
quatre angles



```
width: 100px;  
height: 100px;  
border-radius: 50px;
```

↑
border-radius égal à la moitié
du côté d'un carré :
dessine un cercle



Appliquer les coins arrondis au menu et au header

d. Les dégradés › linear-gradient

valider

```
background-image: linear-gradient(yellowgreen, forestgreen);
```

↑
couleur
du haut

↑
couleur
du bas



Appliquer les dégradés au menu

e. Les transitions › transition

Les transitions donnent entre autres la possibilité d'avoir un effet de fondu entre les changements de couleurs lors d'un survol.

On applique la propriété transition à l'état initial de l'élément, pas à son état survolé (:hover).

```
a {  
  background-color: gray;  
  transition: background-color 0.2s;  
}
```

↑ ↑
variable durée
concernée

```
a:hover {  
  background-color: black;  
}
```

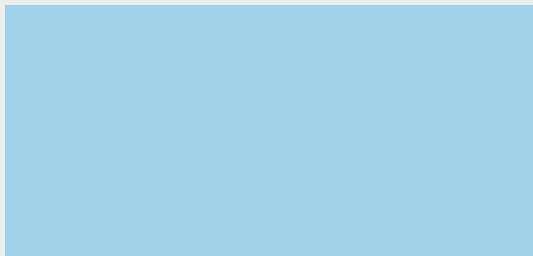
Nous avons terminé de faire le tour des éléments fondamentaux permettant de composer quelques pages web, tant sur le plan du sens que du design.

Attention cependant : en fonction des navigateurs, toutes les commandes ne sont pas prises en compte, notamment pour ce qui concerne les CSS3. Il existe parfois des alternatives dans le langage CSS mais, parfois, il faut recourir à des hacks, c'est-à-dire des scripts permettant de recréer certains effets avec d'autres langages (JavaScript, par ex.)

4. Notions de design graphique

a. Choix des images

Les images doivent être fortes et leur **résolution** doit être suffisante. Les images choisies doivent avoir des dimensions aux moins égales au bloc conteneur, jamais inférieures.



width: 200px;



largeur de l'image : 200px



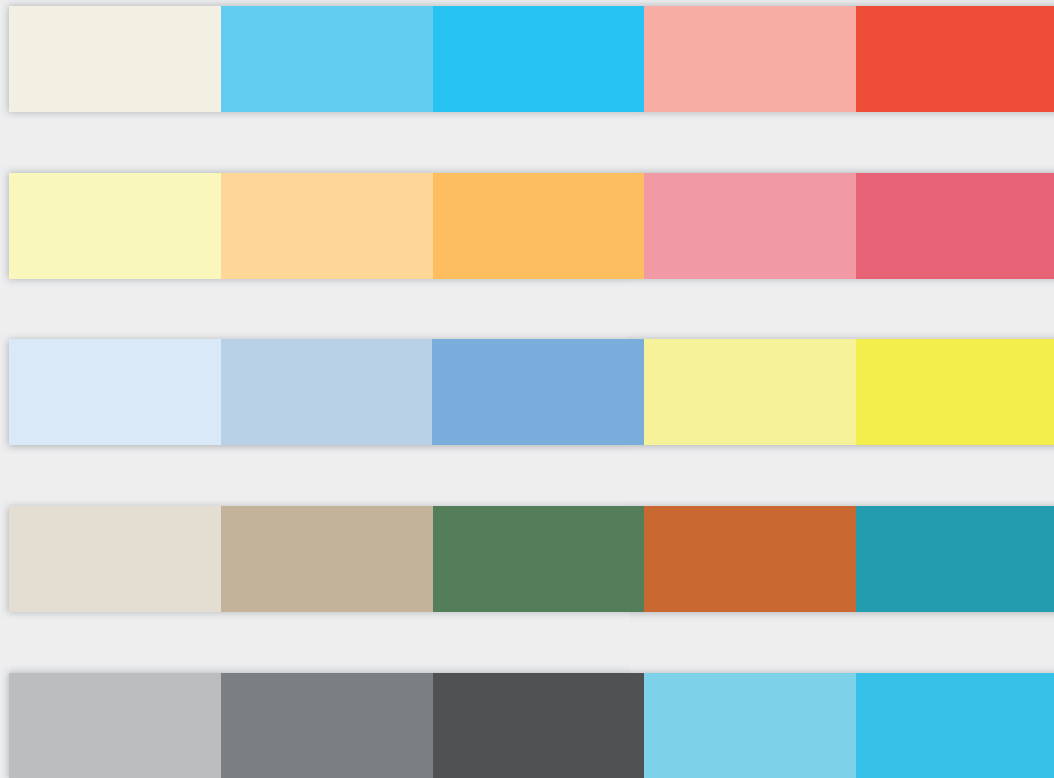
largeur de l'image : 130px

Pour recadrer une image trop grande,
aller sur le site :

www.photofuntoos.com/crop-photo.php

b. Palettes de couleurs

Définir des gammes de couleurs qui s'associent bien. Ne pas multiplier le nombre de couleurs différentes.



c. Textes

L'apparence visuelle des textes est fondamentale dans le design graphique d'un site Internet.

Ils doivent rendre visible la **hiérarchie de l'information** par le bon emploi des tailles, graisses et couleurs.

Par ailleurs, ils doivent rester **lisibles**. On évitera donc les faibles contrastes texte/fond ou encore d'associer deux couleurs très vives pour le texte et son fond.

Enfin, pour agrémenter le rendu visuel, on peut utiliser des **propriétés CSS3** (text-shadow, placement dans un bloc aux coins arrondis, etc.).

Cependant, éviter la surcharge d'effets.

Nulla dui purus, eleifend vel, consequat non, dictum porta, nulla. Duis ante mi, laoreet ut, commodo eleifend, cursus nec, lorem. Aenean eu est. Etiam imperdiet turpis. Praesent nec augue. Curabitur ligula quam, rutrum id, tempor sed, consequat ac, dui. Vestibulum accumsan eros nec magna. Vestibulum vitae dui. Vestibulum nec ligula et lorem consequat ullamcorper.

Nulla dui purus, eleifend vel, consequat non, dictum porta, nulla. Duis ante mi, laoreet ut, commodo eleifend, cursus nec, lorem. Aenean eu est. Etiam imperdiet turpis. Praesent nec augue. Curabitur ligula quam, rutrum id, tempor sed, consequat ac, dui. Vestibulum accumsan eros nec magna. Vestibulum vitae dui. Vestibulum nec ligula et lorem consequat ullamcorper.

ok

~~Live~~

Live

ok

Live

ok