CSS : Notions de base.

par Pierre-Baptiste Naigeon (Tutos, tests et articles web)

Date de publication : 27 Avril 2006

Dernière mise à jour : 02 Mai 2006

Il est très important d'intégrer les bases en CSS, afin d'éviter les écueils par la suite. A la fin de cet article, vous saurez quels sont les types d'éléments, leurs différences, ce qu'est le flux, comment le modifier, et ce qu'est l'héritage en CSS.

Introduction
I - Bloc, en-ligne : les types d'éléments.
II - Le flux, kezako ?
III - Positionnement d'un élement
    III-A - Propriété CSS "position"
    III-B - Propriété CSS "float"
IV - Propriété des éléments : display
V - Et l'héritage, qu'est-ce que c'est ?
VI - Petit complément rapide : le z-index.
VII - Remerciements et liens annexes

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

Introduction

Dans tous les exemples donnés ici, nous utiliserons deux fichiers : 'index.html', qui va contenir le code HTML, et 'style.css', qui va contenir toutes les définitions de style.

Le fichier 'style.css' est inclu comme suit dans notre fichier HTML (entre les balises <head> et </head> :

```
<link rel="stylesheet" type="text/css" href="styles.css">
```


Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

I - Bloc, en-ligne : les types d'éléments.

En CSS, il existe deux grandes familles d'éléments : les éléments de type "bloc", et les éléments de style "en-ligne".

Grosso modo, les éléments de type "bloc" (DIV, UL, FORM, BLOCKQUOTE, PRE, ...), servent à la mise en page générale de la page en créant de grands rectangles 'conteneurs'. Ils peuvent contenir soit d'autres éléments de type "bloc", soit des éléments de type "en-ligne". Les éléments de type "en-ligne" (A, B, FONT, IMG, INPUT, U, SPAN, ...) sont juste des conteneurs plus petits que bloc, avec un comportement spécifique. IMG et INPUT sont des éléments vides d'après la DTD.

D'une manière générale, évitez d'utiliser les balises HTML pour la mise en forme du contenu, dans l'idée de séparer au maximum le fond le la forme (le tag U en HTML peut être avantageusement remplacé par text-decoration:underline; en CSS).

De plus, ces balises ont des marges externes (**margin**) et internes (**padding**) définies par défaut à 0 et non re-définissables.

Par défaut, les éléments de type "bloc" vont prendre toute la largeur de la page.

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

II - Le flux, kezako ?

Le flux, c'est l'ordre d'affichage des éléments.

Ainsi, le navigateur parcours votre page HTML, récupère les différents éléments, et les affiche dans un certain ordre.

Les éléments sont affichés selon leur ordre d'apparition dans le code source.

Les éléments de type "bloc" vont s'afficher les uns en dessous des autres, alors que des éléments de type "en-ligne" vont s'afficher les uns à la suite des autres.

Exemple :

CSS commun aux deux type d'affichage

```
#element1 {
 background-color:#00CCFF; /* bleu ciel */
 }
#element2 {
 background-color:#FF6666; /* rouge */
 }
#element3 {
 background-color:#6699FF; /* mauve */
 }
```

Affichage de type 'bloc' : HTML

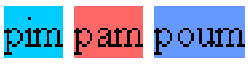
```
<div id="element1">riri</div>
<div id="element2">fifi</div>
<div id="element3">loulou</div>
```

Affichage de type bloc

Télécharger les codes-source de l'exemple

Affichage de type 'en-ligne' : HTML

```
<span id="element1">pim</span>
<span id="element2">pam</span>
<span id="element3">poum</span>
```



Affichage de type en-ligne

Télécharger les codes-source de l'exemple

Voilà pour le flux tel qu'il est normalement interprété. Mais il est possible à l'aide des propriétés CSS de le modifier.

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

Voyons donc comment...

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

III - Positionnement d'un élement

III-A - Propriété CSS "position"

La propriété CSS **position** permet de modifier le comportement d'affichage des blocs :)

Evacuons de suite **position:static**, qui se contente de dire que l'élément doit se comporter normalement. Il suivra donc le flux sans se poser plus de questions.

position:inherit (CSS2) déclare que l'élément en question hérite de la propriété **position** de son parent. Si la position du parent est de type absolue, l'élément concerné le sera aussi...

position:fixed pourrait être une propriété interessante. L'idée est de placer un élément sur la page, et qu'il y reste quel que soit le défilement. Je vous laisse imaginer l'utilité pour un menu par exemple...

Seul problème, certains navigateurs non conformes aux standards CSS2 ne prennent pas en compte cette propriété, il vous va donc falloir tricher (je vous renvoie donc à la **FAQ CSS**).

position:relative ne retire pas l'élément du flux, mais le déplace simplement par rapport à sa position dans le flux.

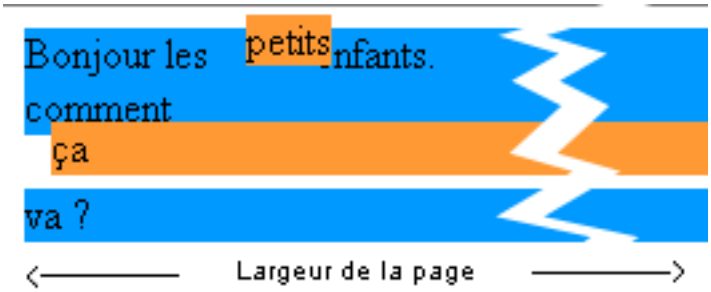
Petit exemple pour mieux comprendre :

Positionnement relatif : CSS

```
.normal {
 background-color:#0099FF; /* bleu ciel */
 }
.posrelative {
 position:relative;
 background-color:#FF9933; /* orange */
 bottom:5px;
 left:10px;
 }
```

Positionnement relatif : HTML

```
<div class="normal">Bonjour les <span class="posrelative">petits</span> enfants.</div>
<div class="normal">comment</div>
<div class="posrelative">&ccedil;a</div>
<div class="normal">va ?</div>
```



Exemple de positionnement relatif

Télécharger les codes-source de l'exemple

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

Vous vous rendez compte ici que le troisième div (le orange) est décalé de 10 pixels vers la droite et de 5 vers le haut par rapport à la position qu'il devrait occuper normalement. Idem pour le span orange, décalé de la même chose par rapport à sa position théorique.

position:absolute retire complètement l'élément concerné du flux, et le place aux coordonées définies par les propriétés **bottom**, **left**, **right** ou **top**.

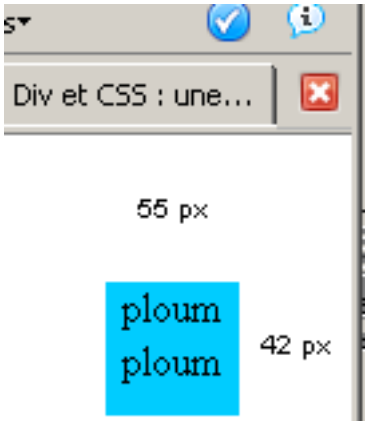
Dans l'exemple, nous allons dimensionner notre DIV pour mieux se rendre compte de l'effet produit :

Positionnement absolu : CSS

```
#element1 {
 position:absolute;
 background-color:#00CCFF; /* bleu ciel */
 width:50px;
 height:50px;
 text-align:center;
 top:55px;
 right:42px;
 }
```

Positionnement absolu : HTML

```
<div id="element1">ploum ploum</div>
```



Exemple de positionnement absolu

Télécharger les codes-source de l'exemple

*Attention cependant... La position n'est absolue que par rapport au conteneur... Dans l'exemple précédent, le conteneur était le **BODY**, donc la fenêtre du navigateur.*

Positionnement absolu 'relatif' : CSS

```
#element1 {
    position:absolute;
    background-color:#00CCFF; /* bleu ciel */
    width:50px;
    height:50px;
    text-align:center;
    top:55px;
    left:42px;
    }
#element2 {
```

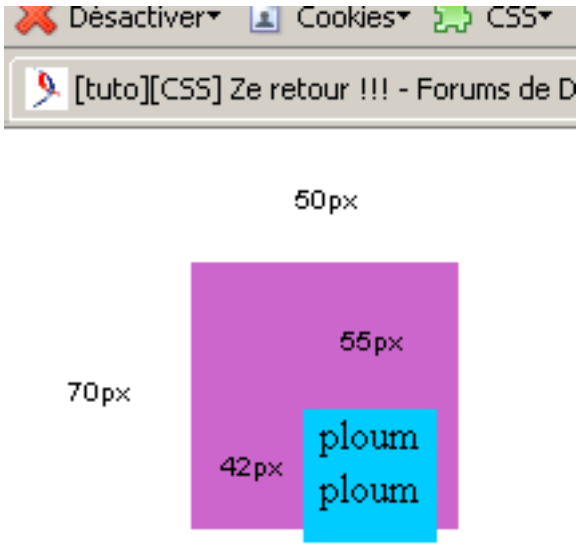

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

Positionnement absolu 'relatif' : CSS

```
        position:absolute;
        background-color:#CC66CC; /* violet */
        width:100px;
        height:100px;
        text-align:center;
        top:50px;
        left:70px;
        }
```

Positionnement absolu 'relatif' : HTML

```
<div id="element2">
    <div id="element1">ploum ploum</div>
</div>
```



Exemple de positionnement absolu 'relatif'

Télécharger les codes-source de l'exemple

Pour gérer la superposition des différents blocs, reportez-vous à la **section VI. Petit complément rapide : le z-index.**.

III-B - Propriété CSS "float"

Voyons maintenant la propriété **float**, celle qui vous servira le plus lors de vos mises en page.

Cette propriété positionne l'élément le plus à gauche (**float:left;**) ou le plus à droite (**float:right;**) de son conteneur.

Le reste des éléments du conteneur s'affichera en suivant les contours des éléments flottants.

Positionnement flotant : CSS

```
#conteneur {
 background-color:#CC9999; /* marron bizarre */
 text-align:justify;
 }
#element1 {
```

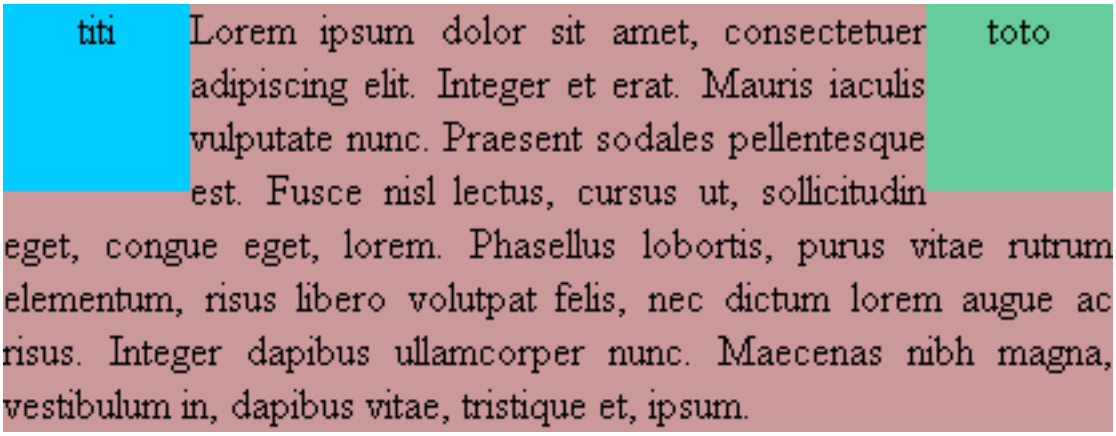

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

Positionnement flotant : CSS

```
    float:left;
    background-color:#00CCFF; /* bleu ciel */
    width:70px;
    height:70px;
    text-align:center;
    }
   #element2 {
    float:right;
    background-color:#66CC99; /* vert */
    width:70px;
    height:70px;
    text-align:center;
    }
```

Positionnement flotant : HTML

```
   <div id="conteneur">
    <div id="element1">titi</div>
    <div id="element2">toto</div>
    Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
    Integer et erat. Mauris iaculis vulputate nunc.
    Praesent sodales pellentesque est.
    Fusce nisl lectus, cursus ut, sollicitudin eget, congue eget, lorem.
    Phasellus lobortis, purus vitae rutrum elementum, risus libero volutpat felis, nec dictum lorem
    augue ac risus.
    Integer dapibus ullamcorper nunc. Maecenas nibh magna, vestibulum in, dapibus vitae, tristique et,
    ipsum.
   </div>
```



Exemple de positionnement flotant

Télécharger les codes-source de l'exemple

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

IV - Propriété des éléments : display

La propriété CSS **display** va nous permettre de modifier le type de nos éléments.

Nous n'allons pas détailler ici toutes les valeurs possibles, mais nous intéresser à deux d'entre elles : **display:block** et **display:inline**.

Dans cet exemple, nous allons créer deux éléments de type "bloc", en leur affectant la propriété display:inline, et deux éléments de type "en-ligne", en leur affectant la propriété display:block.

Propriété display : CSS

```
#bloc1 {
 background-color:#00CCFF; /* bleu ciel */
 display:inline;
 }
#bloc2 {
 background-color:#FF6666; /* rouge */
 display:inline;
 }
#enligne1 {
 background-color:#00CCFF; /* bleu ciel */
 display:block;
 }
#enligne2 {
 background-color:#FF6666; /* rouge */
 display:block;
 }
```

Propriété display : HTML

```
<div id="bloc1">Mickey</div>
<div id="bloc2">Minnie</div>
<span id="enligne1">Donald</span>
<span id="enligne2">Daisy</span>
```

Modification des propriétés des éléments avec 'display'

Télécharger les codes-source de l'exemple

Etrange n'est-ce pas... Les éléments de type "bloc" se comportent maintenant comme des éléments de type "en-ligne", et vice-versa.

Imaginez les possibilités offertes par une telle propriété... Un menu sous forme de liste généré dynamiquement pourrait devenir, sur simple changement du style appliqué, un menu horizontal sans pour autant modifier la fonction de base...

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

V - Et l'héritage, qu'est-ce que c'est ?

Bon, puisque vous avez l'air chaud comme la braise, voici venue l'heure de parler d'héritage, de polymorphisme, et d'encapsulation...

Bon, partez pas en courant, revenez, on va se contenter de l'héritage, le reste n'existe pas en HTML / CSS, et c'est une notion toute simple en plus :)

L'idée, c'est qu'un élément X contenu dans un élément Y va hériter des propriétés de cet élément Y.

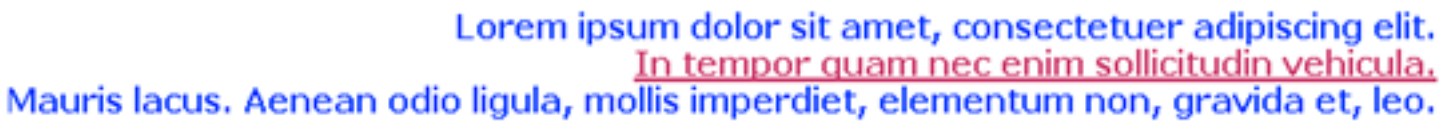
Petit exemple pour mieux comprendre :

Héritage : CSS

```
.niveau1 {
 color:#0033FF; /* bleu */
 text-align:right;
 font-weight:bold;
 font-family:Verdana, Arial, Helvetica, sans-serif;
 }
.niveau2 {
 text-decoration:underline;
 color:#CC3366; /* violet */
 }
```

Héritage : HTML

```
<div class="niveau1">
 Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
 <div class="niveau2">
  In tempor quam nec enim sollicitudin vehicula.
 </div>
 Mauris lacus. Aenean odio ligula, mollis imperdiet, elementum non, gravida et, leo.
</div>
```



Exemple d'héritage

Télécharger les codes-source de l'exemple

Ici, le bloc "niveau2" hérite des propriétés de son père "niveau1". Ainsi, nous avons récupéré l'alignement du texte, le gras, la famille de police et même la couleur.

Nous avons juste choisi de redéfinir la couleur, et de souligner le contenu de l'élément 2, afin de rendre tout ca plus visible.

Bon, pas si compliqué que ça finalement ? Le seul moment où ça se corse, c'est que toutes les propriétés ne peuvent pas être héritées. Ainsi, deux éléments de type "bloc" Y et Z, contenus dans un élément X défini en float se positionneront tout de même l'un au dessus de l'autre

Float conteneur : CSS

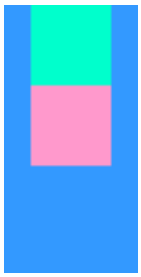
Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

Float conteneur : CSS

```
#conteneur {
 background-color:#3399FF; /* bleu ciel */
 width:50px;
 height:100px;
 }
#contenu1 {
 background-color:#00FFCC; /* vert */
 width:30px;
 height:30px;
 margin-left:auto;
 margin-right:auto;
 }
#contenu2 {
 background-color:#FF99CC; /* rose */
 width:30px;
 height:30px;
 margin-left:auto;
 margin-right:auto;
 }
```

Float conteneur : HTML

```
<div id="conteneur">
 <div id="contenu1"></div>
 <div id="contenu2"></div>
</div>
```



Exemple de float conteneur

Télécharger les codes-source de l'exemple

A noter également, tous les éléments n'ont pas les mêmes propriétés. Ainsi, un élément en-ligne n'ayant pas de marges, il n'héritera donc jamais des marges de son conteneur de type bloc.

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

VI - Petit complément rapide : le z-index.

Dès qu'un des éléments de votre page sort du flux, il peut arriver un effet de superposition.

C'est bien gentil tout ca, mais comment gérer "qui écrase l'autre" ?

La propriété **z-index** est là pour ça. En effet, elle va permettre d'attribuer à chaque élément un 'niveau', celui ayant le plus grand se retrouvant au-dessus.

Petit exemple pour mieux comprendre :

Définissons deux blocs, tous deux positionnés en absolu, et se chevauchant.

Pour chacun d'eux, définissons une propriété z-index différente :

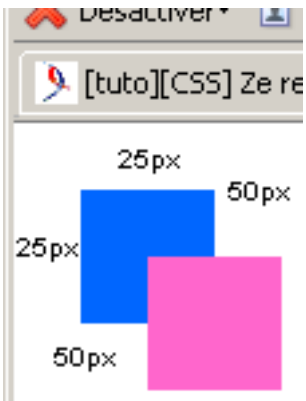
z-index : CSS

```
#element1 {
 background-color:#0066FF; /* bleu */
 position:absolute;
 top:25px;
 left:25px;
 width:50px;
 height:50px;
 z-index:1;
 }
#element2 {
 background-color:#FF66CC; /* rose */
 position:absolute;
 top:50px;
 left:50px;
 width:50px;
 height:50px;
 z-index:2;
 }
```

z-index : HTML

```
<div id="element1"></div>
<div id="element2"></div>
```

Dans ce cas-ci, l'élément2 (le rose) se trouve au dessus de l'élément1 (le bleu).



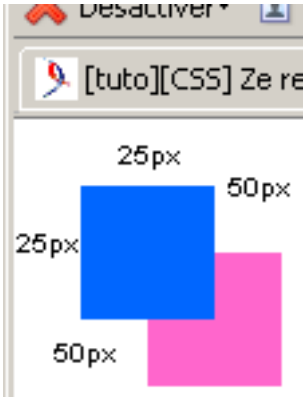
Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

z_index : le rose est au dessus

Intervertissons maintenant les propriétés z-index de nos éléments, sans rien toucher d'autre.

Ainsi, le z-index de 'element1' passe à 2, et celui de 'element2' passe à 1.

Et hop, il n'y a qu'à demander, comme par magie, l'élément1 (en bleu) est passé au dessus de l'élément2 (en rose), sans rien devoir modifier dans le HTML.



z_index : le bleu est au dessus

Télécharger les codes-source de l'exemple

Petite astuce en passant : le z-index peut parfaitement être négatif, ça ne posera de problèmes à personne ;)

Nous n'avons abordé ici qu'une toute petite partie du z-index, qui mériterai à lui seul un article complet. Pour plus de détails, n'hésitez pas à aller lire **Understanding CSS z-index**.

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

VII - Remerciements et liens annexes

Vous avez à présent en main quelques unes des clés pour vous lancer dans la mise en page de vos sites en CSS pur. Adieu TABLE, TD, TR et COLSPAN !!! Le monde est à vous :)

Pour plus d'informations, vous pouvez vous référer aux ***spécifications CSS2 du W3C****, ou à la* ***traduction en français****.*

Et enfin, last but not least, un énorme merci à l'équipe web de developpez.com, et plus particulièrement à **franculo_caoulene** pour ses relectures attentives, ses suggestions ainsi que ses critiques.

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/

Les sources présentés sur cette pages sont libres de droits, et vous pouvez les utiliser à votre convenance. Par contre la page de présentation de ces sources constitue une oeuvre intellectuelle protégée par les droits d'auteurs. Copyright © 2006 - Pierre-Baptiste Naigeon. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts.
http://pbnaigeon.developpez.com/tutoriel/CSS-HTML/notions-de-base/