

CSS I & 2	
Feuilles de style : propriétés, niveaux	
Sélecteurs CSS, balises, classes et identifiants Syntaxe Commentaires css	
Modèles de boîtes Éléments de type «block» et de type «inline»	
Le flottement Sortir du flottement : «clear»	
Le positionnement : statique, relatif, fixe et absolu	
Styler le texte	
Les couleurs en CSS	
Les marges et remplissages	
Les bordures	
Les arrière-plans, couleurs et images	
Les styles de puces des listes Remplacement par une image d'arrière-plan	
Le débordement	
La propriété «display» (affichage)	
Les raccourcis : texte, remplissage, marges	
Les raccourcis : couleurs, bordures et arrières plans	

LES FEUILLES DE STYLE

CSS = feuilles de style en cascade

La page html contient l'information, et non la façon dont l'information est affichée. Pour un unique contenu : plusieurs affichages sont possibles.

Les écrans pour surfer sur le web sont de plus en plus grands et à la fois de plus en plus petits avec l'arrivée des smartphones, des tablettes. Mais aussi, certains internautes ne "regardent" pas l'écran, ils entendent le contenu (pour les aveugles le contenu de la page web est lu), il y a également les impressions papier, ou sur des transparents, impression en braille...

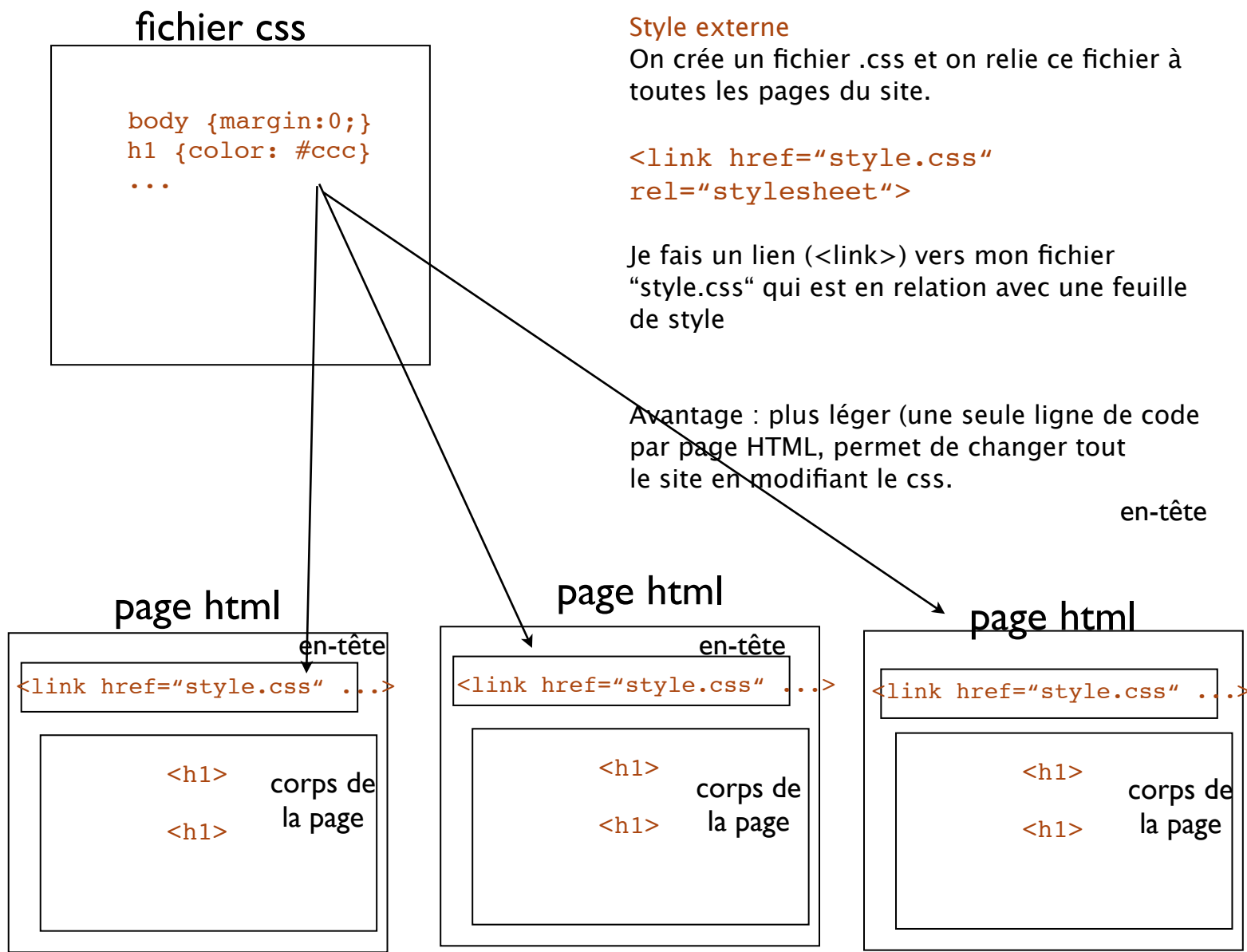
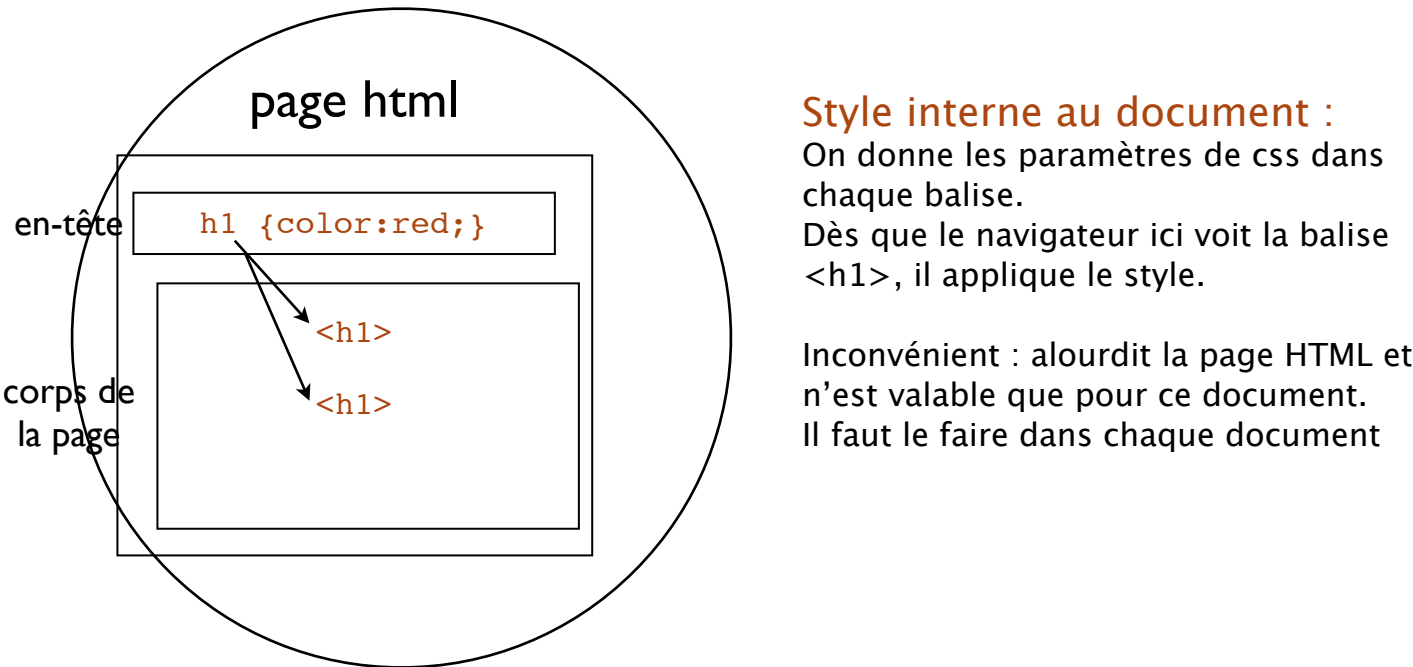
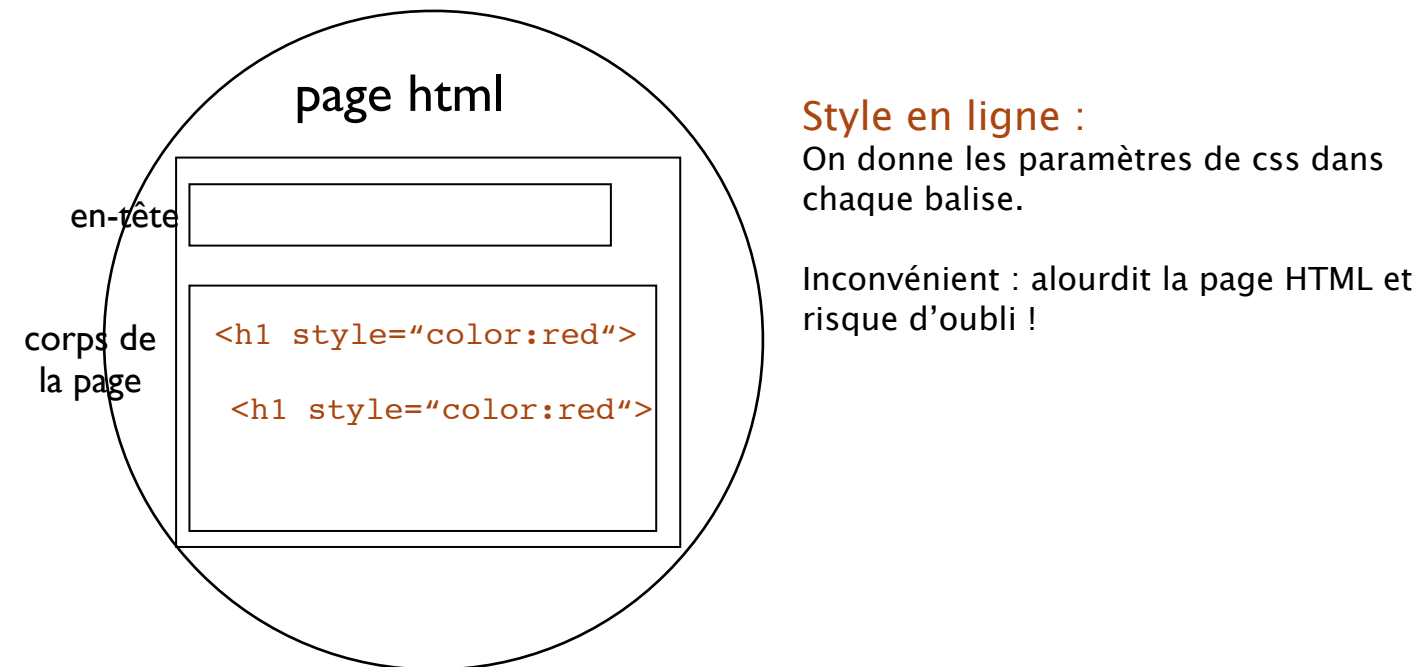
Propriétés

- Police** : taille, style, couleur, gras, italique
- Texte** : alignement, casse, interlignage, espace entre les mots et les lettres.
- Arrière-plans** : couleur ou images.
- Bords et marges** : marges, remplissages, largeur, hauteur.
- Bordures** : style, épaisseurs, couleurs.
- Interface** : puces, indentation, curseur.
- Positionnement** : emplacement exact sur l'écran.
- Affichage et visibilité** : si un élément apparaît et comment.

Le principe de base est le suivant : il faut séparer le contenu de la page, de son apparence.

3 niveaux : en ligne- interne – externe

On privilégie la feuille de style externe, une seule pour tout le site, ce qui facilitera les modifications d'affichage ultérieures.

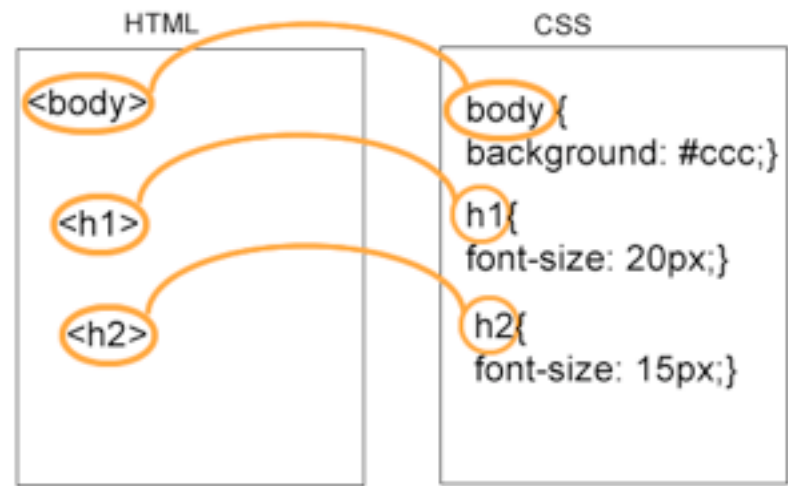


LES SELECTEURS CSS

En sélectionnant un élément de la page HTML, il va être possible de lui attribuer des règles CSS pour changer son aspect, sa position etc

Les sélecteurs de balises

On peut redéfinir les balises de la page HTML.
Par exemple choisir une taille de police pour les balises <h1> qui ont une taille par défaut.



Il est utile de bien analyser les éléments récurrents et les éléments uniques pour déterminer

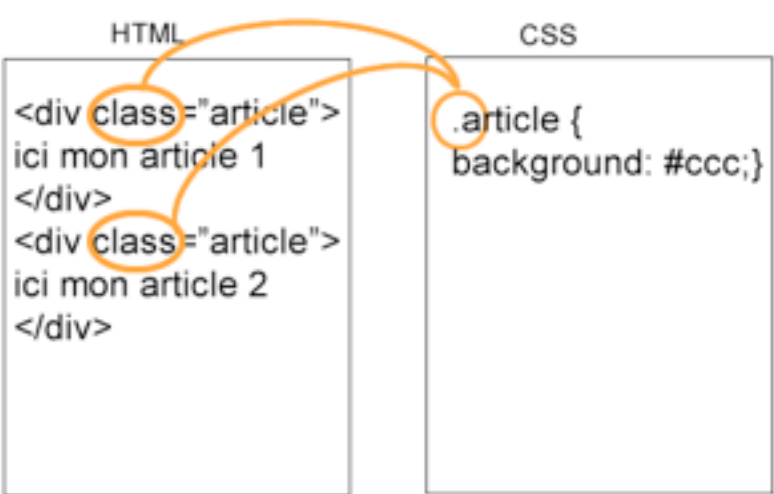
– ce qui sera automatique, (redéfinir les balises),

– ce qui sera ponctuel (appliquer des classes)

– ce qui sera unique (les identifiants)

Les classes personnalisées.

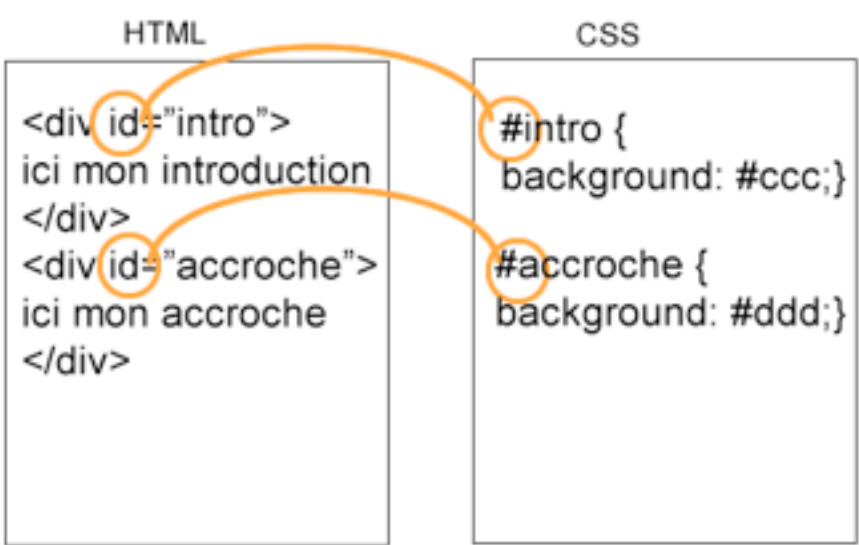
En attribuant une classe personnalisée à une balise, on peut changer aussi son aspect. Il s'agira alors de quelque chose de ponctuellement utilisé dans la page HTML. Il est possible d'utiliser une classe personnalisée autant de fois que l'on souhaite dans la page.
Dans le CSS, on utilise le signe . pour appeler la classe. On parle de sélecteur de classe.



Les ID, ou identifiants.

Ils fonctionnent comme les classes, à la différence près qu'ils ne peuvent être utilisés qu'une seule fois par page. Ils sont uniques. En identifiant une balise dans la page HTML, on peut modifier ses propriétés d'affichage.

On peut identifier des éléments de la page HTML pour les donner des paramètres en CSS.
Un identifiant ne peut être utilisé qu'une seule fois par page.
Dans le CSS, on utilise le signe # pour appeler l'identifiant. On parle de sélecteur d'identifiant



LA SYNTAXE

Le sélecteur, puis des accolades {}
Entre les accolades les paramètres sont séparés par des virgules

Pour un sélecteur de balise :

```
body {
  margin : 0;
  background : #ccc;
}
```

Pour un sélecteur d'identifiant:

```
#nav-haut {
  width : 300px;
  float: left;
}
```

Pour un sélecteur de classe:

```
.imgfloatleft{
  margin-right : 10px;
  float: left;
}
```

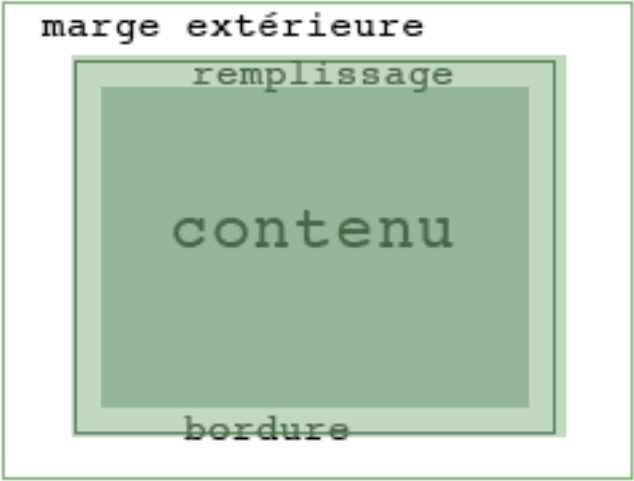
LES COMMENTAIRES CSS

Ils se font comme ceci :

```
/* ceci est un commentaire css */
```

LE MODELE DE BOITE

- Une boîte est composée de
- son contenu
 - son remplissage (marge interne ou padding)
 - sa bordure (border)



Les anciennes versions d’IE ayant un modèle de boîte différent de celui-ci, veuillez bien a définir votre `<!doctype html>` en début de page, pour les forcer à agir comme tous les autres navigateurs

Lorsque qu'on rajoute du remplissage à la boîte, cela augmente la taille de celle-ci. Si une boîte au départ fait 200px de large, et qu'on rajoute du remplissage de 10px de chaque côté, la boîte fera $200 + 10 + 10 = 220\text{px}$. Pour qu'elle garde son emplacement d'origine dans la page HTML, il faut donc dans le code lui donner une largeur de 180px, et un remplissage à gauche de 10px et un remplissage à droite de 10px. La marge externe est à l'extérieur de la boîte, mais pousse aussi les éléments qui seraient adjacents.

LES ELEMENTS DE TYPE «BLOCK»

Les éléments de type blocks prennent toute la largeur du conteneur et se mettent les uns en-dessous des autres.



Les éléments de liste se sont placés les uns en dessous des autres et prennent toute la largeur de la page ou du bloc dans lequel ils sont.

Les éléments de type block vus jusqu’ici sont

```
<p>
<h>
<ul>
<ol>
<li>
<div>
<section>
<article>
<aside>
<header>
<footer>
<table>
<figure>
```

On peut donner des largeurs et des hauteurs aux éléments de type «block»

Nous ne mentionnons ici aucune des balise de formulaire

LES ELEMENTS DE TYPE «INLINE»

Les éléments de type «inline» prennent la largeur de leur contenu, et se placent les uns a côté des autres. Il s’agit pricipalement des liens et des `<span>`. Les images sont en fait inline block. Elles se placent les unes à côté des autres mais ont une largeur

Les éléments de type «inline» prennent la largeur de leur contenu, et se placent les uns a côté des autres.

Les éléments de type inline vus jusqu’ici sont

```
<a>
<span>
```

```
<p>Dans ce paragraphe, un mot sera de couleur <span class="orange">orange</span> pour être mis en valeur</p>
```

On ne peut pas donner des largeur et des hauteurs aux éléments de type «block»

Les **images** se comportent comme des éléments de type inline, mais ont des hauteurs et largeurs intrinsèques, et seront donc des «inline-block»

FLOAT et CLEAR

Les éléments flottent toujours horizontalement

Une boîte flottante est retirée du flux normal, et placée le plus à droite (float: right) ou le plus à gauche (float: left) possible de son conteneur (par défaut la fenêtre). Les boîtes flottantes doivent avoir des largeurs.

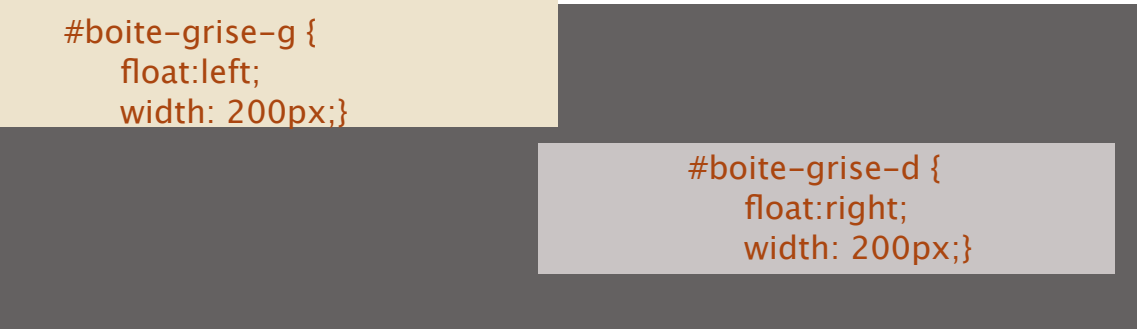
UTILISER LE FLOTTEMENT POUR FAIRE DES COLONNES



La 2e boîte est remonée à côté de la première.

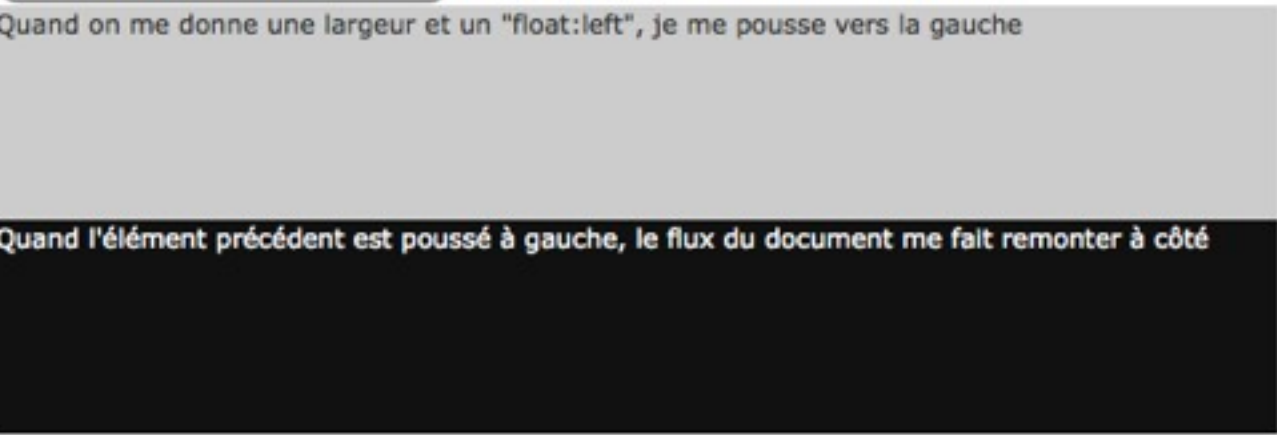


Mais les éléments flottent se mettent les uns à côté des autres, tant qu'ils ont la place. Si on réduit le parent :

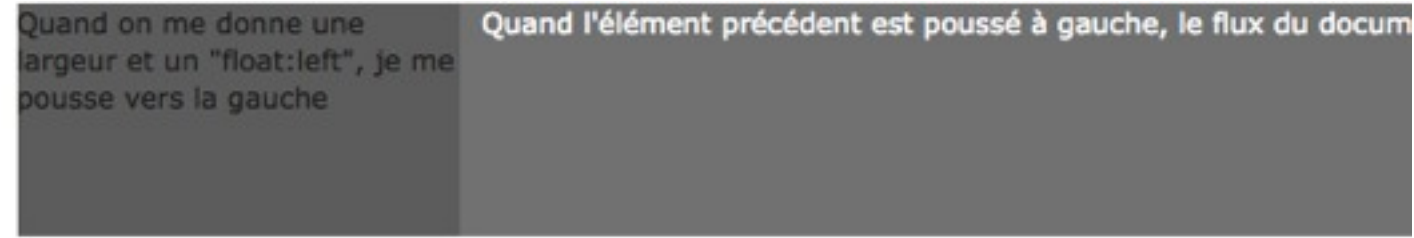


QUE SE PASSE-T-IL QUAND ON FAIT FLOTTER UN ELEMENT ?

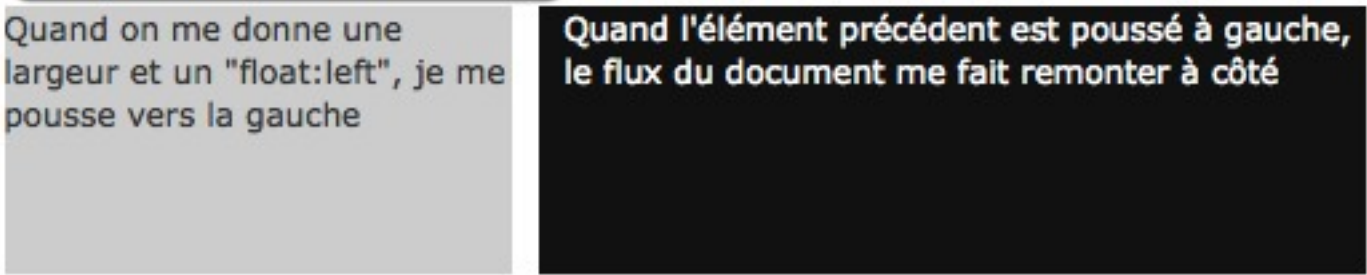
Ici deux blocs l'un sous l'autre, dans le flux normal



Il faut donner une largeur au premier bloc, puis le faire flotter, le deuxième bloc remonte avec le flux du document.



Il faut alors lui donner aussi une largeur

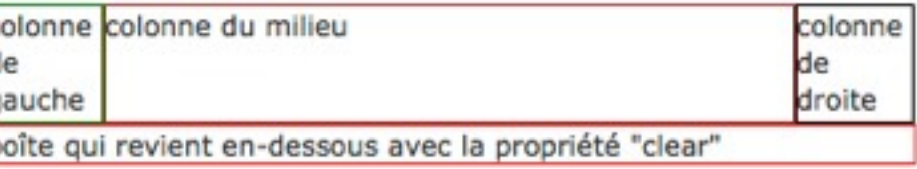


```
#boite-grise {
  float:left;
  width: 200px;}

#boite-noire {
  float:left;
  width: 300px;
  margin-right: 20px;}
```

SORTIR DU FLOTTEMENT : clear

Comme tous les blocs les uns à côté des autres doivent flotter, il faut ensuite dire à celui qui doit rester dessous, de ne pas suivre le flux du document : clear



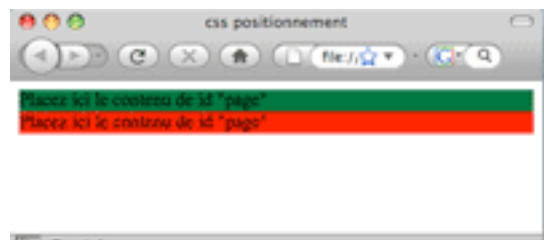
```
#leftcol {width: 100px; float:left;}
#midcol {width: 400px; float:left;}
#rightcol {width: 100px; float:left;}
#footer {width: 600px;clear: both}
```

clear prend 4 paramètres différents : left, right, both et none.

On utilise plus volontiers clear : both, plutôt que clear:left, ou clear : right

LE POSITIONNEMENT

Par défaut, le positionnement est «statique».



Les deux boîtes viennent l’une en dessous de l’autre. Si la première a plus de contenu elle va pousser la deuxième.

Les boîtes viennent naturellement les unes en dessous des autres, comme les paragraphes, les balises d'en-têtes et les listes, et d'autres éléments au contraire, comme les images ou les liens se placent naturellement les uns à côté des autres.

Positionnement relatif. `position:fixed;`

On peut changer le positionnement en donnant la valeur «relative», qui permet de modifier la position de l'élément par rapport à sa position de départ



```
#rouge {
background: red;
position:relative;
top:20px;
}
```

Dans les deux cas, on est dans le flux naturel du document.

Ici on a décalé la boîte rouge de 20px par rapport à la précédente.

ATTENTION

Le positionnement relatif peut générer des superpositions de boîtes

Positionnement fixe : `position:fixed;`

Un élément «fixe» l'est par rapport à la fenêtre du navigateur. Cela veut dire qu'il ne défilera pas en même temps que les autres éléments.

Souvent utilisé pour les icones de réseaux sociaux

```
.pos_fix {
position:fixed;
top:30px;
left :5px;
}
```

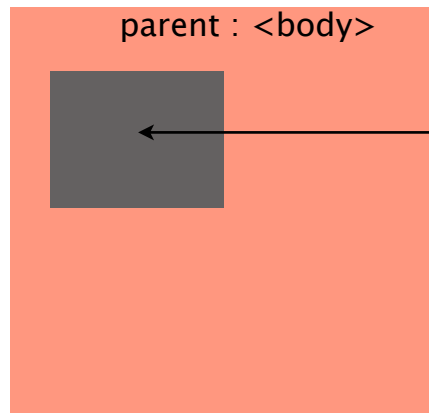
Ci-contre, reste toujours sur le premier écran, quand le reste de la page défile.

Les éléments positionnés en fixed ne prendront par défaut que la largeur et la hauteur de leur contenu

Il est possible de retirer les boîtes du flux normal de la page. Celles-ci n'auront alors aucun effet sur les autres éléments de la page.

Elles se placent par rapport à l'élément parent, si rien n'est précisé, ce sera la fenêtre.

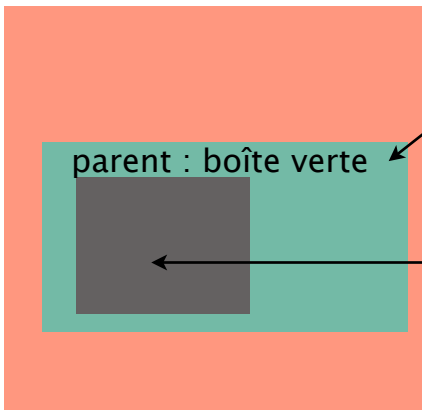
Si elles doivent dépendre d'autres éléments que le <body>, il faudra préciser à ces éléments qu'ils sont parents : soit position:relative; et il faut qu'ils soient dans le bloc parent dans le HTML.



```
position :
absolute;
top: 50px;
left :50px
```

Sur le schéma ci-dessus la boîte grise est à 50px du haut de la page et 50px de la gauche.

Sur le schéma ci-contre elle est à 50px du haut de la boîte verte et 50 px de la gauche de la boîte verte



```
position : relative;

position:absolute;
top: 50px;
left :50px
```

Il n'est pas nécessaire de dire au <body> d'être en relatif, c'est par défaut.

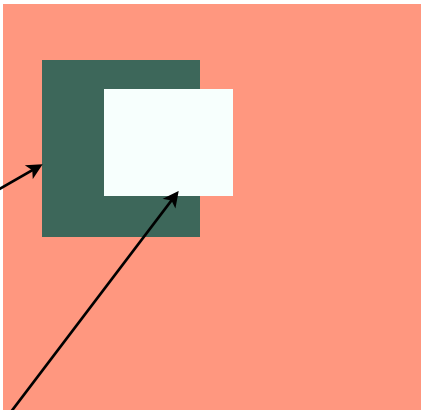
Attention aux chevauchements possibles entre les éléments qui s'ignorent !

Superposition d'éléments (positionnés)

Si deux éléments se superposent, le deuxième dans l'ordre du code sera par-dessus le premier. On peut influencer sur ceci avec le z-index. Plus le z-index est élevé, plus le bloc sera au premier plan

```
#vert
{position:absolute;
background: green;
top: 50px;
left :50px;
z-index: 1;
width: 100px;}

#blanc
{position:absolute;
top: 70px;
left :120px;
z-index: 10;
width: 100px;
background: #fff;}
```



STYLER LE TEXTE en CSS

Nous ne donnerons volontairement, ici que les valeurs les plus utilisées à ce jour, en commentaire les autres possibibilités.

Police	font-family	font-family: Verdana, Arial, sans-serif;
Couleur	color	color: #ccc; ou color: rgb(255,0,0)
Taille	font-size	font-size: 12px; ou font-size: 80%; ou font-size: 0.8em;
Gras	font-weight	font-weight: bold; pour mettre du gras sur un élément qui n'en a pas font-weight: normal; pour enlever du gras à un élément déjà gras
Italique	font-style	font-style: italic; met en italique font-style: normal; pour enlever l'italique à un élément qui le serait par héritage font-style: oblique; simule l'italique d'une police qui n'a pas d'italique
Majuscule	text-transform	text-transform: uppercase; met en majsucules text-transform: lowercase; met en minuscules text-transform: capitalize; met en majsucules la première lettre text-transform: none; enlève les majsucules
Petite majuscule	font-variant	font-variant : capitalize; change en mettant des petites majuscules existantes font-variant : normal; enlève les petites majuscules existantes
Alignement du texte	text-align	text-align : left; aligne le texte à gauche text-align : right; aligne le texte à droite text-align : center; centre le texte text-align : justify; justifie le texte
Interlignage	line-height	line-height : 20px; ou line-height : 120%;
Indentation de la première ligne	text-indent	text-indent : 20px; ou text-indent : 120%;
Espacement entre les mots	word-spacing	word-spacing : 20px; ou word-spacing : normal;
Espacement entre les lettres	letter-spacing	letter-spacing : 20px; ou letter-spacing : normal;
Décoration du texte	text-decoration	text-decoration : underline (souligné) text-decoration : none (enlève le soulignement) text-decoration : overline (ligne au-dessus) text-decoration : line-through (barré) text-decoration : blink (clignote, sauf dans IE)

Raccourcis possibles : voir le chapitre «raccourcis»

Choisir une liste de police installée sur tous les sysèmes et terminer par une famille, de préférence : sans-serif. Respecter la casse des noms de police

Utiliser des valeurs hexadécimales ou rgb

Possibilité d'utiliser des points (pt), pica (pc), pouce (in), millimètre (mm), centimètre (cm), pixel(px), em, ou pourcentage (%)

ATTENTION les pourcentages sont hérités

Il existe plusieurs valeurs de graisse en CSS, mais seules deux sont affichées : gras ou pas gras.

bold, bolder sont équivalents normal ou lighter sont équivalents

SMALL-CAPS

ATTENTION au texte justifié sur les pages web ! On ne gère pas l'espace créé entre les mots, le résultat peut être desastreux.

Il est possible de donner des valeurs négatives.

La valeur «normal» annule l'espacement dont un élément aurait hérité. C'est la valeur par défaut

Il n'est pas judicieux de souligner du texte, s'il ne s'agit pas de liens.

Couleurs

Pour le texte ou les arrière-plans, les couleurs sont indiquées de 3 façons possibles

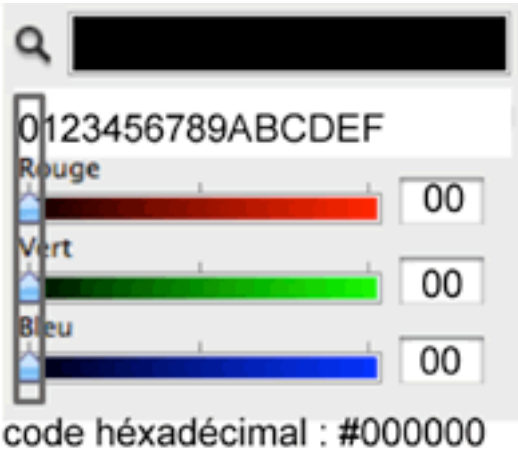
16 peuvent être nommés en anglais



black (noir)
gray (gris) attention à l'orthographe, il faut utiliser l'orthographe américaine et non anglaise
silver (gris argent)
white (blanc)
maroon (marron)
lime (citron vert)
green (vert)
orange (orange)
yellow (jaune)
olive (olive)
aqua (bleu clair)
blue (bleu)
navy (bleu marine)
fuchsia (rose fuchsia)
purple (violet)
teal (bleu vert).

Couleurs hexadécimales

En informatique les couleurs sont créées en additionnant le rouge, le vert et le bleu. En fonction des proportions de chacune on aura des tons différents.
Pour avoir 16 valeurs, on utilise un code hexadécimal. C’est à dire : les chiffres de 0 à 9 et des lettres de A à F, le tout précédé du signe dièse : « # ».
Un code hexadécimal commencera par le dièse suivi de 6 chiffres ou lettres, dont les valeurs correspondront au rouge, vert, bleu.



Pour un noir équivalent à aucun rouge, ni vert, ni bleu, le curseur est à 0 :

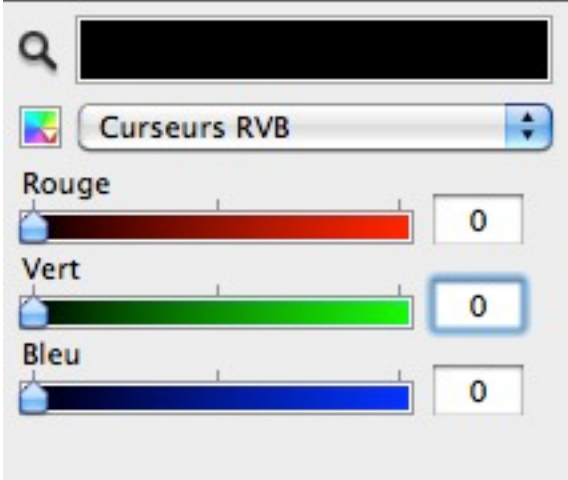
le code hexadécimal sera : #000000

Pour un paragraphe :
`p { color: #000000; }`

Les valeurs dont les chiffres vont par paire peuvent être divisées par deux :
au lieu de #000000
on peut écrire : #000.

Au lieu de : #ff88dd:
on peut écrire : #f8d

Couleurs rgb



Pour un noir équivalent à aucun rouge, ni vert, ni bleu, le curseur est à 0 :

le code rgb sera : 0,0,0

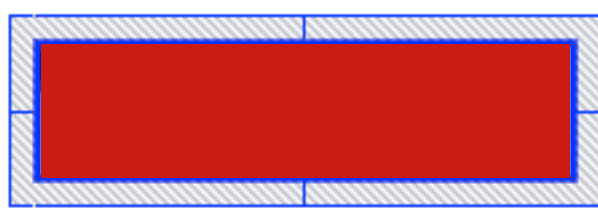
Pour un paragraphe :
`p { color: rgb(0,0,0; }`

En français on a l’habitude de dire RVB, ici il s’agit bien de RGB, pour red, green, blue

MARGES et REMPLISSAGES

Margin

Les marges sont à l'extérieur de la boîte.



Une boîte avec une marge de 10px.
L'arrière plan est gris, le contenu colle aux bords.

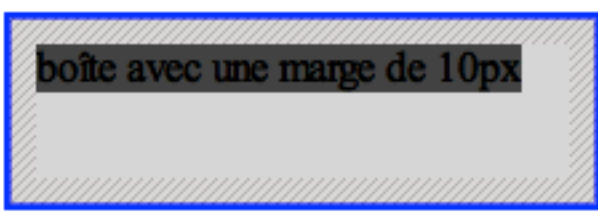
Les hachures représentent la marge.
Cette boîte fait 200px de large, mais occupe 200px + 10px à gauche et 10px à droite, soit : 220px.

Les marges peuvent prendre des valeurs négatives.

Pour remonter une boîte on peut écrire :
margin-top: -10px;
Elle sera plus haute de 10px qu'au départ.

Padding

Les remplissage sont à l'intérieur de la boîte.

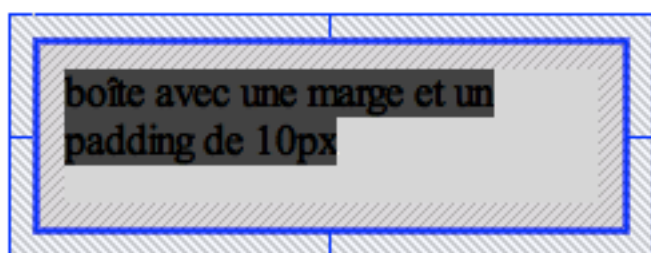


Une boîte avec un remplissage de 10px.
L'arrière plan est gris, le contenu est décollé des bords de 10 pixels.
Les hachures représentent le remplissage.
Cette boîte fait 200px de large, mais le remplissage a augmenté sa largeur. Elle fait donc 200px + 10px à gauche et 10px à droite, soit : 220px.

Les «padding» ne peuvent pas être négatifs

Cumul de margin et de padding

La même boîte avec à la fois la marge et le remplissage. Elle prend 240px de large.



Une seule valeur

Si on ne donne qu'une valeur, elle sera équivalente de tous les côtés.

margin : 10px; signifie qu'il y aura 10px tout autour de la boîte.
padding : 10px; signifie qu'il y aura 10px de remplissage de tous les côtés de la boîte.

Valeurs différentes

Si les valeurs de chaque côté sont différentes, il faut l'indiquer comme suit

margin-top : 10px; pour le haut
margin-right : 5px; pour la droite
margin-left : 20px; pour la gauche
margin-bottom : 8px; pour le bas

padding-top : 10px; pour le haut
padding-right : 5px; pour la droite
padding-left : 20px; pour la gauche
padding-bottom : 8px; pour le bas

Raccourci possible

Pour n'écrire qu'une ligne de code au lieu de 4, reportez vous au chapitre sur les raccourcis

BORDURES

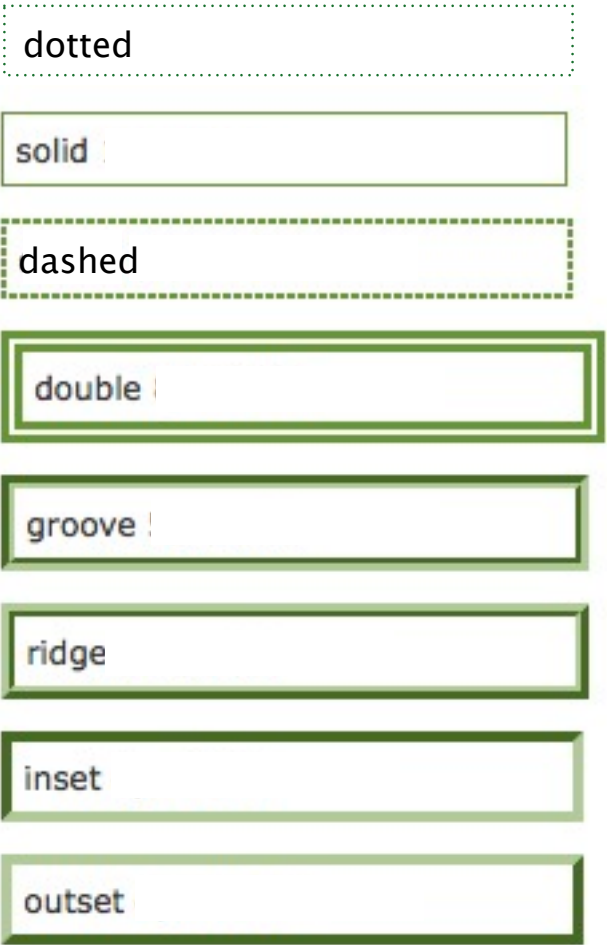
Elles peuvent s'appliquer à n'importe quelle balise.

Elles prennent IMPERATIVEMENT trois paramètres : style, épaisseur, couleur

Les **couleurs** sont données en hexacécimales

Les **styles** sont au nombre de 8 :

- "solid" = trait continu
- "dotted" = pointillé
- "dashed" = tirets
- "double" = double
- "groove" = crevassée
- "ridge" = crête
- "inset" = incrusté
- "outset" = en relief



Les **épaisseurs** peuvent être données en pixels ou "thin" (peu épaisse), "medium" (moyenne), "thick" (épaisse).

Le code sera pour <p> par exemple

```
p {  
  border-style: solid;  
  border-width : 2px;  
  border-color: #693;  
}
```

ATTENTION

Les bordures sont à l'intérieur des blocs. Si vous ajoutez une bordure à un élément, cela agrandira celui-ci, comme le remplissage.

ATTENTION

Si vous utilisez ces valeurs, vous dépendez de ce que les navigateurs interprètent, ce sera donc moins précis.

Raccourci possible

Pour n'écriture qu'une ligne de code au lieu de 3, reportez vous au chapitre sur les raccourcis

ARRIERE-PLANS

On peut mettre des couleurs ou des images

Pour les **couleurs** sur le <body> par exemple :

```
body {background : #ccc;}
```

Les **images** sont des images qui se répètent en mosaïque, par défaut.

Elles peuvent être très utiles pour donner des impressions de texture.
Elles sont légères à charger et se répètent, donc ne compte que pour une fois, ensuite elles sont dans le cache.



```
#fondrouge
{
background: url(fondrouge.gif);
height: 100px;
width: 200px;
color: #fff;
text-align: center;
}
```

Ici nous avons une image de 2px sur 4px, nommée «fondrouge.gif»  qui comporte une ligne de rouge une ligne de rouge plus sombre >> effet de trame

Les couleurs ou images d’arrière-plans peuvent être appliquées sur n’importe quel élément de type **block**.

Les couleurs se verront sur les éléments de type **inline**, mais **pas les images**. Car il faut donner la largeur et la hauteur de l’image d’arrière-plan, ce qui n’est pas possible sur ces éléments

L’appel à une image d’arrière-plan dans le code est **background : url(image.gif);**

Il faut bien entendu donner le bon chemin à l’image, depuis la feuille css

Images qui ne se répètent pas, à gauche et en haut par défaut

Par défaut elles se placeront à gauche et en haut de l’élément.

no-repeat = ne se répète pas

Les positionnements possibles en x et y, séparé par un espace

left top (par défaut)
left center
left bottom

center top
center center
center bottom

right top
right center
right bottom

ou des valeurs en px ou % qui partiront du haut et de la gauche

10px 50px
= 10px de la gauche, et 50px du haut

repeat-x se répète à l’infini (du conteneur) horizontalement

Attention aux raccords !

repeat-y se répète à l’infini (du conteneur) verticalement

Attention aux raccords !

II

Images qui ne se répètent pas, placée en bas et à droite

left top (par défaut)
left center
left bottom

center top
center center
center bottom

right top
right center
right bottom

ou des valeurs en px ou % qui partiront du haut et de la gauche

10px 50px
= 10px de la gauche, et 50px du haut

repeat-x se répète à l’infini (du conteneur) horizontalement

Attention aux raccords !

repeat-y se répète à l’infini (du conteneur) verticalement

Attention aux raccords !

II

Images qui se répètent en x



```
#bloc
{
background: url (spirales_V.jpg) repeat-x;
width: 100px;
color: #000;
padding: 10px 10px 5px 30px;
border : 1px solid #7d755e;
}
```

repeat-x se répète à l’infini (du conteneur) horizontalement

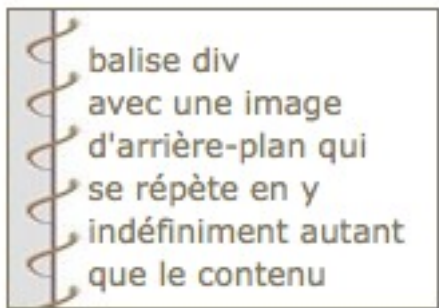
Attention aux raccords !

repeat-y se répète à l’infini (du conteneur) verticalement

Attention aux raccords !

II

Images qui se répètent en y



```
#cahier
{
background: url (spirales_V.jpg) repeat-y;
width: 100px;
color: #000;
padding: 10px 10px 5px 30px;
border : 1px solid #7d755e;
}
```

repeat-x se répète à l’infini (du conteneur) horizontalement

Attention aux raccords !

repeat-y se répète à l’infini (du conteneur) verticalement

Attention aux raccords !

II

DECORATION DES LISTES

Par défaut les listes sont précédées d’une puce.

Avec la propriété : list-style on peut en changer l’aspect.

Le plus utilisé sera sans doute enlever les puces, pour ensuite mettre des images en arrière-plan à la place
`list-style:none ;`

DECORATION DES LISTES avec image d'arrière plan

Il faut créer une petite image, d’environ 12px sur 12px ou moins. ★

Code HTML

```
<ul id="menu-haut">
  <li><a href="../index.html">Accueil</a></li>
  <li><a href="destinations/index.html">Destinations</a></li>
  <li><a href="services/index.html">Services</a></li>
  <li><a href="contact/index.html">Contact</a></li>
</ul>
```

Code CSS

```
#menu-haut li {
  list-style:none;
  background: url(bg-menu-haut.gif) no-repeat left center;
  text-indent : 15px; /* ou padding-left: 14px*/
}

#menu-haut li a{
  color : #F96;
  text-decoration:none;
}
```

Résultat

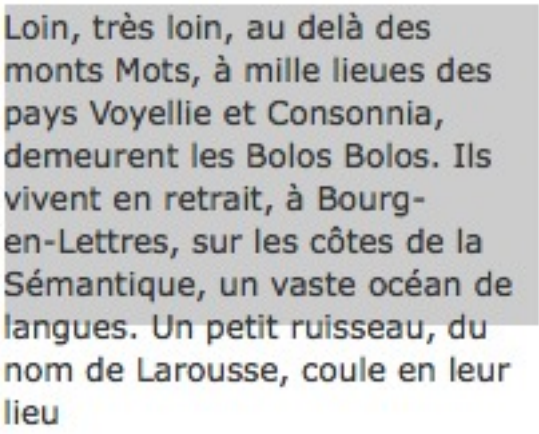


- disc (par défaut)
- disc
- disc
- circle
- circle
- circle
- square
- square
- square
- lower-alpha (des lettres en minuscules)
- a. 1er element
- b. 2e element
- upper-alpha(des lettres en majuscules)
- A. 1er element
- B. 2e element
- georgian
- ⴀ. georgian
- Ⓑ. georgian
- Ⓐ. georgian
- ⓐ. georgian
- armenian
- Ա. armenian
- Ⓐ. armenian
- ⓐ. armenian
- cjk-ideographic
- 一. cjk-ideographic
- 二. cjk-ideographic
- 三. cjk-ideographic
- hebrew
- א. hebrew
- ב. hebrew
- א. hebrew
- hiragana
- あ. hiragana
- い. hiragana
- う. hiragana
- hiragana
- い. hiragana-iroha
- ろ. hiragana-iroha
- は. hiragana-iroha

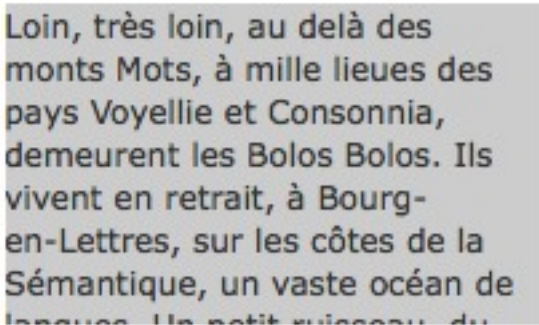
DEBORDEMENT

Par défaut le débordement est visible.
overflow : visible

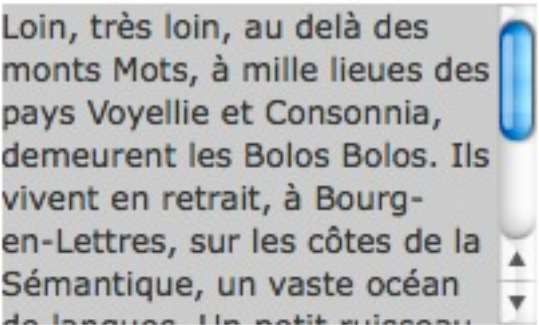
Le contenu d’un bloc peut déborder sur le bloc a une hauteur fixe



On peut masquer le débordement
overflow : hidden



On peut générer automatiquement une barre de défilement si nécessaire overflow : auto



overflow : hidden

il sert également à forcer un conteneur à augmenter sa hauteur et suivre les éléments flottants qu’il contient



Indiquer à ce conteneur : overflow:hidden lui permettra de suivre son contenu.



Attention, cette propriété peut être très gênante dans certains cas, notamment si elle est combinée avec une position relative.

RAPPEL :

Lorsque l’on place des éléments flottants dans un conteneur, celui-ci ne suit pas la hauteur des éléments flottants.

Une solution pour que l’arrière plan continue est de mettre un élément «clear» en bas de celui-ci.

DISPLAY (affichage)

La propriété display sginifie affichage

Elle prend beaucoup de paramètres, donc seuls peu sont compris par tous les navigateurs.

display : block
permet d'afficher comme des éléments de type blocks ceux qui ne l'étaient pas. On peut donc leur donner une largeur et une hauteur, et ils se positionneront les uns sous les autres.

display : inline
permet d'afficher comme des éléments de type inline ceux qui ne l'étaient pas. On ne peut donc plus leur donner une largeur et une hauteur, et ils se positionneront les uns à côté les autres.

display : none
permet de ne pas afficher des éléments

display : inline-block
permet de afficher des éléments de type blocks les uns à côté des autres, sans les faire flotter, ils prennent donc les propriétés des blocs et des inline

La propriété inline-block est de plus en plus utilisée pour éviter les «floats»

Il faut cependant prendre en compte que les versions de IE antérieures à 8 ne la comprennent pas sur les éléments de type block;

Elles posent parfois des problèmes d'espacement non voulus !

voir :
<http://www.alsacreations.com/article/lire/1209-display-inline-block.html>

L'astuce est de faire une feuille de style conditionnelle pour IE7 et moins

```
<!--[if lte IE 7]>
<link rel="stylesheet"
type="text/css"
href="styles-ie7.css" />
<![endif]-->
```

Dans laquelle vous indiquez :

```
votre-element {
    zoom: 1; /*offre le
layout à un élément (*)*/
    display: inline;
}
```

tiré de l'article cité ci-contre

RACCOURCIS

En anglais "shorthand", il s'agit de façons de réduire le code, en évitant les répétitions.

Par exemple, toutes les propriétés qui commencent par "font" peuvent être cumulées en une seule ligne.
On commence donc par «font»

Le texte (font)

```
font-size-adjust: none;
font-weight: bold;
font-style: normal;
font-variant: normal;
font-size: 1em;
line-height: 1.2em;
font-family: sans-serif
```

font: bold 1em/1.2em Verdana, sans-serif

ATTENTION

La couleur ne peut pas en faire partie, puisque la propriété ne commence par «font»

1em/1.2em donne la taille puis l'interlignage

Les remplissages (padding) Equivalents partout

```
padding-top: 4px;
padding-right: 4px;
padding-bottom: 4px;
padding-left: 4px;
```

padding: 4px;

Les marges (margin) Equivalents partout

```
margin-top: 4px;
margin-right: 4px;
margin-bottom: 4px;
margin-left: 4px;
```

margin: 4px;

Les remplissages (padding) Identiques haut-bas et identiques droite-gauche

```
padding-top: 4px;
padding-right: 8px;
padding-bottom: 4px;
padding-left: 8px;
```

padding: 4px 8px;

Les marges (margin) Identiques haut-bas et identiques droite-gauche

```
margin-top: 4px;
margin-right: 8px;
margin-bottom: 4px;
margin-left: 8px;
```

margin: 4px 8px;

On donne d'abord la valeur pour le **haut et bas**, puis après un espace, la valeur **droite-gauche**

Les remplissages (padding) Différents partout

```
padding-top: 4px;
padding-right: 8px;
padding-bottom: 6px;
padding-left: 10px;
```

padding: 4px 8px 6px 10px;

Les marges (margin) Différents partout

```
margin-top: 4px;
margin-right: 8px;
margin-bottom: 6px;
margin-left: 10px;
```

margin: 4px 8px 6px 10px;

On donne dans l'ordre des aiguilles d'une montre en commençant par le haut

HAUT
DROITE
BAS
GAUCHE

Si vous aimez les moyens mnémotechniques et l'anglais cela fait

TRBL (trouble)

RACCOURCIS

Couleurs, bordures et arrière-plans

Couleurs (color)

Si chaque valeur de rouge, vert et bleu est doublée, on peut diviser par deux

```
color : #ffffff;
color: #ff4466;
```

```
color: #fff;
color: #f46;
```

LES RACCOURCIS SONT
SEPARES PAR DES ESPACES

On ne peut donc avoir que 6
ou 3 valeurs.

Les bordures (border)

Elles ont trois paramètres : style, épaisseur et couleur

La même bordure partout

```
border-style : solid;
border-color : red;
border-width : 1px;
```

```
border : solid 1px red;
```

L'ordre dans lequel on
indique la couleur,
l'épaisseur et le style n'est
pas important

Une bordure différente partout :
utiliser le raccourci ci-dessus pour chaque côté

```
border-top-style : solid;
border-top- color : red;
border-top- width : 1px;

border-left-style : dotted;
border-left- color : blue;
border-left- width : 2px;

border-right-style : solid;
border-right- color : red;
border-right- width : 3px;

border-bottom-style : dashed;
border-bottom- color : blue;
border-bottom- width : 4px
```

```
border-top : solid 1px red;
border-left : dotted 2px blue;
border-right : solid 3px red;
border-bottom : dashed 4px blue;
```

Pour 3 identiques et celui du bas
différent :
On commence par dire tous les
mêmes, puis on corrige pour le
dernier. Ici le dernier sera pris en
compte.

```
border : 1px solid red;
border-bottom : 2px solid green
```

Les arrière-plans (background)

L'ordre sera :
couleur image répétition position x et position y

```
background-color : #ccc;
background-image : url(img/bg-body.jpg);
background-repeat : no-repeat;
background-position : center top;
```

```
background : #ccc url(img/bg-body.jpg) no-repeat center top;
```

RAPPEL :

les arrière-plans peuvent avoir
des couleurs et des images.
L'image viendra par-dessus la
couleur, pour un même
élément.

Attention aux espaces,
notamment avant le
«no-repeat», sans quoi IE7 et
8 ne vous afficheront pas du
tout l'image.