

www

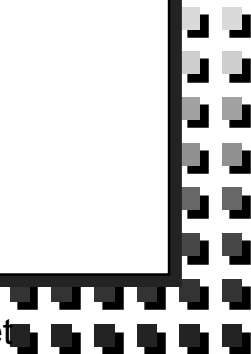
# Programmation sur Internet

## Cours CSS : présentation des documents HTML



Jean-Michel Ilié

IUT Paris 5 - département informatique



# Modèle de styles pour document Cascading Style Sheet (css)

*Style : mise en forme des pages HTML*

- *apparence à donner aux balises et leur contenu (couleur)*
- *mise en forme et positionnement graphique dans la fenêtre du navigateur.*

*HTML $\geq$ 4.0 : style "écrit une fois, afficher partout"*

**Application CSS variée aux balises du document HTML**

**Effet cascade : Propagation des styles des balises à leur contenu HTML**

**Les attributs de style du HTML deviennent obsolètes.**

# Modèle de styles pour document Cascading Style Sheet (css)

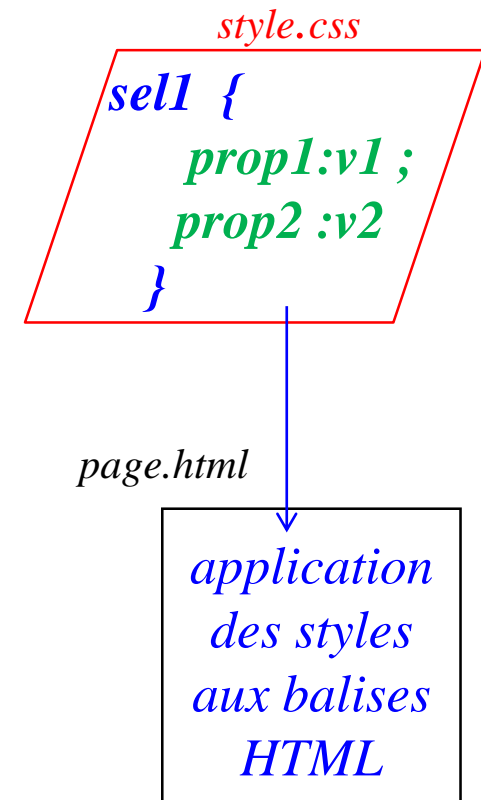
## Déclaration de styles sous forme de règles

- ☞ conditions d'application : **sélecteur**
- ☞ liste de styles à appliquer : **propriétés de style**

*Un sélecteur spécifie  
un sous ensemble des balises  
du document HTML.*

*Les propriétés de style sont pré-définies  
et ont une valeur par défaut,  
on ne fait que modifier les valeurs courantes.*

☞ commentaires : `/**/`



# Attacher un fichier CSS à un document HTML

## Développement séparé HTML et CSS

- ☞ partage du même fichier :
- ☞ plusieurs fichiers de style :

Support de **charte graphique**  
Développement **modulaire**,  
**Adaptation** à différents media.

```
<html><head>  
  <link rel="stylesheet" type="text/css"  
        href="http://.../modele.css" />  
</head>  
<body>...</body></html>
```

par balise **LINK**  
ou mot-clé css **@IMPORT**

# Alternative : déclarations directes des styles dans le fichier HTML

☞ Définition dans l'entête :

```
<html>
<head>
  <style type="text/css">
    ... règles_de_style, exple p {...}
  </style>
</head>
<body> .... </body></html>
```

☞ Définition avec l'attribut **style** dans **une** balise

```
<html><head></head>
<body>. <p style=" ... propriétés_de_style ... " >
  un texte stylisé
</p>
</body></html>
```

# Sélecteurs élémentaires



☞ nom d'une balise

*sélections de toutes  
les balises possédant ce nom.*

☞ symbole \*

*sélections de toutes  
les balises.*

# Exemple

1 <html><head>

<style type="text/css">

h1 { color: blue; text-decoration: underline; }

h2 { color: red; font-style: italic; }

</style></head>

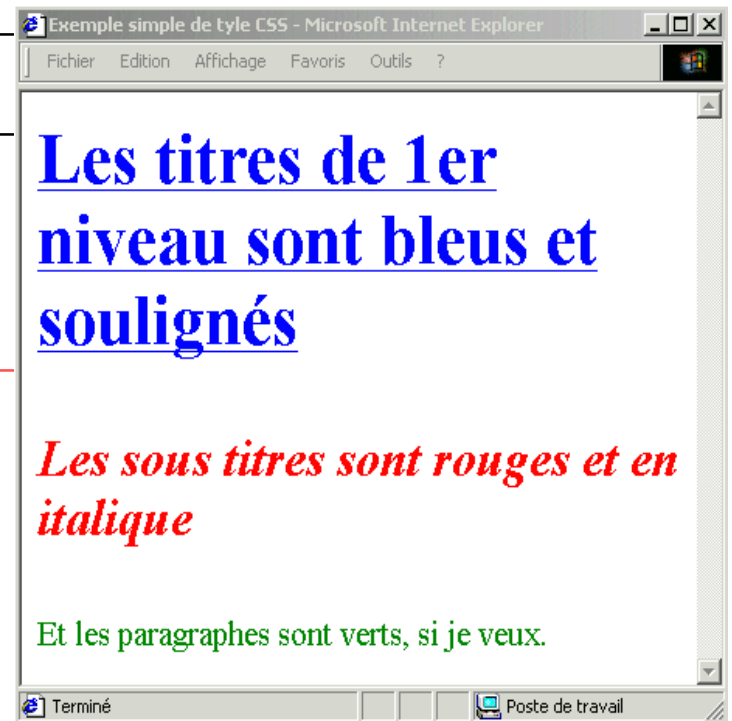
<body>

<h1>Les titres de 1er niveau  
sont bleus et soulignés</h1><br />

<h2>Les sous titres sont rouges  
et en italique</h2><br />

<p style="color: green; text-decoration: none;">  
Et les paragraphes sont verts, si je veux. </p>

</body></html>



# Spécialisation des sélecteurs avec class ou id

Classes de style distinguant DES balises <tag **class="nomClasse"**>

☞ tag.**nomClasse** {attribut: val; ...;}

☞ **.nomClasse** {attribut: val; ...;} //pour toutes les balises avec nomClasse

Définition de style pour UNE balise identifiée <tag **id="nomID"**>)

☞ tag**#nomID** {attribut: val; ...;}

☞ **#nomID** {attribut: val; ...;} //quelque soit la balise avec nomID

- il peut y avoir plusieurs balises avec le même nom de classe
- l'attribut id, servant d'identifiant, n'apparaît que dans une balise,
- Dans une balise, on peut combiner plusieurs valeurs de class et un id

# Exemples avec class et id

```
<html><head><style type="text/css">
```

```
p { color: black; text-decoration: none; }
```

```
.couleur { color: blue; }
```

```
#note { color: red; font-style: italic; }
```

```
</style></head>
```

```
<body>
```

```
<h3 class="couleur"> Section bleue </h3>
```

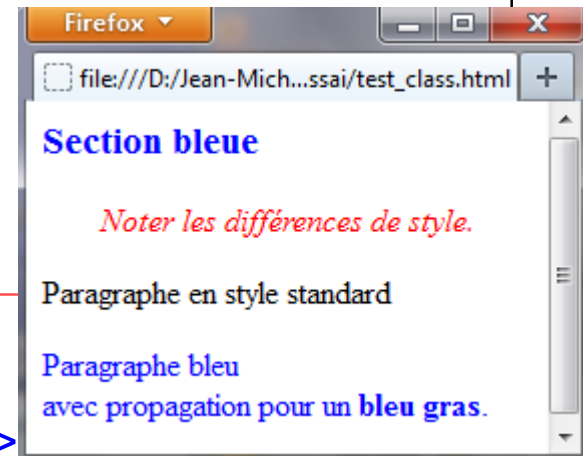
```
<p id="note">Noter les différences de style.</p>
```

```
<p>Paragraphe en style standard</p>
```

```
<p class="couleur">Paragraphe bleu</br>
```

```
avec propagation pour un <b>bleu gras</b>.</p>
```

```
</body></html>
```



# Spécialisation des sélecteurs avec les pseudo-classes

☞ **Concerne l'état des balises interactives** : liens, boutons, ....

|                            |   |
|----------------------------|---|
| <code>a:link { }</code>    | <code>//liens non visités</code>                              |
| <code>a:visited { }</code> | <code>//liens visités</code>                                  |
| <code>a:hover { }</code>   | <code>//liens survolés</code>                                 |
| <code>a:active { }</code>  | <code>//liens activés (clic en cours)</code>                  |
| <code>a:focus { }</code>   | <code>//mise en valeur du dernier activé, après activé</code> |

*Note : les Pseudo-classes peuvent s'appliquer  
à sélecteurs avec classes ou identifiants*

*exple : `a:link.menu{...}`*

# Exemples de pseudo classes

```
<html><head> <style type="text/css">
```

```
a:link {color:blue; text-decoration:none}
```

```
a:hover {color:green; text-decoration:none}
```

```
a:visited {color:red}
```

```
a.italique:link, a.italique:hover, a.italique:visited
```

```
{color: #CC00CC; font-style="italic"; text-decoration:none}
```

```
</style></head>
```

```
<body>
```

```
<a href="aff1.html"> Lien non souligné </a><br>
```

```
<a class="italique" href="..."> Lien en italique </a>
```

```
</body></html>
```



# Spécialisation des styles : balises imbriquées ou adjacentes

- ☞ > ou ' ' Balise imbriquée dans une autre
- ☞ + Balises dépendantes implicitement (exple H1 suivi de H2)

## Exple

```
td > ul {} //style pour UL si descendant direct de TD
td a {} //style pour A qu'il soit descendant direct ou indirect de TD
h1 + h2 {} //style pour H2 si suivante et adjacente de H1
```

*Note : Les sélecteurs peuvent être plus complexes. Par exemple,  
quand les balises sont complétées par un nom de classe ou d'identifiant.*

*div#pr a.st {...}*

# Spécialisation des styles : balises sous contraintes d'attributs

 [*nomAttr*]                      Balise repérée par ses attributs  
 [*nomAttr*="val"]

## Exple

a[title] {}            //style pour A si possède explicitement un attribut 'title'.

a[title="ext"] {}        //style pour A si son attribut 'title' égal "ext"

a[title~="ext"] {}        //style pour A si son attribut 'title' contient "ext"

a[title|= "ext"] {}        //style pour A si son attribut 'title' débute par "ext"

*Note : les contraintes se combinent*

*exple      a.st [title][href=""]:visited {....}*

# Raccourcis d'écriture

## ☞ Factorisation des propriétés :

le même jeu d'attributs de style peut être défini pour plusieurs sélecteurs



```
input.gris { color:grey }
```

## ☞ Factorisation de style :

exploiter le style principal de la balise pour factoriser les règles communes à ces différentes classes.



```
input { width : 2em }  
input.rouge { color:red }  
input.gris { color:grey }
```

## ☞ Description modulaire des classes :

définir des classes simples et les composer.



```
.text { color : blue }  
.dim { width : 2em }
```

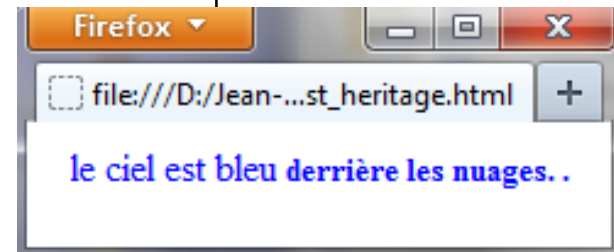
```
<input class="text">  
<input class="dim">  
<input class="text,dim">
```

# Héritage des styles

## 👉 héritage de style dû à l'imbrication des balises :

```
<html><head><style type="text/css">  
  .bleu {color: blue; }  
  b {font-size:small}  
</style></head>
```

```
<body><p class="bleu">  le ciel est bleu  
  <b>derrière les nuages.</b> </p>
```



👉 les styles ne sont pas propagés aux balises complexes (tableaux, formulaires)

# Dimensionnement et positionnement graphique

Les balises non vides d'un document sont vues comme un ensemble de **conteneurs imbriqués**, suivant 2 types d'affichage:

- ☞ Les conteneurs **inline** sont placés au fil de l'eau par le navigateur (le flux naturel du navigateur).
- ☞ Les conteneurs **block** (blocs) pourront être:
  - placés graphiquement dans la fenêtre
  - mais aussi dimensionnés et décorés.

- Exple : `<b><i>texte</i></b>` (b contient i qui contient texte)
- body est le bloc parent le plus externe
- 2 conteneurs supplémentaires spécialisés prévus pour l'organisation du document :
  - `<div>` type bloc
  - `<span>` type inline.

# Exemple d'utilisation des conteneurs "inline" avec span

```
<html> <body>
```

Que fait ce mot

```
<span style="color:blue;
font-weight:bold;
font-size:larger;
text-transform:uppercase"
font-family:"Times New Roman";
```

```
>
```

**bleu**

```
</span>
```

dans ce paragraphe.

```
</body></html>
```



# Exemple d'utilisation des conteneurs de type "blocs" avec div

```
<html> <body>
```

```
<div id="header">
```

entete de la page HTML

```
</div>
```

```
<div id="nav">
```

onglets de la page HTML

```
</div>
```

```
<div id="page">
```

contenu applicatif de la page HTML

```
</div> <!-- fin page -->
```

```
<div id="footer">
```

bas de la page HTML

```
</div>
```

```
</body></html>
```

Division de la page en conteneurs positionnables et décorables suivant définitions de styles css.

# Définition des types de conteneurs par la propriété display

Toutes les balises ont la propriété **display**.

## display :

- ☞ **block** → *le contenu HTML de la balise est affiché comme un nouveau bloc.  
Il peut être retiré du flux d'écriture et positionné.*
- ☞ **inline** → *le contenu HTML reste dans le flux d'écriture de la balise*
- ☞ **none** → *pas d'affichage*

## Valeur par défaut de l'attribut display dans les balises :

**block** : **body**, **p**, **h1**, ..., **table**, **ul**, **li**, **div**

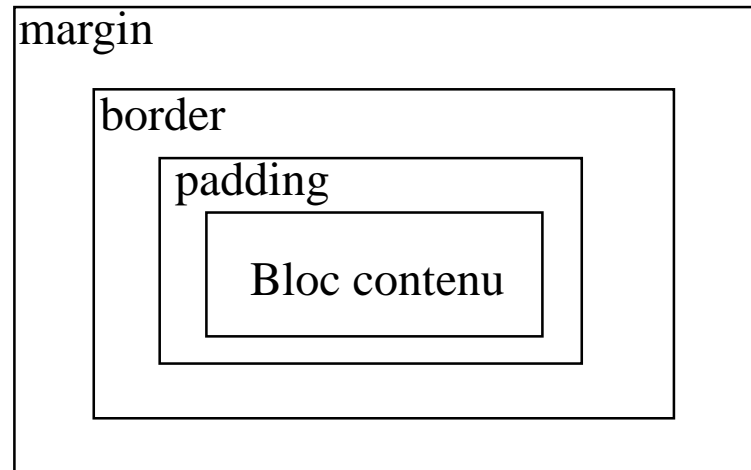
**inline** : **i**, **b**, **input**, ..., **span**

La valeur de display  
peut être re-définies.

# Décoration des blocs

## 3 propriétés de style :

- ☞ **padding** : espacement autour du contenu affichable
- ☞ **border** : cadre (visible ou invisible), ajustable sur chaque côté
- ☞ **margin** : espacement inter-conteneur.



# Valeurs et unités de mesures pour le dimensionnement des blocs

## ☞ unités fixes :

**px** (*pixels*), **pt** (*points*), **cm** (*centimètre*),  
*pour un travail précis*

## ☞ **pourcentage** (%)

*relative à la taille du bloc parent*

## ☞ **em** proportionnelle à la taille de la police courante

ou **ex** : proportionnel au 'x' courant.

*Exple : 0.8em est équivalent 80% de la taille de la police*

## ☞ valeurs symboliques

- **auto** automatique, fonction des choix classiques du navigateur
- **inherit** héritage de valeur définie pour la propriété dans le **bloc parent**
- *small, smaller, x-small, xx-small, medium, large, larger, x-large, xx-large*

Pour une présentation robuste,  
prendre garde aux transformations  
(effet de zoom et redimensionnement)

**Préférer les unités proportionnelles  
ou symboliques**

# Indications des dimensions des blocs



## Globalement pour tous les côtés

### ☞ *padding, border, margin*

- 4 dimensions : *top, right, bottom, left*
- 3 dimensions : *top et left-right et bottom*
- 2 dimensions : *top-bottom et left-right*
- 1 dimension : *partout.*

```
div#principal {  
    margin: 0 2pt 0 0.2pt;  
    padding: 2em 2em;  
    border: 0;  
}
```

## ou explicitement pour chaque côté séparément

### ☞ *-top, -right, -bottom, -left*

### ☞ *border-left, border-top, border-right, border-bottom*

### ☞ *margin-left, margin-top, margin-right, margin-bottom*

```
li {  
    margin: 0 ;  
    padding-left: 2em;  
    border-bottom: 2;  
}
```

# Définition des bordures

## Formule détaillée

☞ **border-color** : 2 à 4 valeurs //couleur du bord

☞ **border-width** : 2 à 4 valeurs //largeur du bord  
 ☞ symboles : *thin* | *medium* | *thick*

☞ **border-style** : // style du bord  
 symboles : *hidden* | *dashed* | *dotted*  
*double* | *solid* ...

*code du nb de valeurs*  
 1 val : toutes  
 2 val : LT et RB  
 3 val : L et TB et R  
 4 val : L, T, R, B

*Formes compactes permettant de citer plusieurs valeurs en une fois :*

**border** : <épaisseur> <couleur> <style> ;

Exple : **border** : 0.8em green solid;

Idem pour **border-top**, **border-left**, **border-right**, **border-bottom**.

# Décoration d'un fond : couleur et image

**background-color** : *val* / *transparent*

*//couleur de fond*

**background-image** : *url (...)*

*//URL de l'imade de fond*

**background-repeat** : *no-repeat* / *repeat* / *repeat-x* / *repeat-y* *//répétition d'image*

**background-position** :

*// décalage de l'image des bords gauche et haut*

*val* / *val%* /

*// avec 1 valeur commune aux axes X et Y*

*val,val* / *val%,val%* /

*//ou des valeurs soit littérales*

*[top* / *center* / *bottom* ], *[left* / *center* / *right]* *//soit symboliques*

**background-attachement** : *fixed* / *scroll* *//déroutement du fond lors des ascenceurs*

**background-size** : *val% val%* / *cover* *//ajustement du fond à la taille du conteneur*

*Forme compacte :*

**background** : *<couleur>* *url(http://...)* *<position>* *<repetition>* *<fixation>* ;

**background** : *#545454* *url(images/img.gif)* *0 6px* *repeat-x* *fixed*;

# Propriétés de dimensionnement des blocs

## *height, width*

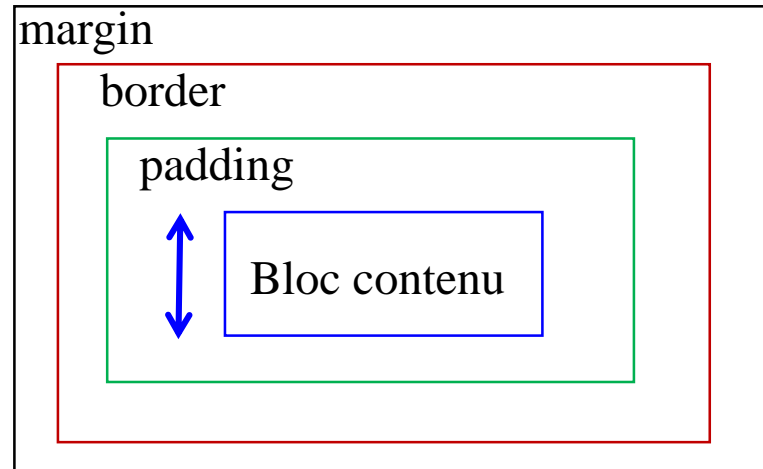
### **Largeur par défaut :**

- ☞ ce qui reste de largeur au moment de son placement dans le bloc parent.

### **Hauteur par défaut:**

- ☞ hauteur du contenu du bloc
- ☞ nulle si le bloc est vide

*height*



*width*



*Contrôler le dimensionnement height-width avec :  
(la marge reste non incluse):*

***box-sizing** : content-box / padding-box / border-box*

# Propriétés de placement flottant des blocs

*On peut forcer l'affichage de blocs à être placé sur la gauche ou la droite de leur bloc parent. Le reste du flux d'écriture coule alors sur le reste de la ligne à gauche ou à droite suivant l'indication de flottement. quand la taille du bloc est dépassée, le flux passe sous le bloc. Par exemple : une image avec un texte qui coule autour.*

## **float**

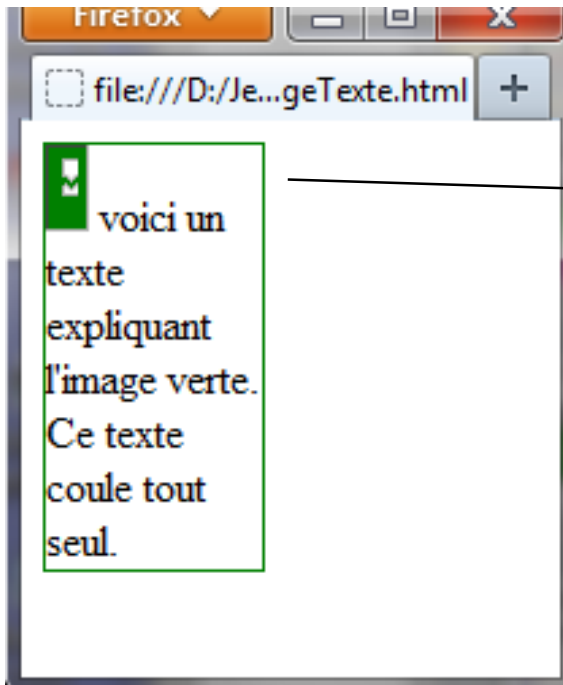
- ☞ **right** // placement sur la marge droite du bloc parent
- ☞ **left** // placement à gauche
- ☞ **none** //défaut : affichage normal au sein du flux.

**clear** //permet d'annuler un effet du flottement (dû à un précédent bloc 'float')

- ☞ **right | left | both**

# Exemple de placement flottant

L'image est appelée à gauche.



```
<html><body >
```

```
<div style="width:5em;" >
```

```
<img style="float:left" />
```

*voici un texte expliquant l'image verte.*

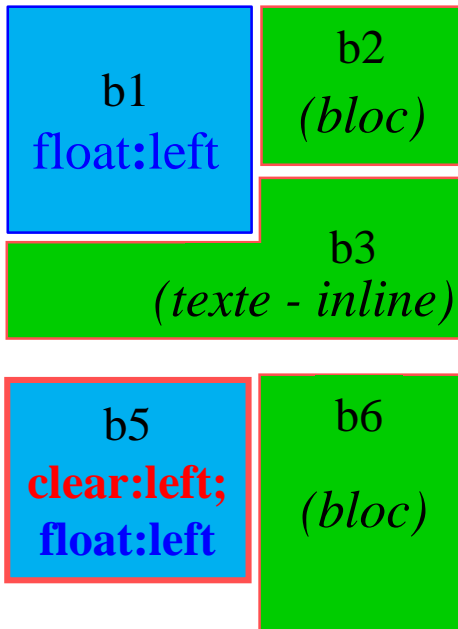
*Ce texte coule tout seul.*

```
</div>
```

```
</body></html>
```

# Exemple de placement avec flottement de texte

conteneur parent



```
//style
```

```
#b1 { float:left; }
```

```
#b2 { display:block }
```

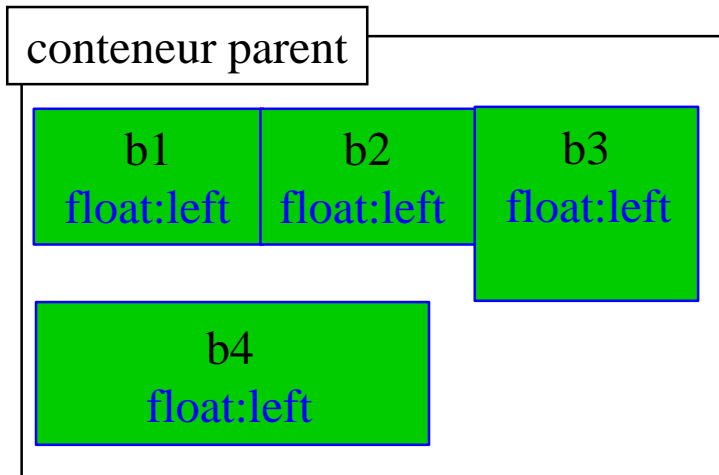
```
#b3 { display:inline }
```

```
#b5 { clear:left;float:left; }
```

```
#b6 { display:block }
```

- le placement des blocs float est prioritaire sur le placement du reste du flux
- 2 systèmes de flottement sont séparés par un clear
- le texte et les éléments inline coule mais pas les blocs

# Exemple de placement flottant de blocs



```
//style
#b1 {float:left; width:5em;height:5em}
#b2 {float:left; width:5em;height:5em}
#b3 {float:left; width:5em; height:7em}
#b4 {float:left; width:8em;height:5em}
```

s'il y a assez de place sur la ligne,  
les éléments float de même type (left ou right) successifs,  
se suivent sur la même ligne,  
sinon le navigateur place les derniers blocs sur une nouvelle ligne.

# Propriétés de positionnement des blocs



*Hors positionnement static, fait par défaut, le bloc est retiré du flux pour être placé.*

## *position*

- ☞ `static` // (valeur défaut) placement dans le flux HTML courant ,  
s'affiche en début d'une nouvelle ligne.
- ☞ `absolute` // indication de placement absolue dans la page  
référence : point haut gauche du *body*.
- ☞ `fixed` // indication de placement dans la fenêtre  
référence : point haut gauche de la fenêtre  
*insensible au mouvement d'ascenseur !*
- ☞ `relative` // indication de décalage relatif du bloc.

# Exemple de positionnement static

```
<html><body>
```

*avant d1*

```
<div style="width:5em; height:3em" >  
  </div>
```

*après d1*

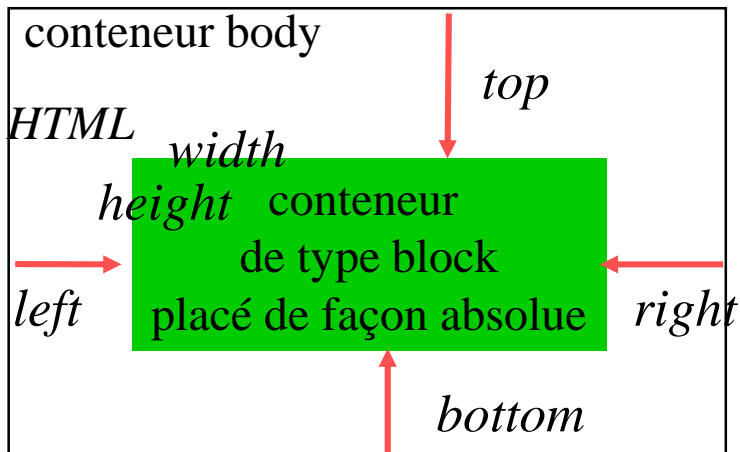
```
<div style="width:5em; height:2em" >  
  </div>
```

```
</body></html>
```



# Placement des blocs en 'absolu'

Positionnement de référence : *coin haut gauche du body*



## Dimensionnement :

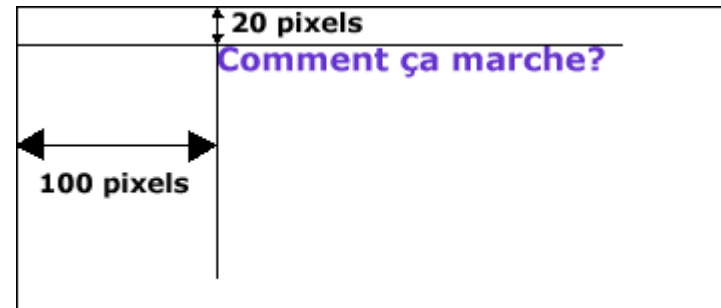
- valeur défaut : *auto* (automatique)
- dimensions : *height*, *width*
- réduction en rapport au bloc conteneur:
  - *left*, *top*,
  - *right*, *bottom*

### Note :

- pour une image : préciser nécessairement la taille maximale à respecter pour le conteneur (*height* et *width*).

# Exemple de positionnement absolu

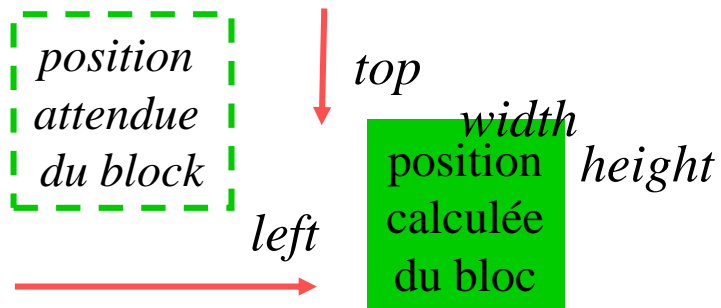
```
<html><body>  
<div style="position: absolute;  
           top: 20 px; left: 100 px;">  
Comment ça marche?</div>  
</body></html>
```



# Placement des conteneurs de type bloc en relatif

Décalage par rapport à la position  
*courante d'affichage*  
dans le bloc parent .

conteneur parent

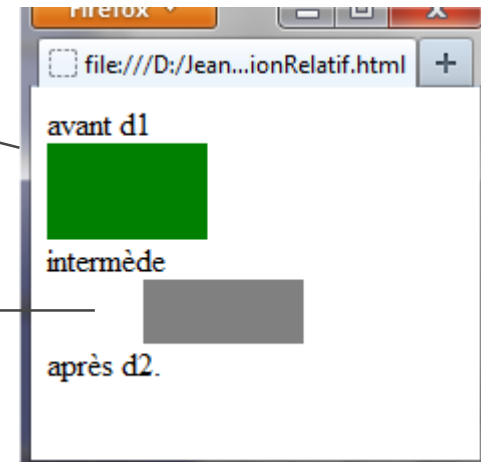


Mesures :

- valeur défaut : *auto*
- dimensions : *height*, *width*
- décalage : *left*, *top*

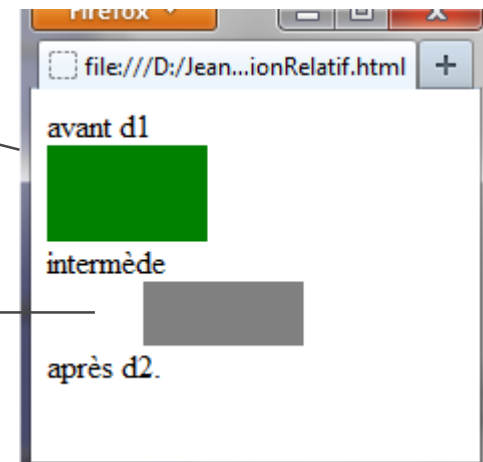
# Placement des conteneurs de type bloc en relatif

```
<html><body >
  avant d1
  <div position: relative
    style="width:5em; height:3em;
    top:0em; left:0em;">
    <div>
  intermède
  <div position: "relative"
    style="width:5em; height:2em;
    top:0em; left:3em;">
  </div>
  après d2.
</body></html>
```



# Exemple menu en relatif

```
<html><body >
avant d1
<div position: relative
    style="width:5em; height:3em;
    top:0em; left:0em;">
    </div>
intermède
<div position: "relative"
    style="width:5em; height:2em;
    top:0em; left:3em;">
    </div>
après d2.
</body></html>
```



# Comparaison des largeur occupées par les div

```

<html><body>
<div style="border:1pt blue solid"> normal (défaut) ! </div>
<div style="border:1pt blue solid; width:20em;" > normal + width</div>
<div style="position: absolute; border:1pt red solid
top: 20 px; left: 100 px;">absolu !</div>
<div style="position: absolute; border:1pt red solid; width:15em;
top: 20 px; left: 100 px;">absolu + width</div>
<div style="position: relative; top: 1em; left: 5em; border:1pt grey solid">relatif !</div>
<div style="position: relative; top: 1em; left: 5em; border:1pt grey solid; width:15em;">
relatif + width</div></body></html>

```

normal (défaut) !

normal + width

relatif !

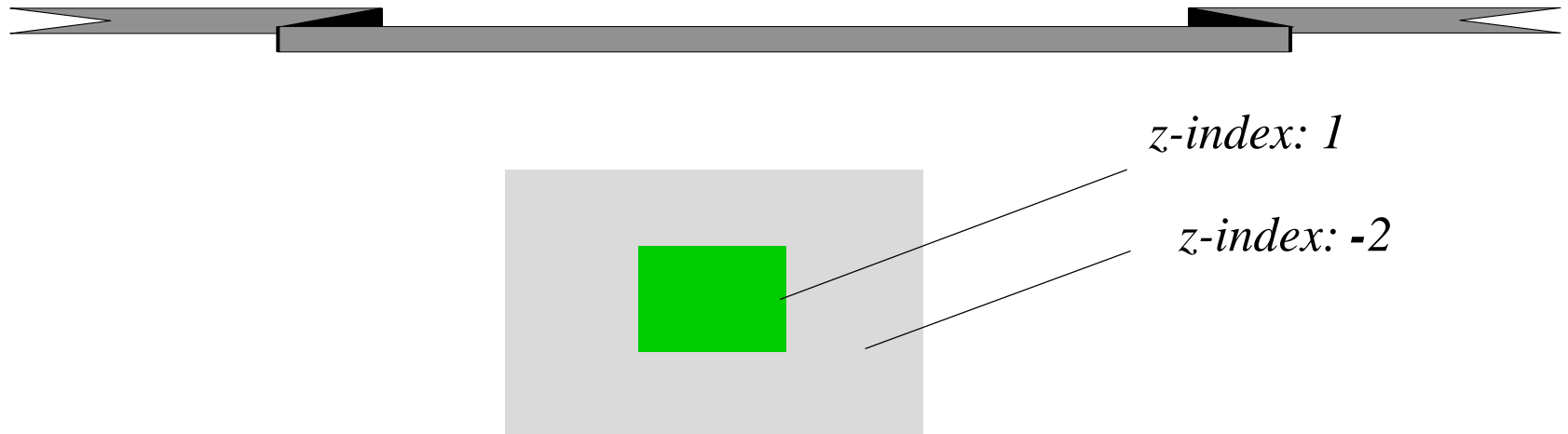
relatif + width

absolu !

absolu + width

*Sans dimensions spécifiées, les div simples ou relatives occupent l'espace à droite, les div-absolues collent au texte.*

# Autres propriétés de placement et d'affichage des blocs



**z-index** // Valeur à donner à chaque bloc en cas de chevauchement de blocs. donnant un entier / auto (le bloc de plus haut z-index est prioritaire)

**visibility** // visibilité du conteneur (emplacement réservé : diff de DISPLAY:none)  
☞ hidden / visible