

# [ présentation ]



[MCours.com](http://MCours.com)

HTML & CSS



Les feuilles de styles (CSS) existent depuis 1996.

Elles ne sont vraiment correctement supportées par les navigateurs que depuis 2000-2001.

C'est donc depuis cette date qu'elles sont devenues très présentes sur le Web.

Les feuilles de styles permettent de **simplifier** et de rendre plus **puissant** un codage HTML.



### Simplifier :

**Mise en forme à part du code HTML**, en général dans un fichier extérieur, mais ce n'est pas obligatoire. Le code HTML va se concentrer sur le contenu uniquement.

**Les styles s'appliquent à toutes les pages** du site. Plus de mise en forme page par page.

#### ▶ **Gain de temps**

développement plus rapide, code plus léger, et plus rapide à charger.

#### ▶ **Accélération de la maintenance :**

pour changer une couleur de titre, on n'a plus besoin de retoucher 50 articles un par un. On change seulement, en **une seule fois**, la définition du style "titre".

D'où une maintenance plus simple, et la possibilité de "*skiner*" un site.



### Améliorer :

Les codes de mise en forme du HTML sont moins puissants que ceux des feuilles de style (ou CSS version 2). D'où leur abandon progressif.

En combinant les CSS avec des programmes Javascript, PHP ou ASP, on peut modifier de façon dynamique l'apparence d'un site, bien plus puissamment qu'avec un code HTML mélangeant contenu et mise en forme.

### Rendre compatible :

Les navigateurs interprètent tous de la même façon le code CSS, réduisant considérablement les problèmes de compatibilités qui se posent souvent avec le code HTML (avec quelques réserves concernant le navigateur de Microsoft...)



Le langage CSS (*Cascading Style Sheets*) est utilisé pour **définir l'aspect** de votre site, comme par exemple la couleur du fond de la page ou le type de police.

Le CSS (ou feuille de style), c'est un **fichier externe** (exemple "style.css") lié aux pages html, dans lequel vous allez définir la forme de votre site.



## Pourquoi utiliser le CSS ?

### ▶ structure et présentation gérées séparément.

Vous pouvez modifier les couleurs, polices de caractères directement dans le fichier **CSS** et la structure (texte, images) directement dans la page HTML.

### ▶ portabilité du contenu facilitée

Une fois le contenu HTML correctement mis en place, il devient facile de changer l'apparence de son site.

Le code HTML est **réduit en taille** et en complexité.



## Exemple concret ?

**Je souhaite le fond des pages en gris.**

Si j'ai un site de 5 pages, en HTML je vais répéter **5 fois** :

```
<body bgcolor="#CCCCCC">
```

En CSS je vais mettre dans mon fichier (par exemple : *styles.css*)

```
body { background-color: #cccccc; }
```

Si un jour je souhaite modifier la couleur de toutes mes pages, avec le CSS, j'ai **un seul** fichier à modifier.



Le code CSS peut être placé à 3 endroits dans le site différentes dans la page.

- ▶ **en interne** : dans la page HTML
- ▶ **en externe** : dans un fichier indépendant (ex : « *styles.css* »)
- ▶ dans la **balise** HTML elle-même





## ① en interne

Placer le code CSS dans la page HTML, entre les deux balises **<head>** et **</head>**

```
<head>  
<title>titre de ma page</title>  
<style type="text/css">  
=> Ici le code CSS  
</style>  
</head>
```



## ② en externe (le plus recommandé)

Placer le code CSS dans un fichier indépendant (appelé souvent « *styles.css* »)

Le **lien** vers le fichier CSS doit aussi être inclus entre les deux balises **<head>** et **</head>**

```
<head>  
<title>titre de ma page</title>  
<link rel="stylesheet" type="text/css"  
href="styles.css" />  
</head>
```



### ③ dans une balise (le moins recommandé)

Placer le code CSS directement dans une balise HTML

```
<body style="background-color: #cccccc">
```



La syntaxe du CSS est très simple :

**sélecteur {propriété: valeur;}**

```
body {background-color: #cccccc;}
```

Chaque sélecteur (*ici body*) peut avoir **plusieurs propriétés** avec des valeurs indépendantes :

```
body {  
  background-color: #cccccc;  
  font-family: Trebuchet MS, Verdana, Arial;  
}
```

On remarquera un point virgule à la fin de chaque propriété.



## ■ Règles CSS : 4 types

### ▶ les sélecteurs HTML

Redéfinissent n'importe quelle balise HTML

ex : **p** s'appliquera à toutes les balises `<p>` de la page

### ▶ les classes

Règles librement choisies qu'on peut appliquer à n'importe quelle balise HTML

ex : **.gauche** s'appliquera à toutes les balises `<p class= "gauche">`

### ▶ les id (identifiants)

Même principe que les classes, mais ne peuvent s'appliquer qu'une seule fois dans une page

ex : **#header** s'appliquera à une seule balise `<div id= "header" >`

### ▶ les pseudo-classes de lien

Règles propres aux liens

# HTML & CSS

## ■ écriture HTML / CSS



```
<div id="cadre">
```

```
<p class="texte-bleu">
```

Le plus souvent on écrit du code CSS dans un fichier spécial ayant l'extension .css (contrairement aux fichiers HTML qui ont l'extension .html).</p>

```
<p class="texte-vert">
```

C'est la méthode la plus pratique et la plus flexible, on sépare la forme du contenu même au niveau des fichiers...</p>

```
</div>
```

# HTML

```
┌ #cadre
```

```
{background-color: yellow;  
padding: 10px;  
border: solid black 10px;}
```

```
┌ .texte-bleu
```

```
{color: blue;}
```

```
┌ .texte-vert
```

```
{color: green;}
```

```
┌ #cadre p
```

```
{margin-bottom: 5px;}
```

# CSS

Le plus souvent on écrit du code CSS dans un fichier spécial ayant l'extension .css (contrairement aux fichiers HTML qui ont l'extension .html).

C'est la méthode la plus pratique et la plus flexible, on sépare la forme du contenu même au niveau des fichiers...



**Déclaration du type de document qui indique au navigateur dans quel type et version de HTML la page a été écrite.**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Portfolio</title>
<style type="text/css">
  <!-- Code CSS -->
</style>
</head>
<body>
</body>
</html>
```

# [ sélecteurs css ]







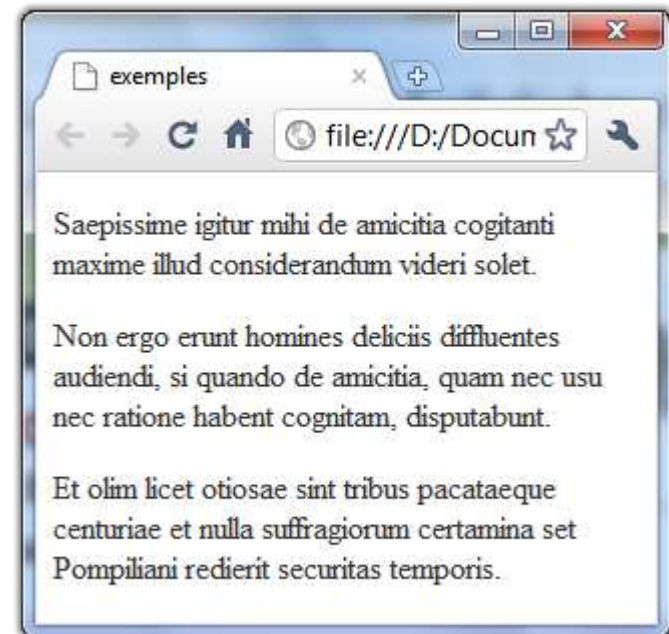
Dans une règle CSS, le **sélecteur** indique à quoi se rapportent un ensemble de propriétés.

### Sélecteur css de **BALISE** :

```
p {color: #FFFFFF;}
```

- Tous les paragraphes de la page auront une couleur de texte noire.

```
<p>
Saepissime igitur mihi de amicitia
cogitanti maxime illud considerandum
videri solet.</p>
<p>
Non ergo erunt homines deliciis
diffluentes audiendi, si quando de
amicitia, quam nec usu nec ratione
habent cognitam, disputabunt.</p>
<p>
Et olim licet otiosae sint tribus
pacataeque centuriae et nulla
suffragiorum certamina set Pompiliani
redierit securitas temporis.</p>
```





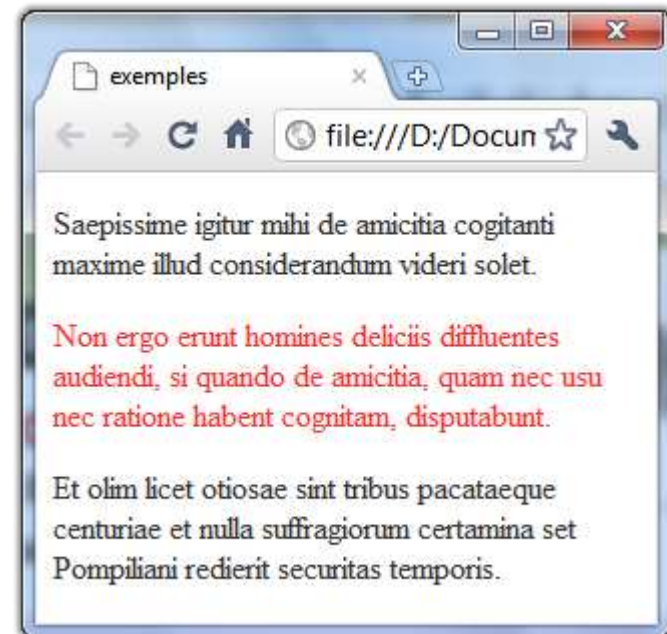
Dans une règle CSS, le **sélecteur** indique à quoi se rapportent un ensemble de propriétés.

### Sélecteur css de **CLASSE** :

```
.texterouge {color: #CC0000;}
```

► permet de sélectionner une balise ayant une **class "texterouge"**.

```
<p>
Saepissime igitur mihi de amicitia
cogitanti maxime illud considerandum
videri solet.</p>
<p class="texterouge">
Non ergo erunt homines deliciis
diffluentes audiendi, si quando de
amicitia, quam nec usu nec ratione
habent cognitam, disputabunt.</p>
<p>
Et olim licet otiosae sint tribus
pacataeque centuriae et nulla
suffragiorum certamina set Pompiliani
redierit securitas temporis.</p>
```





Dans une règle CSS, le **sélecteur** indique à quoi se rapportent un ensemble de propriétés.

Sélecteur css d' **ID** (identifiant) :

```
#textechapeau {background-color: #CCCCCC;}
```

► permet de sélectionner une balise ayant un id "*textechapeau*" .

```
<p id="textechapeau">  
Saepissime igitur mihi de amicitia  
cogitanti maxime illud considerandum  
videri solet.</p>  
<p class="texterouge">  
Non ergo erunt homines deliciis  
diffluentes audiendi, si quando de  
amicitia, quam nec usu nec ratione  
habent cognitam, disputabunt.</p>  
<p>  
Et olim licet otiosae sint tribus  
pacataeque centuriae et nulla  
suffragiorum certamina set Pompiliani  
redierit securitas temporis.</p>
```



Une seule  
valeur d'id  
par page



Un même **sélecteur** peut intervenir sur plusieurs propriétés.

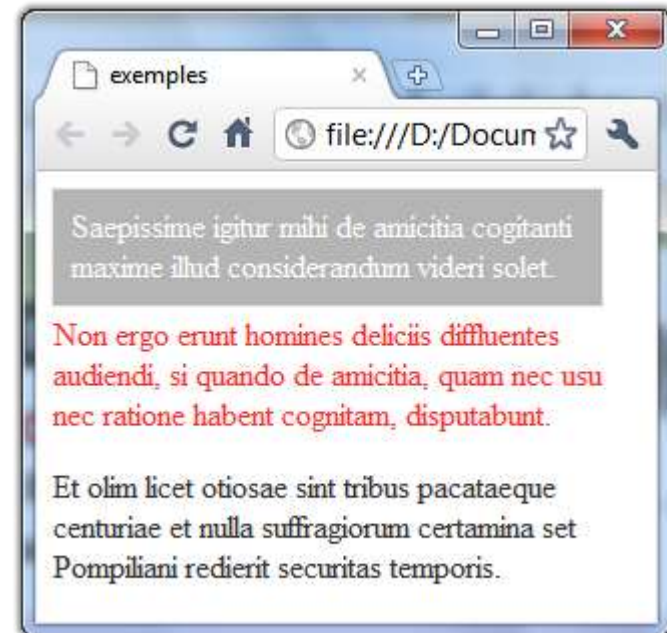
```
#textechapeau {  
    background-color: #CCCCCC;  
    color: #FFFFFF;  
    padding: 10px; }
```

← couleur d'arrière-plan

← couleur du texte

← marges intérieures

```
<p id="textechapeau">  
Saepissime igitur mihi de amicitia  
cogitanti maxime illud considerandum  
videri solet.</p>  
<p class="texterouge">  
Non ergo erunt homines deliciis  
diffluentes audiendi, si quando de  
amicitia, quam nec usu nec ratione  
habent cognitam, disputabunt.</p>  
<p>  
Et olim licet otiosae sint tribus  
pacataeque centuriae et nulla  
suffragiorum certamina set Pompiliani  
redierit securitas temporis.</p>
```

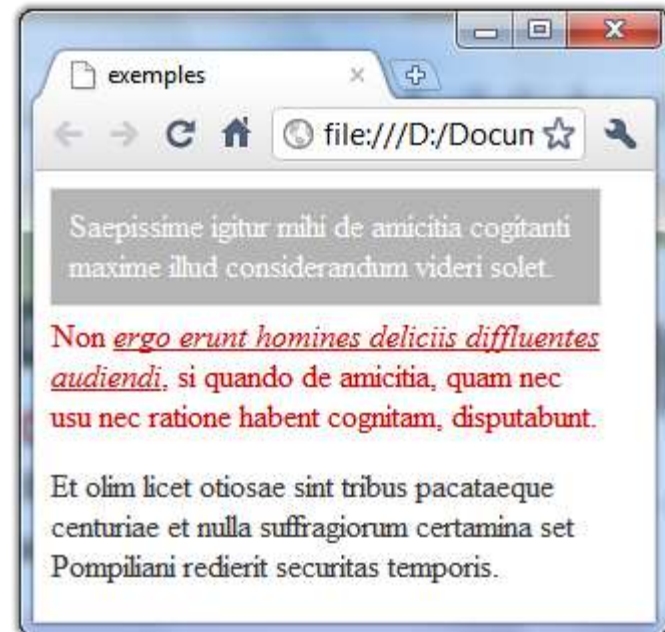




La balise **SPAN** est utilisée pour appliquer un style à un **contenu** (en général du texte).

```
.souligne_italic {  
    font-style: italic;  
    text-decoration: underline;}  
}
```

```
<p id="textechapeau">  
Saepissime igitur mihi de amicitia  
cogitanti maxime illud considerandum  
videri solet.</p>  
<p class="texterouge">  
Non <span class="souligne_italic">ergo  
erunt homines deliciis diffluentes  
audiendi</span>, si quando de  
amicitia, quam nec usu nec ratione  
habent cognitam, disputabunt.</p>  
<p>  
Et olim licet otiosae sint tribus  
pacataeque centuriae et nulla  
suffragiorum certamina set Pompiliani  
redierit securitas temporis.</p>
```



# [ mise en page ]





Le **flux d'un document** html correspond au comportement naturel d'affichage des éléments d'une page web.

Les éléments se succèdent dans l'ordre où ils sont déclarés dans le code HTML.

Les éléments s'affichent verticalement, commençant en haut de l'écran pour aller jusqu'en bas, et horizontalement de gauche à droite.

```
<p class="jaune">Une boîte jaune</p>
<p class="verte">Une boîte verte</p>
```





## ■ « bloc » ou « en-ligne »

Les éléments de type « **bloc** » s'affichent les uns en-dessous les autres

Les principaux éléments créant des boîtes **bloc** sont :

l'élément **div**

les titres **h1, h2, h3, h4, h5, h6**

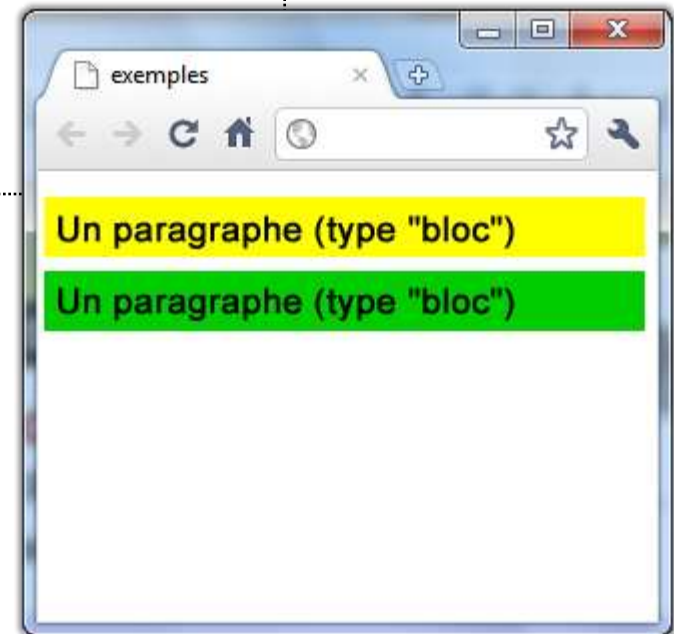
le paragraphe **p**

les listes et éléments de liste **ul, ol, li, dl, dd**

Les formulaires **form**

Les tableaux **table**

Le bloc de citation **blockquote**



```
<p class="jaune">Un paragraphe (type "bloc")</p>
<p class="verte">Un paragraphe (type "bloc")</p>
```





## ■ « bloc » ou « en-ligne »

Les éléments de type « **en-ligne** » s'affichent les uns à côté des autres

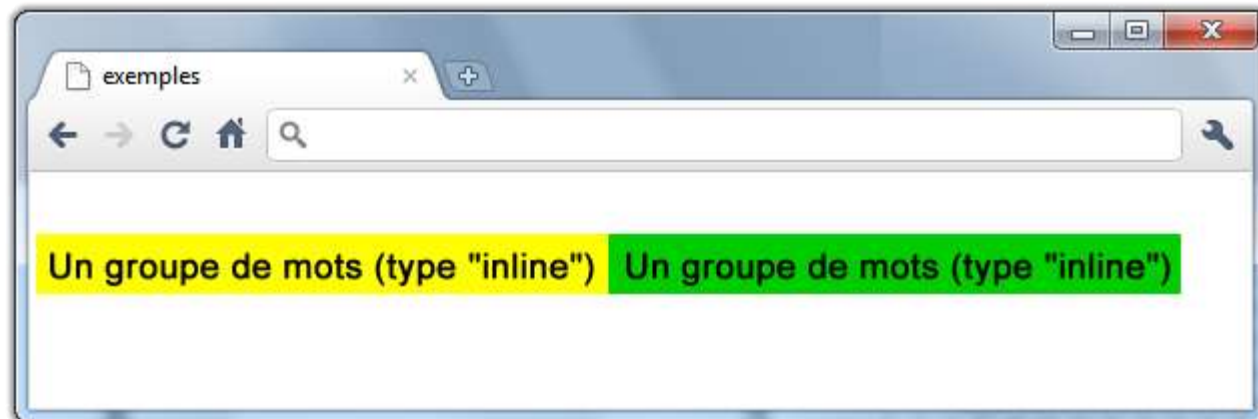
Les principaux éléments créant des boîtes **inline** sont :

l'élément **span**

le lien **a**

l'image **img**

les emphases **em** et **strong**



```
<span class="jaune">Un groupe de mots (type "inline")</span>  
<span class="verte">Un groupe de mots (type "inline")</span>
```



Le **flux** d'un document c'est donc le traitement linéaire du contenu de la page.

Alignement des éléments de type **en-ligne** :

**horizontal**

Alignement des éléments de type **bloc** :

**vertical**



## ■ « bloc » ou « en-ligne »

Une balise de type **bloc** peut s'identifier à une brique, à un conteneur qu'on va pouvoir manipuler aisément.

Il se différencie des éléments de type en ligne sur différents points :

- il occupe par défaut toute de la **largeur** de son conteneur
- il occupe par défaut la **hauteur** de son contenu
- il permet l'attribution de **marges** verticales
- il permet la modification de sa **hauteur** et **largeur**

C'est l'utilisation de ces propriétés qui va permettre la mise en page.

Tout élément peut être "reclassé" dans la famille opposée grâce à la propriété **display**.



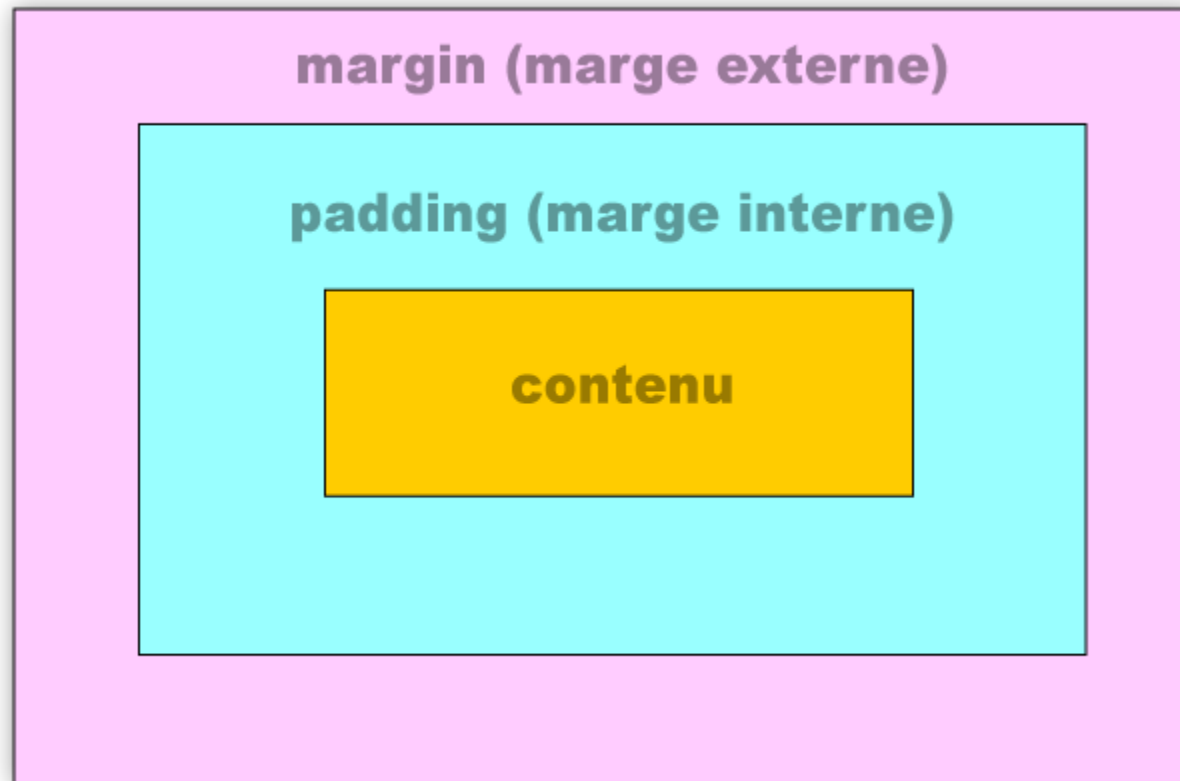
```
<p class="jaune">Un paragraphe (type "bloc")</p>
<p class="verte">Un autre paragraphe (type "bloc")</p>

<p><span class="rose">Un groupe de mots (type "inline")</span>
<span class="bleu">Un groupe de mots (type "inline")</span>
</p>
```





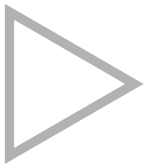
En CSS la taille d'un élément "bloc" se définit par **largeur, hauteur + marges intérieures + marges extérieures**





La propriété **position** va permettre de gérer le positionnement des éléments de type « bloc ».

C'est la propriété vers laquelle l'on se doit de se tourner en premier lieu dès que l'on considère une mise en page pour laquelle les possibilités du flux ne suffisent plus.



**3 types** de positionnement CSS :

- **relatif**
- **absolu**
- **flottant**



### ► positionnement **relatif**

La position relative (**position: relative**) permet de **décaler** un élément par rapport à sa position de référence : celle qu'il occupait dans le flux.

Les éléments qui le suivent et le précèdent **ne sont pas influencés** par ce décalage puisque l'élément est considéré être toujours à sa position initiale dans le flux .



### ► positionnement **relatif**

```
<p><span class="rose">Un groupe de mots</span>  
<span class="bleu">Un groupe de mots</span>  
<span class="vert">Un groupe de mots</span></div>
```



```
.rose {background-color: #FF99FF;}  
.bleu {  
  background-color: #66FFFF;  
  position: relative;  
  top: 25px;  
  left: 15px;  
}  
.vert {background-color: #99CC00;}
```





### ► positionnement **absolu**

La position absolue (**position: absolute**) permet de **positionner** un élément par rapport à son conteneur.

L'élément **sort totalement du flux** et les autres éléments l'ignorent.

Un élément positionné en absolu se réfère non pas à son parent direct, mais au premier ancêtre *positionné* qu'il rencontre.



### ► positionnement **absolu**

```
<p><span class="rose">Un groupe de mots</span>  
<span class="bleu">Un groupe de mots</span>  
<span class="vert">Un groupe de mots</span></div>
```



```
.rose {background-color: #FF99FF;}  
.bleu {  
  background-color: #66FFFF;  
  position: absolute;  
  top: 0px;  
  right: 0px;  
}  
.vert {background-color: #99CC00;}
```



### ► positionnement **flottant**

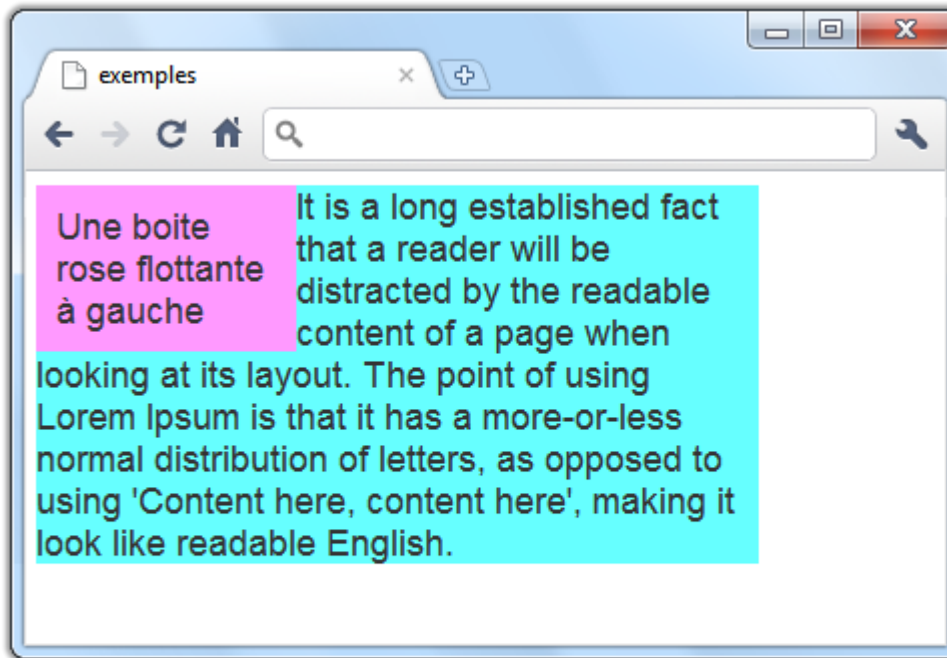
Le positionnement flottant (**position: float**) retire partiellement un élément du flux normal pour le placer à l'extrême gauche (**float: left**) ou l'extrême droite (**float: right**) dans son conteneur.

Les autres éléments du flux s'**écoulent autour**.

L'utilisation du positionnement flottant va permettre par exemple de faire une mise en page en plaçant des éléments de type « blocs » côte à côte.



### ► positionnement **flottant**



```
<p class="rose">Une boîte rose  
flottante à gauche</p>  
<p class="bleu">It is a long  
established fact that a reader will  
be distracted by the readable  
content of a page when looking at  
its layout..... </p>
```

```
.rose {  
    background-color: #FF99FF;  
    float: left;  
    width: 110px;  
    padding: 10px;  
    margin: 0px;  
}  
.bleu {  
    background-color: #66FFFF;  
}
```

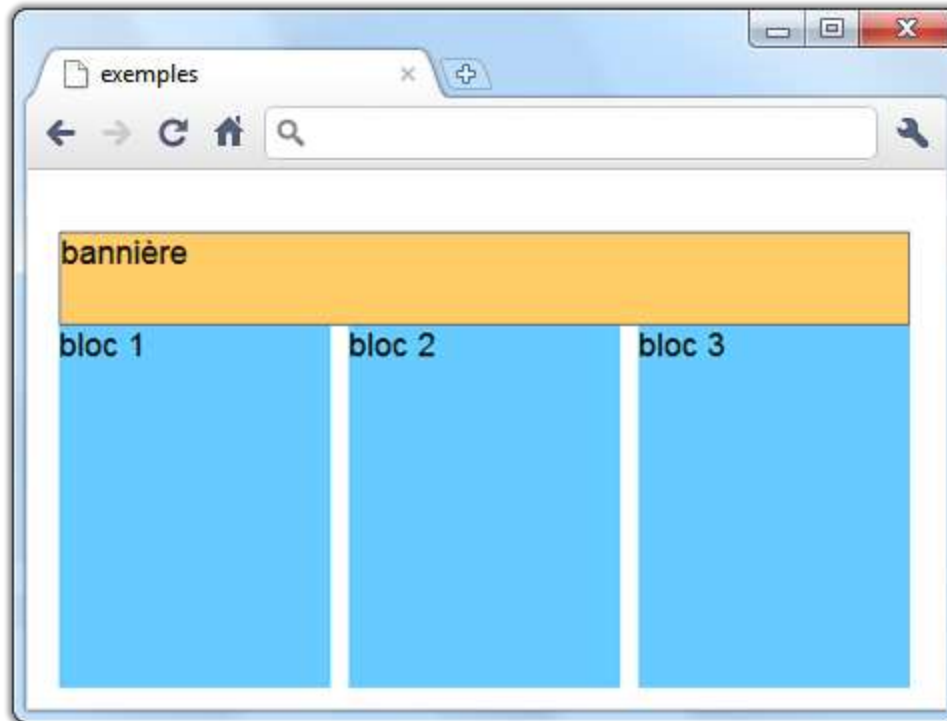
Le navigateur place d'abord la **boîte rose**, alignée ici sur le bord gauche de la page, puis reprend le cours normal du flux dans l'espace laissé libre à sa droite pour afficher la **boîte bleu**.



### ► positionnement **flottant**

```
<div id="banniere">bannière</div>
<div class="flottant-gauche">bloc 1</div>
<div class="flottant-gauche">bloc 2</div>
<div class="flottant-gauche">bloc 3</div>
```

Utilisation des **flottants** pour créer une mise en page avec 3 colonnes (la bannière est dans le flux, les 3 blocs sont flottants).



```
.flottant-gauche {
  float: left;
  width: 150px;
  margin-right: 10px;
}
#banniere {
  height: 50px;
  width: 468px;
}
```