



# Ses C

ot  
ti  
n  
e  
m  
e  
m

2<sup>e</sup> édition  
Raphaël Goetter



EYROLLES

# Généralités

## Balises en-ligne (XHTML et HTML strict)

Les éléments en ligne servent à définir la mise en forme de groupes de mots ou d'éléments au sein de blocs (liens, images, mise en exergue...). Ils peuvent s'imbriquer entre eux mais ne peuvent contenir d'éléments blocs.

`<a>` `<abbr>` `<acronym>` `<b>` `<bdo>` `<big>` `<br />` `<cite>` `<code>`  
`<dfn>` `<em>` `<i>` `<img>` `<input>` `<kbd>` `<label>` `<q>` `<samp>` `<select>`  
`<small>` `<span>` `<strong>` `<sub>` `<sup>` `<textarea>` `<t>` `<var>`

## Balises bloc et tables (XHTML et HTML strict)

Les éléments de type bloc définissent la mise en forme applicable à des paragraphes, entêtes, titres et autres zones de la page. Ils s'affichent les uns sous les autres. Sauf exception, un élément de type bloc peut contenir d'autres éléments blocs ou des éléments en-ligne.

`<address>` `<blockquote>` `<dd>` `<div>` `<dl>` `<dt>` `<fieldset>` `<form>`  
`<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>` `<hr />` `<li>` `<noscript>` `<ol>` `<p>`  
`<pre>` `<table>` `<tbody>` `<td>` `<tfoot>` `<th>` `<thead>` `<tr>` `<ul>`

## Balises mixtes, selon contenu (XHTML et HTML strict)

Ces éléments sont traités comme des blocs s'ils contiennent un élément de type bloc.

`<button>` `<del>` `<ins>` `<map>` `<object>` `<script>`

## Sélecteurs

**NOTE** Dans cette liste, les sélecteurs `div` et `span` sont cités à titre d'exemple. Ils représentent les balises HTML `<div>` et `<span>` mais peuvent être remplacés par d'autres balises.

- **\* {}** sélecteur universel désigne tous les éléments
- **div {}** sélecteur d'élément désigne la balise `<div>`
- **.maClasse {}** sélecteur de classe désigne les éléments ayant reçu la `class="maClasse"`
- **div.maClasse {}** sélecteur de classe désigne les éléments `<div>` ayant reçu la `class="maClasse"`
- **#monId {}** sélecteur d'identificateur désigne l'élément unique ayant reçu `id="monId"`
- **div#monId {}** sélecteur d'identificateur désigne l'élément `<div>` ayant reçu `id="monId"`
- **div span element {}** hiérarchie désigne les éléments contenus dans un élément `<span>` lui-même contenu dans des éléments `<div>`
- **div, span, element {}** sélecteur collectif désigne les éléments `<div>`, les éléments `<span>`, et les éléments `<element>`
- **div + span {}** sélecteur adjacent désigne tout élément `<span>` immédiatement précédé par un élément `<div>`
- **div > span {}** sélecteur d'enfants désigne tout élément `<span>` directement enfant de l'élément `<div>`
- **a:link | visited {}** pseudo-classes de liens désigne le lien `<a>` non visité (link) ou déjà visité (visited)
- **element:active | hover | focus {}** où *element* peut être toute balise ou classe

## Pseudo-classes dynamiques active, hover et focus

**active**: entre le moment où l'utilisateur presse le bouton de la souris et le relâche

**hover** : l'élément est survolé

**focus** : quand un élément reçoit l'attention, que ce soit au clavier ou d'autres formes d'entrées de texte

**IE** jusqu'à sa version 6 comprise, ne reconnaît la pseudo-classe `:hover` que si elle est appliquée sur le sélecteur de la balise `a`



# Généralités

- **div:lang(fr) {}** pseudo-classe lang désigne tout élément <div> qui emploie une langue désignée, ici le français.
- **div[class] {}** sélecteur par attributs désigne tout élément <div> qui possède un attribut class
- **div[class="maClasse"] {}** sélecteur par attributs désigne tout élément <div> qui possède un attribut class ayant pour valeur *maClasse*
- **div[class~="maClasse"] {}** sélecteurs par attributs désigne tout élément <div> qui possède un attribut class ayant plusieurs valeurs séparées par des espaces et dont l'une d'entre-elles est *maClasse*
- **div[class|="maClasse"] {}** sélecteur par attributs désigne tout élément <div> qui possède un attribut class ayant plusieurs valeurs séparées par des tirets et dont la première est *maClasse*.
- **div:first-child {}** pseudo-élément désigne le 1<sup>er</sup> enfant de l'élément <div>
- **div:first-line {}** pseudo-élément s'applique à la 1<sup>ère</sup> ligne de l'élément <div>
- **div:first-letter {}** pseudo-élément désigne la 1<sup>ère</sup> lettre de texte de l'élément <div>

**IE** IE6 ne reconnaît ni les sélecteurs d'enfants, ni ceux d'éléments adjacents, ni ceux par attributs (ce bogue sera peut-être corrigé dans IE 7).

## CSS 3

- div **p:last-child {}** sélectionne le dernier paragraphe des éléments div
- div **p:nth-child(2) {}** sélectionne le 2<sup>e</sup> paragraphe des éléments div
- div **:not([class="maClasse"]) {}** ne sélectionne pas les div dont l'attribut class est *maClasse*
- div **[class^="maClasse"] {}** sélectionne tout élément div dont la valeur de l'attribut class commence par *maClasse*
- div **[class\$="maClasse"] {}** sélectionne tout élément div dont la valeur de l'attribut class finit par *maClasse*
- div **[class\*="maClasse"] {}** sélectionne tout élément div dont la valeur de l'attribut class contient la chaîne *maClasse*

## Ordre des pseudo-classes de liens

1. :link      2. :visited      3. :hover (:focus)      4. :active

Moyen mnémotechnique : "LoVe Hate"

## Unités de valeurs

**0** → inutile de préciser une unité pour la valeur zéro (0)

**cm** → centimètres.

**em** → cadratin. Unité relative : 1em correspond à la taille de texte hauteur de la lettre majuscule M de l'élément parent.

**ex** → unité relative : 1ex correspond à la taille de texte largeur de la lettre x de l'élément parent.

**in** → inches.      **mm** → millimètres.

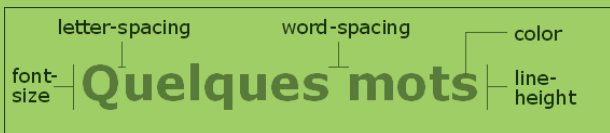
**pc** → picas.      **pt** → points picas conseillés pour le média imprimante.

**px** → pixels d'écran.      **IE** Unité figée sur IE pas d'agrandissement

0.75	2.12	9	21.2	54	pixel (Windows)
1	2.83	12	28.3	72	pixel (Mac)
1	2.83	12	28.3	72	point (pt)
0.36	1	4.23	10	25.4	mm
0.083	0.236	1	2.36	6	pica (pc)
0.035	0.1	0.42	1	2.54	cm
0.014	0.039	0.167	0.39	1	inch (in)



# Typographie



- **font-family :** *nom de police générique (sans-serif | monospace | etc.)* ; *font-family: "Trebuchet MS", arial, helvetica, sans-serif;*

Permet de choisir la police appliquée à l'élément sélectionné. possibilité d'indiquer plusieurs valeurs par ordre décroissant de choix. Les noms de polices contenant des espaces sont à placer entre guillemets.

- **font-size :** *taille pourcentage* ; *font-size: 1.2em;*

Détermine la taille de police de l'élément. Les unités recommandées sont em, ex et % afin de préserver la fluidité des textes.

- **color :** *couleur rgb() couleur hexa mot-clé (silver | fuschia | navy | etc.) transparent inherit* ; *color: #ff00ff;*

Applique une couleur de police au texte de l'élément sélectionné. Une palette de 16 couleurs de base (black, white, yellow, fuschia, silver, etc.) est prise en charge par l'ensemble des navigateurs graphiques et écrans. La palette élargie (256 couleurs) peut également être définie par des mots-clés (antiquewhite, chocolate, olive, salmon, etc.).

- **font-style :** *normal italic oblique inherit* ; *font-style: italic;*

Détermine l'orientation de police : italique (penché à droite) ou oblique (penché à gauche si la police est prévue pour admettre cette valeur).

- **font-weight :** *mots-clés (bold | bolder | normal | etc.) valeur numérique* ; *font-weight: bold;*

Choisit la graisse de police dans l'élément sélectionné. Les mots-clés (thin, thick) et les valeurs numériques (100, 200, etc.) ne sont pas reconnues par l'ensemble des navigateurs. Il est préférable de se restreindre aux seuls mots-clés bold, bolder et normal.

- **font-variant :** *normal small-caps* ; *font-variant: small-caps;*

Permet d'afficher la police d'un élément en petites majuscules.

## Ordre pour la syntaxe raccourcie

font : 1. font-style,  
2. font-variant,  
3. font-weight,  
4. font-size/line-height,  
5. font-family ;

exemple : `p{ font : bold 12px/24px verdana; }`



# Typographie

- **text-decoration :** none underline overline line-through ; *text-decoration: underline overline;*

Détermine le soulignement ou le surlignement de l'élément. Un texte peut être souligné, surligné, barré. Il est possible de cumuler les valeurs.

- **text-align :** left right center justify inherit ; *text-align: justify;*

Définit l'alignement du texte dans un élément.

- **text-transform :** none capitalize uppercase lowercase inherit ; *text-transform: uppercase;*

Mise en majuscule ou en minuscule du texte. La valeur capitalize affiche les premières lettres en majuscule. uppercase affiche toutes les lettres en majuscules et lowercase affiche toutes les lettres en minuscules.

- **text-indent :** valeur pourcentage ; *text-indent: 5px;*

Détermine la valeur de l'alinéa (retrait de la première ligne de texte).

- **line-height :** normal nombre longueur pourcentage inherit ; *line-height: 1em;*

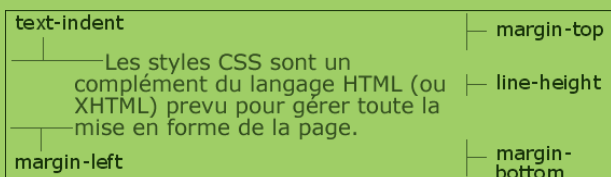
Précise la hauteur de ligne de texte. À ne pas confondre avec les marges entre chaque paragraphe.

- **letter-spacing :** normal longueur inherit ; *letter-spacing: -2px;*

Précise l'espace entre chaque lettre de texte. Une valeur négative est autorisée afin de rapprocher les lettres.

- **word-spacing :** normal longueur inherit ; *word-spacing: 3px;*

Précise l'espace entre chaque mot. Une valeur négative est autorisée afin de rapprocher les mots.



# Bordures

Applicables à tous les éléments, de type bloc comme en-ligne.

**NOTE** Il est préférable de ne pas les appliquer aux éléments en-ligne car ceux-ci, en règle générale, ne permettent pas d'être dimensionnés.

■ **border :** épaisseur  
style  
couleur  
inherit ; *border : 5px double black ;*

Peut contenir les différentes valeurs d'épaisseur (width), de style (style) et de couleur (color) et s'applique aux quatre bordures.

border est la forme condensée des propriétés border-top, border-right, border-bottom et border-left.

■ **border-width :** épaisseur  
inherit ; *border-width : 2em ;*

Spécifie la valeur de l'épaisseur de bordure du haut, de droite, du bas et de gauche d'un élément.

border-width est la forme condensée des propriétés border-top-width, border-right-width, border-bottom-width et border-left-width.

■ **border-color :** couleur  
transparent  
inherit ; *border-color : #ff9 ;*

Détermine la couleur des quatre bordures d'un élément.

border-color est la forme condensée des propriétés border-top-color, border-right-color, border-bottom-color et border-left-color.

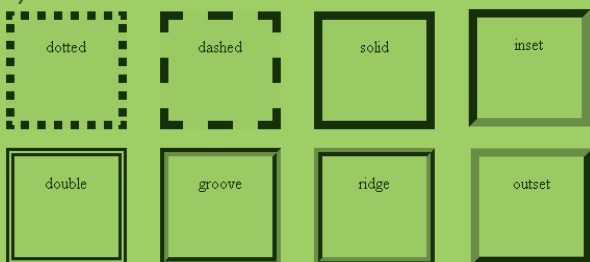
**NOTE** La valeur transparent conservera l'épaisseur de la bordure.

■ **border-style :** style (dashed | solid)  
inherit ; *border-style : groove ;*

S'applique aux quatre bordures de l'élément à la fois.

border-style est la forme condensée des propriétés border-top-style, border-right-style, border-bottom-style et border-left-style.

## Styles de bordures



**CSS 3** **border-image** Crée une bordure en image. Valeurs : url d'image, largeur, stretch, repeat, round. *border-image: url("border.png") 27 round stretch ;*

**border-radius** (non reconnue par IE8) Arrondit les coins de l'élément. Valeurs : rayon, rayon. *border-radius: 10px 5px ;*



# Arrière-plans

Applicables à tous les éléments, de type bloc comme en-ligne.

- **background** : *couleur*  
*url(image)* ; *background* : *white*  
*répétition* ; *url(image.png) no-repeat scroll*  
*attachement* ; *left top* ;  
*position*  
*inherit*

background est la forme condensée des propriétés expliquées ci-après background-color, background-image, background-repeat, background-attachment et background-position.

- **background-color** : *couleur*  
*inherit* ; *background-color* : *#f9f9f9* ;

Détermine la couleur de fond d'un élément. Peut se cumuler avec une image d'arrière-plan (par exemple une illustration en partie transparente).

- **background-image** : *url(image)*  
*inherit* ; *background-image* : *url(image.png)* ;

Spécifie une image d'arrière-plan qui vient se superposer à la couleur de fond de l'élément.

**NOTE** Éviter lorsque possible les guillemets dans l'URL car ils peuvent poser des problèmes de compatibilité avec d'anciens navigateurs.

- **background-repeat** : *repeat*  
*repeat-x*  
*repeat-y*  
*no-repeat*  
*inherit* ; *background-repeat* : *repeat-x* ;

Spécifie la répétition de l'image d'arrière-plan (repeat est la valeur par défaut). La valeur repeat-y spécifie que l'image ne doit se répéter que dans le sens vertical, repeat-x dans le sens horizontal. La valeur no-repeat empêche la répétition de l'image.

- **background-attachment** : *scroll*  
*fixed*  
*inherit* ; *background-attachment* : *fixed* ;

Indique si l'image est fixée par rapport au document, c'est-à-dire si elle défile ou non selon la zone de visualisation et l'ascenseur. Un arrière-plan dont la valeur est fixed ne se déplace pas lorsque l'ascenseur fait défiler l'élément.

- **background-position** : *pourcentage*  
*longueur*  
*mots-clés*  
*inherit* ; *background-position* : *50% 20%* ;

Indique la position initiale de l'image d'arrière-plan. Lorsqu'une seule valeur est indiquée (ex : background-position : 5px ;), celle-ci correspond à la position horizontale, l'image étant verticalement centrée. Lorsque deux valeurs sont indiquées (ex : background-position : right top ;), la première correspond à la position horizontale et la seconde correspond à la position verticale. Ces valeurs peuvent être exprimées par un nombre (ex : 2cm), par un pourcentage (ex : 25%) ou par des mots-clés (top, right, bottom, left et center). Il est possible de combiner des valeurs en pourcentage et en longueur (ex : background-position : 40% 20 px ;), mais pas de combiner des mots-clés avec d'autres valeurs.

**CSS 3** **opacity** (non reconnue par IE8) Indice d'opacité d'un élément.

Exemple : opacity: 0.7 ;



# Tableaux

■ **caption-side** : top  
bottom  
left  
right  
inherit ; *caption-side : top ;*

Cette propriété indique où va se placer la boîte de légende (balise <caption>) par rapport à la table.

■ **table-layout** : auto  
fixed  
inherit ; *table-layout : auto ;*

La propriété table-layout contrôle l'algorithme employé pour la disposition des cellules, des rangées et des colonnes de la table.

Cf. <http://www.yoyodesign.org/doc/w3c/css2/tables.html#width-layout>

■ **border-collapse** : collapse  
separate  
inherit ; *border-collapse : collapse ;*

Détermine le choix du modèle de bordures d'un tableau : bordures fusionnées (collapse) ou bordures séparées (separate).

**border-collapse : separate**

Cellule	Cellule	Cellule
Cellule	Cellule	Cellule
Cellule	Cellule	Cellule

**border-collapse : collapse**

Cellule	Cellule	Cellule
Cellule	Cellule	Cellule
Cellule	Cellule	Cellule

■ **border-spacing** : longueur  
inherit ; *border-spacing : 0.5em ;*

Détermine la distance qui sépare les bordures des cellules adjacentes.

■ **empty-cells** : show  
hide  
inherit ; *empty-cells : show ;*

Indique si les bordures doivent être affichées (show) ou non (hide) pour les cellules vides de contenu.

■ **vertical-align** : baseline  
top  
bottom  
middle  
inherit ; *vertical-align : middle ;*

Appliquée à une cellule de tableau (<td>) ou à un élément de type table-cell, cette propriété définit l'alignement vertical du contenu dans l'élément.





# Positionnement 1

## Propriétés de positionnement

■ <b>display :</b>	inline block none list-item run-in compact marker table inline-table	table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption inherit	;	<i>div { display : inline ; }</i> <i>span { display : block ; }</i>
--------------------	--	--	---	--

Permet de donner à un élément appartenant à une certaine structure (inline, bloc...) les caractéristiques d'affichage d'une autre structure. On pourra ainsi doter un élément en ligne des caractéristiques de hauteur normalement dévolues à un élément de type bloc.

Valeurs les plus couramment employées : block et inline qui affectent la structure apparente des éléments, et none qui permet de ne pas prendre en compte l'élément.

**IE** Les versions inférieures à IE8 ne supportent pas toutes les valeurs standards de la propriété display, notamment table-cell.

■ <b>visibility :</b>	visible hidden collapse inherit	;	<i>visibility : hidden;</i>
-----------------------	--	---	-----------------------------

Affiche ou masque un élément. La valeur hidden réserve un espace vide de la taille qu'aurait eu l'élément s'il était représenté.

■ <b>position :</b>	static absolute relative fixed	;	<i>position : absolute;</i>
---------------------	---	---	-----------------------------

Détermine l'emplacement de l'élément, et lui affecte des caractéristiques de type bloc. La valeur par défaut, static, correspond au positionnement par défaut dans le flux. La valeur relative correspond à un décalage dans le flux. Les valeurs absolute et fixed correspondent à un positionnement *hors du flux*.

**IE** IE6 ne supporte pas la valeur fixed de la propriété position.

■ <b>top / right / bottom / left :</b>	distance pourcentage auto inherit	;	<i>top : 2em;</i>
--	--	---	-------------------

Contrôle la distance (haute, droite, basse ou gauche) entre l'élément et son éventuel conteneur, à défaut la page. Ne s'appliquent qu'aux éléments ayant reçu une déclaration position différente de static.

■ <b>z-index :</b>	auto entier inherit	;	<i>z-index : 10;</i>
--------------------	---------------------------	---	----------------------

Détermine et modifie l'ordre d'empilement des éléments. Cette propriété ne s'applique qu'aux éléments ayant reçu une déclaration position différente de static.

■ <b>float :</b>	left right none inherit	;	<i>float : left;</i>
------------------	----------------------------------	---	----------------------

Fait passer l'élément en mode de positionnement flottant et détermine de quel côté du parent celui-ci doit s'afficher (voir section Positionnement 2). À noter qu'un positionnement flottant ne permet pas de centrer un élément.

Liens utiles : [www.openweb.eu.org](http://www.openweb.eu.org) : référence francophone sur les standards du Web



# Positionnement 1

■ **clear :** none  
left  
right  
both  
inherit ; *clear: both;*

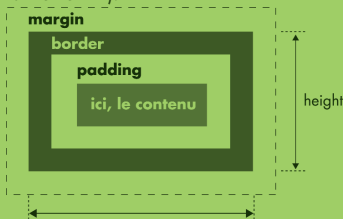
Détermine si l'élément peut se trouver sur la même bande horizontale qu'un élément flottant.

Ne s'applique que si la page contient des éléments en mode de positionnement flottant (voir section Positionnement 2).

## Propriétés de dimensionnement

Ne s'appliquent qu'aux éléments de type bloc sauf exceptions telles que <img>.

■ **width :** largeur  
pourcentage  
auto  
inherit ; *width : 30%;*



Spécifie la largeur du contenu de l'élément. Selon le modèle standard, la largeur de la boîte de l'élément correspond à width + padding + border.

**IE** Selon le modèle Microsoft, appliqué quand le document (X)HTML n'est pas valide, la largeur de la boîte correspond uniquement à la valeur de la propriété width.

■ **height :** hauteur  
pourcentage  
auto  
inherit ; *height : 250px;*

Spécifie la hauteur du contenu de l'élément. Les modèles sont identiques à ceux de la propriété width.

■ **min-width / max-width :** largeur  
pourcentage  
none  
inherit ; *max-width : 95%;*

Spécifie la largeur minimale ou maximale de l'élément. Les valeurs doivent toujours être positives.

**IE** ignore ces propriétés – jusqu'à sa version 6 incluse.

■ **min-height / max-height :** longueur  
pourcentage  
none  
inherit ; *max-height : 50px;*

Spécifie la largeur minimale ou maximale de l'élément. Les valeurs doivent toujours être positives.

**IE** ignore ces propriétés – jusqu'à sa version 6 incluse.

■ **margin :** distance  
inherit ; *margin : 2px 3px 4px 0;*

Spécifie les espacements externes à l'élément (entre l'élément et ses frères ou les bords du parent).

L'exemple correspond aux marges haute (2px), droite (3px), basse (4px) et gauche (0).

**ORDRE POUR LA SYNTAXE RACCOURCIE** margin est le raccourci pour les propriétés margin-top, margin-right, margin-bottom et margin-left.



# Positionnement 1

■ **padding** : *distance* *inherit* ; **padding** : 2px 3px 4px 0;

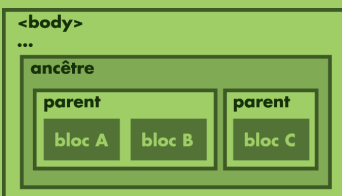
Spécifie les espacements internes à l'élément (entre le contenu et la bordure). L'exemple correspond aux marges haute (2px), droite (3px), basse (4px) et gauche (0).

**ORDRE POUR LA SYNTAXE RACCOURCIE** padding est le raccourci pour les propriétés padding-top, padding-right, padding-bottom et padding-left.

**CSS 3** **columns** Nombre de colonnes et leur largeurs respectives. Valeurs : largeur, nombre. Exemple : `columns : auto 3 ;`

## Positionnement 2

### Hierarchie des éléments



- A et B sont frères car ont un parent commun
- A, B et C ont le même ancêtre mais des parents différents
- Tous les éléments sont contenus dans l'élément `<body>`

### Positionnement en flux

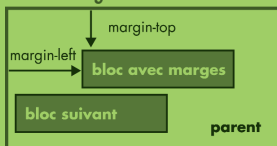
Lorsque aucun positionnement n'est spécifié, les éléments se placent dans la page selon :

1. l'ordre dans lequel ils apparaissent dans la partie HTML : la première balise du code HTML sera la première à se placer sur la page et ainsi de suite. D'où une notion de flux.
2. leur structure (bloc ou en-ligne) : un élément de type bloc se place sous l'élément précédent ; un élément de type en-ligne se place à côté de l'élément précédent.
3. les marges internes (padding) et externes (margin) permettent d'espacer et de positionner chaque élément par rapport aux autres tout en restant dans le flux.
4. Le positionnement relatif, associé aux propriétés top, right, bottom et left permet de décaler un élément tout en restant dans le flux. Les éléments frères suivants ne seront pas influencés par ce décalage.

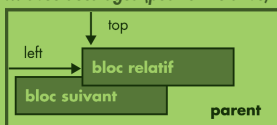
#### Positionnement en flux



#### ... avec marges



#### ... avec décalages (position relative)



## Positionnement 2

### Positionnement absolu

`position : absolute ;`      `p { position : absolute ; top : 2em ; left : 0 ; }`

S'opère lorsque l'élément se voit appliquer la règle `position : absolute`. L'élément est alors dit positionné.

- l'élément se place par rapport au dernier parent ou ancêtre positionné, ou par défaut par rapport au bords du document
- l'élément sort du flux (son positionnement sera le même quelle que soit la place de la balise dans le code HTML). Il n'affecte pas les autres éléments en flux.
- l'élément est en général placé par rapport aux valeurs données aux propriétés `top`, `right`, `bottom` et/ou `left`

### Positionnement fixé

`position : fixed ;`      `#menu { position : fixed ; top : 20px ; left : 10px ; }`

S'opère lorsque l'élément se voit appliquer la règle `position : fixed`. La seule différence avec le positionnement absolu tient au fait que l'élément fixé reste en place même si la page défile avec l'ascenseur.

IE 6 ne reconnaît pas ce mode de positionnement.

### Positionnement flottant

`float : left | right | none | inherit ;`      `img { float : left ; }`

Permet de positionner un élément à gauche ou à droite dans son parent. Le reste du contenu du parent occupera alors l'espace laissé libre en commençant par s'écouler autour du flottant.

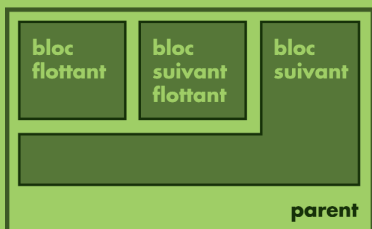


### Gérer les flottements avec la propriété clear

`clear : left | right | both | none | inherit ;`      `div { clear : both ; }`

Empêche un élément de se trouver sur la même ligne qu'un élément flottant. L'élément se placera automatiquement et intégralement sous l'élément flottant.

Gestion des  
avec "clear"



# Médias (imprimante, vocal...)

## Types de médias

all, print (imprimante), screen (écran), projection (présentations projetées), braille (tablettes brailles), embossed (impression braille), aural (propriétés auditives CSS2.0), speech (propriétés auditives CSS2.1), handheld (assistants numériques), tv (Web-TV).

**NOTE** Les propriétés CSS de ces divers médias – hormis screen et print – sont très mal implémentées par les supports existants (navigateurs, synthèses vocales, etc.). Le média handheld prévu pour les supports mobiles est encore peu pris en charge par les navigateurs natifs de téléphones.

## Déterminer le média

Exemples employant la balise <link>

```
<link rel="stylesheet" type="text/css" href="styles.css" media="screen" />  
<link rel="stylesheet" type="text/css" href="impression.css" media="print" />
```

Exemples employant la propriété CSS2 @import

```
@import url(styles.css) screen;  
@import url(impression.css) print;
```

## Propriétés de sortie imprimante (print)

■ @page : { propriétés ; } ; @page { size : landscape; }

La propriété CSS2 @page permet d'établir les règles de mise en page avec les propriétés décrites ci-dessous.

■ size :   
longueur  
auto  
portrait  
landscape  
inherit ;

 ; @page { size : 21.0cm 29.7cm; }

Définit la taille et l'orientation (portrait ou paysage) de la page imprimée. La taille comprend dans l'ordre la largeur et la hauteur séparées par un espace.

■ marks :   
crop  
cross  
none  
inherit ;

 ; @page { marks : cross; }

Hirondelles (repères imprimés sur la page pour en faciliter la coupe).

■ page-break-before :   
auto  
always  
avoid  
left  
right  
inherit ;

 ; @page { page-break-before : always; }

Impose un saut de page avant (before), après (after) ou dans (inside) l'élément désigné.

■ orphans :   
entier  
inherit ;

 ; @page { orphans : 3; }

Évite les orphelines (lignes isolées sur leur page car ne tenant pas sur la page précédente) en définissant un nombre minimum de lignes par page imprimée.

■ widows :   
entier  
inherit ;

 ; @page { widows : 3; }

Évite les lignes veuves en précisant un nombre de lignes avant le saut de page.



## Propriétés du média vocal

Le média aural (devenu `speech` depuis CSS 2.1) permet de spécifier des propriétés de restitution par synthèse vocale : type de voix (`voice-family`), origine du son (`azimuth`), direction (`elevation`), volume (`volume`), pauses (`pause-before`, `pause-after`), timbre (`pitch`) et portée de la voix (`richness`), prononciation (`speak`, `speak-numeral`, `speak-punctuation`), etc.

**ATTENTION** La plupart des navigateurs, excepté Opéra 8, ne reconnaissent pas la majorité de ces propriétés.

## Listes

■ **list-style** : 

<code>type</code>	;	<code>list-style : circle inside ;</code>
<code>position</code>		
<code>url(image)</code>		
<code>inherit</code>		

Cette propriété est la forme condensée des propriétés individuelles `list-style-type`, `list-style-position` et `list-style-image`.

Il est possible de combiner la valeur `type` et `url(image)`. Dans ce cas, le type choisi sera affiché en cas de non disponibilité de l'image.

■ **list-style-image** : 

<code>none</code>	;	<code>list-style-image : url(puce.png) ;</code>
<code>url(image)</code>		
<code>inherit</code>		

Désigne l'image qui sera affichée comme marqueur (puce) d'élément de liste.

■ **list-style-type** : 

<code>disc</code>	<code>lower-alpha</code>	<code>hiragana</code>	;	<code>list-style-type : circle ;</code>
<code>circle</code>	<code>lower-latin</code>	<code>katakana</code>		
<code>square</code>	<code>upper-alpha</code>	<code>hiragana-iroha</code>		
<code>decimal</code>	<code>upper-latin</code>	<code>katakana-iroha</code>		
<code>decimal-leading-zero</code>	<code>hebrew</code>	<code>none</code>		
<code>lower-roman</code>	<code>armenian</code>	<code>inherit</code>		
<code>upper-roman</code>	<code>georgian</code>			
<code>lower-greek</code>	<code>cjk-ideographic</code>			

Définit l'aspect du marqueur (puce) d'élément de liste. Lorsque la valeur est `none`, aucune puce n'apparaît. Lorsque la propriété `list-style-image` est définie, cette dernière remplace la propriété `list-style-type`.

■ **list-style-position** : 

<code>inside</code>	;	<code>list-style-position : inside ;</code>
<code>outside</code>		
<code>inherit</code>		

Indique l'emplacement de la puce par rapport à la liste. La valeur `inside` place la puce dans la liste, le reste de celle-ci s'écoulant autour de cette puce (équivalent à un positionnement flottant à gauche). La valeur `outside` exclut la puce du contenu de liste.

## Chez le même éditeur...

**CSS2 - Pratique du design web, 3<sup>e</sup> ed.**

R. GOETTER

**Mémento XHTML, 2<sup>e</sup> ed.** R. GOETTER

**Mémento Calc.** S. GAUTIER

**Mémento Excel.** M. BERGAME

**Mémento Firefox et Thunderbird.** M. GREY

**Mémento Mac OS X.** G. GETE

**Mémento Unix/Linux.** I. HURBAIN

**Mémento LaTeX.** C. JACQUET

Code éditeur : G12542

ISBN : 978-2-212-12542-9





**Merci  
d'avoir acheté  
ce memento.**

**Vous pouvez le retrouver  
ainsi que toute la collection  
en version papier dans toutes  
les bonnes librairies !**

**Découvrez toute la collection memento  
sur : [www.eyrolles.com](http://www.eyrolles.com)**



Rodolphe Rimelé  
**Mémento HTML5**  
978-2-212-13420-9



Raphaël Goetter  
**Mémento CSS 3**  
978-2-212-13281-6



Elie Sloïm , Opquast  
*Memento Sites web -  
Les bonnes pratiques*  
978-2-212-12802-4



Jean-François Lépine  
*Memento -  
Industrialisation PHP 5*  
978-2-212-13480-3

## Chez le même éditeur :



Raphaël Goetter  
*CSS avancées*  
978-2-212-13405-6



Rodolphe Rimelé  
*HTML5*  
978-2-212-12982-3

Tous les livres informatiques sont sur  
[www.editions-eyrolles.com](http://www.editions-eyrolles.com)

Et pour toutes les nouveautés numériques,  
retrouvez-nous sur Twitter et Facebook

 @ebookEyrolles  
 EbooksEyrolles