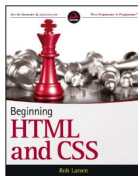


- 1 Introduction
- 2 Sélecteurs CSS
- 3 Propriétés CSS
- 4 Positionnement
- 5 Modules CSS3

- 1 Introduction
- 2 Sélecteurs CSS
- 3 Propriétés CSS
- 4 Positionnement
- 5 Modules CSS3



Rob Larsen
Beginning HTML and CSS,
Wrox, 2013



Matthew MacDonald
HTML5, the missing manual,
2nd Edition, O'reilly. 2014



John Duckett.
Beginning Web Programming with HTML,
XHTML, and CSS,
2nd Edition, Wrox. 2008.

Validation CSS

- W3C Validator at <http://jigsaw.w3.org/css-validator/>

Spécifications/Ressources

- alsacreation at <http://www.alsacreation.com>
- sitepoint at <http://www.sitepoint.com>

CSS3

- caniuse at <http://caniuse.com>
- moderniz at <http://modernizr.com>
- CSS3please at <http://css3please.com>

HTML permet de structurer le contenu d'un document.

CSS (Cascading Style Sheets) permet la mise en forme d'un document, en jouant par exemple avec :

- la couleur et la taille des polices,
- les bordures,
- l'espacement entre les éléments,
- ...

Plusieurs versions de CSS:

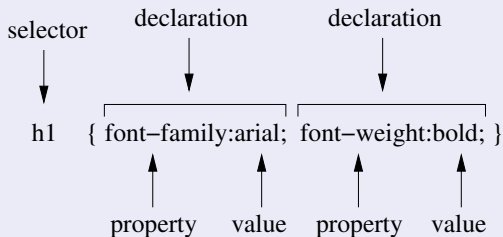
- CSS1 (CSS level 1), la première spécification CSS à devenir une Recommandation W3C officielle,
- CSS2 (CSS level 2), publiée comme Recommandation en Mai 1998,
- CSS3, en développement depuis Décembre 2005.

Attention

Ne pas utiliser des éléments et/ou attributs HTML pour contrôler l'apparence d'un document.

Les règles CSS sont associées à des éléments contenus dans un document. Ces règles déterminent la façon dont ces éléments seront affichés. Chaque règle est composée de deux parties :

- *un sélecteur* qui indique à quel(s) élément(s) (sous forme d'une liste dont le séparateur est la virgule) la déclaration s'applique ;
- *une déclaration* qui indique (entre accolades) comment les éléments doivent être rendus ("stylés") sous forme d'une liste de couples propriété:valeur avec le point-virgule comme séparateur
 - ▶ la propriété concerne les éléments sélectionnés
 - ▶ la valeur est une spécification pour la propriété



Exemple

```
h1, h2, h3 {  
    font-weight:bold;  
    font-family:arial, verdane, sans-serif;  
    color:#000000;  
    background-color:#FFFFFF;  
}
```

Dans cet exemple, le contenu de tous les éléments <h1>, <h2>, <h3> présents dans le document seront :

- écrits en caractère Arial gras (ou Verdane gras, si Arial n'est pas installé, ou sans-serif gras si Arial et Verdane ne sont pas installés) ;
- écrits en noir sur fond blanc.

Remarque

Toujours placer un ; après chaque couple propriété:valeur même si il s'agit du dernier couple de la liste.

Associer des règles CSS à une page HTML

Il existe 3 façons d'associer des règles CSS à un document HTML:

- utiliser une feuille de style externe (règles déclarées dans un fichier externe)
- utiliser une feuille de style interne (règles déclarées dans un élément `<style>` placé dans l'en-tête du document)
- utiliser des règles de style en ligne (via l'attribut `style`) au niveau de différents éléments du document

Remarque

Pour une même propriété, une règle en ligne est prioritaire à une règle dans une feuille de style interne, elle-même prioritaire à une règle dans une feuille de style externe.

Pour utiliser un fichier séparé contenant les différentes règles, vous devez utiliser un élément `<link>` sans contenu au sein de l'élément `<head>` du document. Cet élément `<link>` doit être accompagné de deux attributs :

- `rel` dont la valeur doit être `"stylesheet"`;
- `href` dont la valeur doit être une URL absolue ou relative correspondant au fichier séparé.

Exemple

Supposons que nous ayons un fichier *mySheet.css* dans un sous-répertoire *css*. Dans un document HTML, nous pouvons l'utiliser comme suit :

```
<head>
...
<link rel="stylesheet" href="css/mySheet.css" />
...
</head>
```

Feuille de style externe

L'élément `<link>` accepte aussi l'attribut `media`. Cet attribut nous permet d'associer différentes feuilles de style à différents supports d'affichage. Certaines des valeurs possibles sont :

Valeur	Utilisation
screen	écran d'ordinateur (valeur par défaut)
tv	écran de télévision (faible résolution, scrolling limité)
print	imprimante
aural	lecteur d'écran

Exemple

```
<link rel="stylesheet" href="theme.css" />
<link rel="stylesheet" href="print.css" media="print"/>
```

Remarque

Les commentaires dans les fichiers CSS sont placés entre `/*` et `*/`

Feuille de style interne

Lorsque les règles de style sont contenues dans un élément `<style>` contenu lui-même dans l'élément `<head>`, on parle de feuille de style interne.

Exemple

```
<link rel="stylesheet" href="mySheet.css" />
<style>
  h1 { color:red; }
</style>
```

Remarque

Une feuille de style interne peut être appropriée pour inclure quelques règles complémentaires qui ne s'appliquent pas à d'autres documents partageant le même style (général).

Lorsque des attributs `style` sont utilisés sur des éléments HTML, on parle de règles de style en ligne. Le style est directement appliqué à l'élément portant l'attribut `style`.

Exemple

```
<td style="font-family:courier; padding:5px;"> 10 </td>  
<td style="border-style:solid; border-width:1px;"> 20 </td>
```

Attention

Éviter d'utiliser des règles de style en ligne.

Plan

- 1 Introduction
- 2 **Sélecteurs CSS**
- 3 Propriétés CSS
- 4 Positionnement
- 5 Modules CSS3

Sélecteurs simples et universel

- un sélecteur de *type simple* spécifie (le nom d')un seul type d'élément ; e.g.,
`p { color:red; }`
- un sélecteur de *type* spécifie plusieurs types d'élément dans une liste (le séparateur est la virgule) ; la règle s'applique à tous ; e.g.,
`h1, h2, p { color:red; }`
- le sélecteur *universel* `*` s'applique à tous les éléments du document ; e.g.,
`* { color:red; }`

Attention

Ne pas confondre

```
h1, h2, p { color:red; }
```

avec

```
h1 h2 p { color:red; }
```

L'espace a un sens tout à fait différent de la virgule (voir plus loin).

Sélecteurs de classe et d'id

- un sélecteur de *classe* s'applique à tous les éléments portant un attribut `class` dont la valeur est celle spécifiée dans le sélecteur (précédée par `.`) ; e.g.,
`.rouge { color:red; }`
- un sélecteur de classe peut être combinée avec un sélecteur de type pour limiter son application aux éléments du type donné ; e.g.,
`p.rouge { color:red; }`
- un sélecteur d'id s'applique uniquement à l'élément portant l'attribut `id` avec la valeur spécifiée dans le sélecteur (précédée par `#`) ; e.g.,
`#idTable1 { color:red; }`

Remarque

Il est possible d'enchaîner les sélecteurs de classe (sans espace). Par exemple :

```
p.code.important { color:red; font-weight:bold; }
```

```
...
```

```
<p class="code important"> Texte </p>
```



```
* { font-size:16pt; }
td, li {font-weight:bold; }
#idLi1 { color:red; }
.vert { color:green; }
...
<p class="vert"> Bonjour </p>
<table>
  <tr>
    <td class="vert"> un </td>
    <td> deux </td>
  </tr>
</table>
<ol>
  <li id="idLi1"> un </li>
  <li> deux </li>
</ol>
```

Bonjour

un deux

1. un

2. deux

Sélecteurs d'enfant, de descendant ...

- un sélecteur *d'enfant* s'applique à un élément qui est l'enfant d'un autre ; un sélecteur d'enfant est constitué de 2 (ou plus) sélecteurs séparés par >, e.g.,

```
td>strong { color:red; }
```

- un sélecteur de *descendant* s'applique à un élément qui est le descendant d'un autre ; un sélecteur de descendant est constitué de 2 (ou plus) sélecteurs séparés par un espace, e.g.,

```
table strong { color:red; }
```

- un sélecteur *d'adjacence* (adjacent sibling) s'applique à un élément qui est l'enfant suivant d'une même fratrie ; un sélecteur d'adjacence est constitué de 2 (ou plus) sélecteurs séparés par +, e.g.,

```
h1+p { color:red; }
```

- un sélecteur *d'adjacence générale* (general adjacent sibling) s'applique à tous les éléments qui suivent directement un élément particulier ; un sélecteur d'adjacence générale est constitué de 2 sélecteurs séparés par ~, e.g.,

```
h~p { color:red; }
```



```
p { font-family:arial,  
      verdana, sans-serif; }  
div>p { border:1px solid #000000; }  
p+p+p { background-color:#999999; }  
  
<p>  
  Here is an example of some adjacent  
  sibling and child selectors.  
</p>  
<div>  
  <p>One</p>  
  <p>Two</p>  
  <p>Three</p>  
  <p>Four</p>  
  <p>Five</p>  
</div>
```

Here is an example of some adjacent sibling and child selectors.

One

Two

Three

Four

Five

- un sélecteur *d'existence* [att] s'applique aux éléments portant l'attribut att, e.g.,

```
div[id] { color:red; }
```

- un sélecteur *d'égalité* [att="val"] s'applique aux éléments portant l'attribut att avec la valeur "val", e.g.,

```
p[id="summary"] { color:red; }
```

- un sélecteur *espace* [att~="val"] s'applique aux éléments portant l'attribut att avec une valeur qui est une liste de mots (espace comme séparateur), l'un d'entre eux étant "val", e.g.,

```
p[class~="HTML"] { color:red; }
```

- un sélecteur *hyphen* [att|="val"] s'applique aux éléments portant un attribut att dont la valeur est "val" ou commence par "val" suivie par un trait d'union, e.g.,

```
div[lang|="en"] { color:red; }
```



```
p[id] {  
  border : 1px solid #000000;  
}  
p[id="summary"] {  
  background-color : #999999;  
}  
p[class~="HTML"] {  
  border : 3px solid red;  
}  
p[lang|="en"] {  
  color : #ffffff;  
  background-color : green;  
}
```

```
<p id="introduction">Paragraph 1.</p>  
<p id="summary">Paragraph 2.</p>  
<p class="old HTML">Paragraph 3.</p>  
<p lang="en-us">Paragraph 4.</p>
```

Paragraph 1.

Paragraph 1.

Paragraph 3.

Paragraph 4.

- un sélecteur `[att^="val"]` s'applique aux éléments portant un attribut `att` dont la valeur commence par "val" , e.g.,
`td[class^="b"] { color:red; }`
- un sélecteur `[att$="val"]` s'applique aux éléments portant un attribut `att` dont la valeur se termine par "val" , e.g.,
`p[class$="x"] { color:red; }`
- un sélecteur `[att*="val"]` s'applique aux éléments portant un attribut `att` dont la valeur contient "val" , e.g.,
`li[class*="on"] { color:red; }`



```
p[class^="b"]{  
  border : 4px solid #333333;  
}  
p[class*="on"] {  
  color : #ffffff;  
  background-color : #333333;  
}  
p[class$="x"] {  
  border : 1px solid blue;  
}  
  
<p class="begins">Paragraph 5.</p>  
<p class="contains">Paragraph 6.</p>  
<p class="suffix">Paragraph 7.</p>
```

Paragraph 5.

Paragraph 6.

Paragraph 7.

Pseudo-éléments et pseudo-classes

CSS introduit les concepts de pseudo-éléments et de pseudo-classes afin de permettre une mise en forme sur la base d'une information non présente dans la structure d'arbre du document.

- Pseudo-elements create abstractions about the document tree beyond those specified by the document language. For instance, document languages do not offer mechanisms to access the first letter or first line of an element's content. Pseudo-elements may also provide style sheet designers a way to assign style to content that does not exist in the source document (e.g., the `:before` and `:after` pseudo-elements give access to generated content).
- Pseudo-classes classify elements on characteristics other than their name, attributes or content; in principle characteristics that cannot be deduced from the document tree. Pseudo-classes may be dynamic, in the sense that an element may acquire or lose a pseudo-class while a user interacts with the document. The exceptions are `:first-child`, which can be deduced from the document tree, and `:lang()`, which can be deduced from the document tree in some cases.



Les pseudo-classes sont permises partout dans les sélecteurs tandis que les pseudo-éléments peuvent seulement se placer après le dernier sélecteur simple.

- `::first-letter` s'applique à la première lettre de la première ligne d'un bloc de texte, si elle n'est pas précédée par un autre contenu (tel que des images) sur sa ligne. Ce pseudo-élément peut être utilisé pour des effets typographiques.
- `::first-line` s'applique au contenu de la première ligne d'un paragraphe
- `::before` peut être utilisé pour insérer du contenu de manière automatique avant le contenu d'un élément
- `::after` peut être utilisé pour insérer du contenu de manière automatique après le contenu d'un élément
- `::selection` représente une partie du document surligné (p. ex. sélectionné avec la souris ou un autre appareil de pointage) par l'utilisateur.

Remarque

On trouve souvent ces éléments préfixés uniquement par `:` au lieu de `::`



```
p.pageOne::first-letter {
  font-size:42px; width:200px;
}
p.pageOne::first-line {
  font-weight:bold;
}
h1::before {
  content:url(images/penguin.jpeg);
}
h2::after {
  content:"(you need to complete the story)";
}

<p class="pageOne">
  Once upon a time, there was a pseudo-class
  that allowed you to specify ...
</p>
<h1> A story </h1>
<h2> First Chapter </h2>
<p>
  Once upon a time, there was a pseudo-class
  that ...
</p>
```

Once upon a time, there was a pseudo-class that allowed you to specify a different rule for the first letter of an element's content, ...



A story

First Chapter (you need to complete the story)

Once upon a time, there was a pseudo-class that ...

La propriété `content` est utilisée avec les pseudo-éléments `:before` et `:after`, pour insérer du contenu généré automatiquement. Certaines valeurs possibles :

- une chaîne de caractères (ne contenant aucune guillemet)
- une URL
- un compteur pour numéroté des éléments sur une page. Exemple d'instructions :
 - ▶ `counter-reset:chapter; /* Set chapter to 0 */`
 - ▶ `counter-increment:chapter; /* Add 1 to chapter */`
- `open-quote` et `close-quote` pour insérer des guillemets d'ouverture et de fermeture appropriées
- `no-open-quote` et `no-close-quote` pour incrémenter (décrémenter) le niveau d'imbrication pour les guillemets (n'ajoute pas de contenu)



```
body { counter-reset:chapter; }
h1::before {
  content:"Chapter " counter(chapter) ": ";
  counter-increment:chapter;
}
h1 { counter-reset:section; }
h2::before {
  content:counter(chapter)". "counter(section)": ";
  counter-increment:section;
}
```

```
<h1> Web </h1>
<h2>HTML</h2>
<h2>CSS</h2>
<h2>Javascript</h2>
<h1> Docs</h1>
<h2>Text</h2>
<h2>Lists</h2>
<h2>Tables</h2>
```

Chapter 1: Web

1.1: HTML

1.2: CSS

1.3: Javascript

Chapter 2: Docs

2.1: Text

2.2: Lists

2.3: Tables



```
blockquote::before { content:open-quote; }
blockquote::after { content:close-quote; }
p.forBlockquote { display:inline; }

<p>Here are some quotes from Oscar Wilde:</p>
<blockquote>
  <p class="forBlockquote"> Consistency is
    the last refuge of the unimaginative.</p>
</blockquote>
<blockquote>
  <p class="forBlockquote"> If you want to
    tell people the truth, make them laugh,
    otherwise they'll kill you.</p>
</blockquote>
<blockquote>
  <p class="forBlockquote"> It is a ... </p>
</blockquote>
```

Here are some quotes from Oscar Wilde:

“Consistency is the last refuge of the unimaginative.”

“If you want to tell people the truth, make them laugh, otherwise they'll kill you.”

“It is a very sad thing that nowadays there is so little useless information.”

Agissent comme des restrictions au niveau de la sélection (: se lit “tel que”) :

- `:link` s'applique aux éléments (liens) qui n'ont pas encore été visités
- `:visited` s'applique aux éléments qui ont déjà été visités
- `:focus` s'applique aux éléments qui possèdent le focus
- `:hover` s'applique aux éléments pointés (avec par exemple une souris) par l'utilisateur, mais non activés
- `:active` s'applique aux éléments activés

Exemple

```
a.warning:link { color: red; }  
a.external:visited { color:blue; }  
a:focus:hover { background-color:white; }  
img:hover { border: 5px solid green; }
```

Attention

Il faut respecter l'ordre des règles comme dans l'énumération. Par exemple, une règle pour `a:hover` doit se placer après les règles pour `a:link` et `a:visited`.

Agissent comme des restrictions au niveau de la sélection (: se lit “tel que”) :

- `:lang(code)` s'applique aux éléments avec l'attribut `lang` matchant le code donné
- `:first-child`, `:last-child`, `:nth-child(n)`, `:nth-last-child(n)`, `:only-child`
- `:first-of-type`, `:last-of-type`, `:nth-of-type(n)`, `:nth-last-of-type(n)`, `:only-of-type`
- `:enabled`, `:disabled`, `:checked` (pour les `<input>`)
- `:empty` s'applique aux éléments vides
- `:not(S)` où `S` est un sélecteur
- `:contains(text)` s'applique aux éléments dont le contenu textuel contient la sous-chaîne donnée en argument.
- `:target`
- `:root`



```
a:link {
  color : #0000ff;
  text-decoration : none;
}
input:disabled { border : 2px solid black; }
p:first-child { color: red; }
p:nth-child(3) { color : green; }
p:last-child { background-color:yellow; }
div.empty {
  height : 100px; width : 200px;
  background : #666666;
}

<p>First child of the body.</p>
<p>Second child. This input is disabled
  <input type="text" disabled /></p>
<p>Third paragraph.</p>
<p>Two links, one is <a href="toto.com">
  secure, starting with http:// </a> the other
  is to a <a href="titi">.org</a></p>
<div></div>
<p>Last child</p>
```

First child of the body.

Second child. This input is disabled

Third paragraph.

Two links, one is [secure](#), starting with 'http://' the other is to a [.org](#)



Last child

Il existe quelques règles supplémentaires :

- @import importe une autre feuille de style dans la feuille courante
- @charset indique le jeu de caractères utilisé pour la feuille de style
- @font-face décrit une font face
- !important associé à une propriété indique que celle-ci doit toujours être appliquée, quelques soient les autres règles.

Exemple

```
@import "mySheet1.css"          /* first way */  
@import url("mySheet2.css") /* second way */  
@charset "iso-8859-1"  
p { font-size:18px !important; }
```

Attention

Les règles @import et @charset doivent apparaître au début du document (sans même un espace avant celles-ci).

Plan

- 1 Introduction
- 2 Sélecteurs CSS
- 3 Propriétés CSS**
- 4 Positionnement
- 5 Modules CSS3

La table suivante montre les propriétés CSS principales disponibles.

FONT	FONT (ct.)	TEXT (ct.)	TEXT (ct.)
font	font-variant	text-align	white-space
font-family	font-weight	text-decoration	word-spacing
font-size	TEXT	text-indent	BACKGROUND
font-size-adjust	color	text-shadow	background
font-stretch	direction	text-transform	background-attachment
font-style	letter-spacing	unicode-bidi	background-color

BACKGROUND (ct.)	BORDER (ct.)	DIMENSIONS (ct.)	TABLE (ct.)
background-image	border-top-style	min-width	table-layout
background-position	border-top-width	width	LIST and MARKER
background-repeat	border-width	POSITIONING	list-style
BORDER	MARGIN	bottom	list-style-image
border	margin	clip	list-style-position
border-bottom	margin-bottom	left	list-style-type
border-bottom-color	margin-left	overflow	marker-offset
border-bottom-style	margin-right	right	CONTENT
border-bottom-width	margin-top	top	content
border-color	PADDING	vertical-align	counter-increment
border-left	padding	z-index	counter-reset
border-left-color	padding-bottom	OUTLINES	quotes
border-left-style	padding-left	outline	CLASSIFICATION
border-left-width	padding-right	outline-color	clear
border-right	padding-top	outline-style	cursor
border-right-color	DIMENSIONS	outline-width	display
border-right-style	height	TABLE	float
border-right-width	line-height	border-collapse	position
border-style	max-height	border-spacing	visibility
border-top	max-width	caption-side	
border-top-color	min-height	empty-cells	

De nombreuses propriétés utilisent des longueurs comme valeurs. Les longueurs peuvent être mesurées de trois façons :

- unités absolues
 - ▶ pt (point) : 1 pt est équivalent à $1/72$ d'un pouce (inch)
 - ▶ pc (pica) : 1 pc est équivalent à 12 points (ou $1/12$ d'un pouce)
 - ▶ in (inch) : 1 in est équivalent à 2.54 centimètres
 - ▶ cm (centimètre)
 - ▶ mm (millimètre)
- unités relatives
 - ▶ px (pixel) : il s'agit de la plus petite unité de résolution sur l'écran ; elle est donc relative à la résolution de l'écran ;
 - ▶ em : cette unité correspond directement à la taille de police de l'élément de référence qui est soit cet élément ou l'élément parent
 - ▶ ex : une unité ex doit être égal à la hauteur de la lettre x minuscule dans la police de l'élément
- pourcentages : un pourcentage donne une valeur en relation avec une autre

Les unités relatives et les pourcentages permettent un ajustement de taille en fonction du support (media) d'affichage

Nouvelles unités de longueur (CSS3)

Ce sont des unités relatives :

- rem : cette unité correspond à la taille de police de l'élément racine ("root em") ; cela simplifie la gestion des tailles de font relatives ;
- vh : cette unité correspond à 1% de la hauteur de la vue (viewport height) ;
- vw : cette unité correspond à 1% de la largeur de la vue (viewport width).

Remarque

Si la vue change (passage de portrait à landscape sur une tablette par exemple), les unités vh et wh vont permettre l'adaptation.

Illustration

Absolute

xx-small

x-small

small

medium

large

large

large

Pixels

9 px

12 px

14 px

18 px

24 px

36 px

48 px

Points

9 pt

12 pt

14 pt

18 pt

24 pt

36 pt

48 pt

Picas

1 pc

2 pc

3 pc

4 pc

Relative Sizes

smaller

no style

larger

Illustration

Ems

1em

2em

3em

4em

Exs

1ex

2ex

3ex

4ex

5ex

6ex

Percents

50%

75%

100%

150%

200%

400%

Viewport

1vh

2vh

4vh

6vh

Root Element

1rem

2rem

3rem

4rem

Il est possible d'utiliser l'une de ces couleurs en indiquant son nom : `aqua`, `black`, `blue`, `fuchsia`, `gray`, `green`, `lime`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, `white`, and `yellow`.

... et bien d'autres couleurs (un extrait ci-dessous) :









<u>BlueViolet</u>	<u>#8A2BE2</u>	
<u>Brown</u>	<u>#A52A2A</u>	
<u>BurlyWood</u>	<u>#DEB887</u>	
<u>CadetBlue</u>	<u>#5F9EA0</u>	
<u>Chartreuse</u>	<u>#7FFF00</u>	
<u>Chocolate</u>	<u>#D2691E</u>	
<u>Coral</u>	<u>#FF7F50</u>	
<u>CornflowerBlue</u>	<u>#6495ED</u>	

Figure : Extrait de http://www.w3schools.com/html/html_colornames.asp

Couleurs : modèle RGB

CSS permet évidemment d'utiliser le modèle RGB (Red, Green, Blue). Une couleur est codée en indiquant l'intensité pour chacune de ces couleurs primaires.

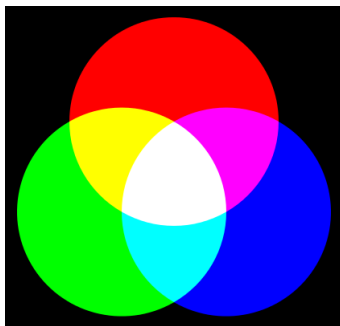


Figure : Modèle RGB (Source wikipedia)

Couleurs : modèle RGB

Certaines propriétés utilisent des couleurs comme valeurs. Une couleur peut donc être définie par son nom ou en mode RGB par :

- un code hexadécimal, i.e, un code à 6 chiffres (en hexadécimal) représentant les quantités de rouge vert et bleu constituant la couleur, précédé par #
- une expression rgb de la forme *rgb(x,y,z)* où *x*, *y* et *z* sont des entiers compris entre 0 et 255
- une expression rgb de la forme *rgb(x%,y%,z%)* où *x*, *y* et *z* sont des nombres compris entre 0 et 100

Color	Hexadecimal Code	rgb (form 1)	rgb (form 2)
red	#FF0000	rgb(255,0,0)	rgb(100%,0%,0%)
olive	#808000	rgb(128,128,0)	rgb(50%,50%,0%)
silver	#C0C0C0	rgb(192,192,192)	rgb(75%,75%,75%)

Exemple

```
p { color:gray; background-color:rgb(120,45,203); }  
td { color:#01E2C1; }
```

Couleurs : modèle HSL

CSS3 introduit le modèle HSL (Hue, Saturation, Lightness), en français TSL (Teinte, Saturation Luminosité).

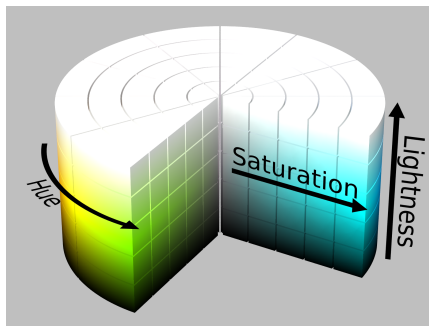


Figure : Modèle HSL (Source wikipedia)

Une couleur peut être définie en mode HSL par une expression `hsl` de la forme `hsl(hue, saturation, lightness)` où :

- *hue* est une valeur entière comprise entre 0 et 360 qui représente la teinte ; l'unité est le degré sur la roue de couleurs avec 0 (ou 360) pour rouge, 120 pour vert, 240 pour bleu.
- *saturation* est une valeur entière comprise entre 0 et 100 qui représente la saturation ; l'unité est le pourcentage (suffixe %).
- *lightness* est une valeur entière comprise entre 0 et 100 qui représente la luminosité (0-noir à 100-blanc); l'unité est le pourcentage (suffixe %).

Exemple

```
#p1 { color: hsl(120,100%,50%); } / * green */  
#p2 { color: hsl(120,100%,75%); } / * light green */  
#p3 { color: hsl(120,100%,25%); } / * dark green */  
#p4 { color: hsl(120,60%,70%); } / * pastel green */
```



```
div { height:20px; width:150px; }  
.one { background-color:hsl(100,50%,50%); }  
.two { background-color:hsl(200,50%,50%); }  
.three { background-color:hsl(300,50%,50%); }  
.four { background-color:hsl(150,20%,50%); }  
.five { background-color:hsl(150,50%,50%); }  
.six { background-color:hsl(150,80%,50%); }  
.seven { background-color:hsl(150,50%,20%); }  
.eight { background-color:hsl(150,50%,50%); }  
.nine { background-color:hsl(150,50%,80%); }
```

```
<h1>Hue</h1>  
<div class="one"></div>  
<div class="two"></div>  
<div class="three"></div>  
<h1>Saturation</h1>  
<div class="four"></div>  
<div class="five"></div>  
<div class="six"></div>  
<h1>Lightness</h1>  
<div class="seven"></div>  
<div class="eight"></div>  
<div class="nine"></div>
```

Hue



Saturation



Lightness



CSS3 permet de gérer la transparence des éléments. Il existe trois façons de le faire :

- en utilisant *rgba* qui prend un quatrième paramètre par rapport à *rgb* ; ce paramètre est la valeur alpha, une valeur entre 0 et 1 qui définit le taux de transparence (0 pour totalement transparent et 1 pour totalement opaque)
- en utilisant *hsla* qui prend un quatrième paramètre par rapport à *hsl* ; ce paramètre est la valeur alpha
- en utilisant la propriété *opacity* ; la propriété s'applique à l'élément entier (y compris ses fils)

Exemple

```
#one { color:rgba(255,0,0,0.5); }  
#two { background-color:hsla(0,100%,50%,0.75); }  
#three { opacity:0.3; }
```



```
p { border:solid 2px black; padding:5px; width:180px; }  
.one { z-index:1; }  
.two { z-index:2;  
    position:relative; left:40px; top:-60px; }  
p:nth-child(2) { background-color:hsl(100,50%,50%); }  
p:nth-child(4) { background-color:hsla(100,50%,50%,0); }  
p:nth-child(6) { background-color:hsla(100,50%,50%,.5); }  
p:nth-child(8) { background-color:hsla(100,50%,50%,1); }
```

```
<p class="one">Voici le paragraphe 1.  
  Il n'est pas très intéressant.</p>  
<p class="two">Voici le paragraphe 2.  
  Il est placé un peu n'importe où.</p>  
<p class="one">Voici le paragraphe 1.  
  Il n'est pas très intéressant.</p>  
<p class="two">Voici le paragraphe 2.  
  Il est totalement transparent.</p>  
<p class="one">Voici le paragraphe 1.  
  Il n'est pas très intéressant.</p>  
<p class="two">Voici le paragraphe 2.  
  Il est un peu transparent.</p>  
<p class="one">Voici le paragraphe 1.  
  Il n'est pas très intéressant.</p>  
<p class="two">Voici le paragraphe 2.  
  Il est totalement opaque.</p>
```

Voici le paragraphe 1.

Il n'
inté

Voici le paragraphe 2.
Il est placé un peu
n'importe où.

Voici le paragraphe 1.

Il n'est pas très
intéressant.

Voici le paragraphe 2.
Il est totalement
transparent.

Voici le paragraphe 1.

Il n'est pas très
intéressant.

Voici le paragraphe 2.
Il est un peu
transparent.

Voici le paragraphe 1.

Il n'
inté

Voici le paragraphe 2.
Il est totalement
opaque.

Contrôler les polices (fonts)

Les propriétés suivantes vous permettent d'affecter directement la police et son apparence.

Propriété	Objet
font	permet de combiner plusieurs des propriétés suivantes
font-family	Spécifie la famille de police à utiliser (l'utilisateur doit avoir celle-ci installée sur son ordinateur)
font-size	Spécifie la taille de la police
font-weight	Spécifie si la police doit être normale, grasse ou plus grasse que l'élément parent
font-style	Spécifie si la police doit être normale, italique, ou oblique (an oblique font is the normal font on a slant rather than a separate italic version of the font)
font-stretch	Permet de contrôler la largeur des lettres de la police (mais pas les espaces entre eux)
font-variant	Spécifie si la police doit être normale ou en petite capitale (small caps)
font-size-adjust	Permet d'altérer le ratio largeur/hauteur des caractères de la police

Typefaces et Fonts

Une font (police) n'est pas la même chose qu'une typeface :

- une typeface est une famille de polices ; e.g. Times et Arial sont des typefaces;
- une font est un membre précis d'une typeface, telle que Arial 12-point bold.



Proportional
Monospace

Serif

The small decorative pieces on the ends of each character are called Serifs

San Serif

Propriété font-family

La propriété font-family permet de lister plusieurs noms. Si le navigateur ne supporte pas la première typeface, il essaie la suivante, etc. (système "fallback"). Il y a deux types de noms pour les familles de police :

- les noms de familles précises telles que "times", "courier", "arial", etc.
- les noms de familles génériques telles que "serif", "sans-serif", "cursive", "fantasy", "monospace".

Remarque

Si le nom d'une typeface contient des espaces, tel que times new roman, vous devez placer le nom entre guillemets.

Exemple

```
p { font-family:"Times New Roman", Georgia, Serif; }
```



La propriété font-size vous permet de spécifier la taille d'une police de différentes manières :

- taille absolue : "xx-small", "x-small", "small", etc.
- taille relative : "smaller", "larger"
- longueur : exprimée en px, em, ex, pt, in, cm ou mm
- pourcentage (en relation avec l'élément parent)

Exemple

```
p.one { font-size:xx-small; }  
p.twelve {font-size:12px; }  
p.thirteen {font-size:3pc; }  
p.fourteen { font-size:10%; }
```

Parmi les autres propriétés, on trouve :

- `font-weight` : "normal", "bold", "bolder", "lighter", "100", "200", etc.
- `font-style` : "normal", "italic", "oblique"
- `font-variant` : "normal", "smallcaps"

Exemple

```
p.one { font-weight:bold; }  
p.twelve { font-style:italic; }  
p.thirteen { font-weight:bold; font-variant:smallcaps; }
```

Formater le texte

Les propriétés suivantes permettent d'affecter directement l'apparence ou le formatage du texte.

Propriété	Objet
color	Spécifie la couleur du texte
text-align	Spécifie l'alignement du texte au sein de l'élément parent
vertical-align	Alignement vertical du texte au sein de l'élément parent et en relation avec l'élément parent
text-decoration	Spécifie si le texte doit être soulignée, surlignée, barré ou clignotant
text-indent	Spécifie une indentation à partir de la bordure gauche pour le texte
text-transform	Spécifie que le contenu de l'élément doit être en majuscule, minuscule, ou capitalized
text-shadow	Spécifie que le texte doit avoir une ombre
letter-spacing	Contrôle la largeur entre les lettres (connu sous le terme de kerning)
word-spacing	Contrôle l'espacement entre chaque mot
white-space	Spécifie si les espaces doivent être regroupés (en 1), préservés, or prevented from wrapping
direction	Spécifie la direction du texte (similaire à l'attribut HTML dir)
unicode-bidi	Permet de créer du texte bidirectionnel

Parmi les propriétés, on trouve donc :

- `text-align` : "left", "right", "center", "justify"
- `text-decoration` : "underline", "overline", "line-through", "blink"
- `text-indent` : une longueur pour l'indentation de la première ligne de texte dans un élément
- `text-transform` : "none", "capitalize", "uppercase", "lowercase"

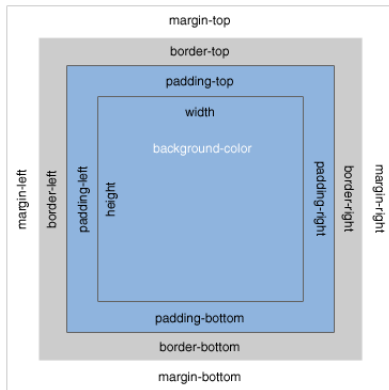
Exemple

```
p { text-align:justify; }  
p.twelve { text-decoration:underline; }  
p.thirteen { text-indent:3em; }  
td { text-transform:uppercase; }
```

Le modèle des boîtes

Chaque élément est traité comme une boîte (box) en CSS, et chaque boîte a trois propriétés :

- **border** : même si on ne peut pas la voir, elle existe ;
- **margin** : c'est l'espace entre la bordure et la boîte adjacente ;
- **padding** : c'est l'espace entre le contenu de la boîte et la bordure.



Le modèle des boîtes

La boîte pour les éléments en ligne se place dans son élément parent. La boîte pour les éléments bloc s'étend sur toute la largeur de la page.

Exemple

```
body, h1, p, img, b {  
  border-style:solid;  
  border-width:2px;  
  border-color:#000000;  
  padding:2px;  
}
```

Remarque

Quand la marge basse d'un élément rencontre la marge haute d'un autre, seule la plus grande des deux est affichée.



```
body, h1, p, img, b {
  border:solid 2px black;
  padding:2px;
}
h1, b {
  background-color:#cccccc;
}
```

```
<h1>Les boites</h1>
<p class="description">When you are styling ...
... of <b>boxes</b>.</p>
<p>Each element is treated as if it generates
a new box. Each box can have ... with it.</p>
<p>  </p>
<p>As you can see from the diagram above,
each box has a <b>border</b>. ...
... can have <b>padding</b>, and outside
... can have a <b>margin</b> ... </p>
```

Les boites

When you are styling a web page with CSS page you must start to think in terms of **boxes**.

Each element is treated as if it generates a new box. Each box can have new rules associated with it.



As you can see from the diagram above, each box has a **border**. Between the content and the border you can have **padding**, and outside of the border you can have a **margin** to separate this box from any neighboring boxes.

Propriétés de bordures

Les propriétés de bordure sont :

- `border-color` : vous pouvez utiliser des codes hex, des noms de couleurs, des expressions `rgb` de la forme `rgb(x,y,z)` ou `rgb(a%, b%, c%)`.
- `border-style` : `"none"`, `"solid"`, `"dotted"`, `"dashed"`, `"double"`, `"groove"`,...
- `border-width` : `"thin"`, `"medium"`, `"thick"`, ou une longueur (mais pas un pourcentage)

Remarque

Vous pouvez changer de manière individuelle les propriétés des bordures droite, gauche, haute et basse. Par exemple, vous pouvez utiliser les propriétés `border-color-right`, `border-color-bottom`, `border-color-left`, et `border-color-top`.

Remarque

La propriété `border` permet de spécifier à la fois la couleur, le style et la largeur :

```
p { border: 4px solid red; }
```

Propriétés padding et margin

Pour les boîtes, on peut donc utiliser :

- `padding` : une longueur, un pourcentage (de la boîte) ou "inherit" (la valeur padding de l'élément parent)
- `margin` : une longueur, un pourcentage (de la boîte) ou "inherit" (la valeur padding de l'élément parent)

Exemple

```
<style type="text/css" >
  .a, .b { border-style:solid; border-color:#000000;
           border-width:2px; width:100px; }
  .b { padding:5px; }
</style>
...
<p class="a">Without padding between the edge of ...</p>
<p class="b">With padding between the edge of ... </p>
```



```
body { border:solid 2px red; }  
p { border:solid 2px black; width:150px; }  
p:first-child { margin:0px; }  
p:nth-last-child(2) {  
    margin:5px; padding:5px;  
}  
p:last-child {  
    margin:20px; padding:20px;  
}
```

<p>Voici un premier paragraphe. Pas
de marge explicite et pas de padding.</p>
<p>Voici un second paragraphe. Une
marge (5px) et du padding (5px). </p>
<p>Voici un troisième paragraphe. Une
marge (20px) et du padding (20px).</p>

Voici un premier
paragraphe. Pas de
marge explicite et
pas de padding.

Voici un second
paragraphe. Une
marge (5px) et du
padding (5px).

Voici un troisième
paragraphe. Une
marge (20px) et
du padding
(20px).

Les dimensions d'une boite sont contrôlées par les propriétés suivantes :

- `height` et `width` : une longueur, un pourcentage ou "auto" (valeur par défaut)
- `line-height` : permet d'augmenter l'espacement entre les lignes de texte
- `min-width` and `max-width` : spécifie une largeur minimale et maximale pour une boite ; ces propriétés empêchent une boite de devenir trop large ou trop étroite
- `min-height` et `max-height` : spécifie une hauteur minimale et maximale pour une boite ; attention, le contenu peut dépasser de la boite
- `overflow` : spécifie ce qu'il faut faire lorsque le contenu déborde ; les valeurs possibles sont :
 - ▶ `"visible"` : ce qui déborde n'est pas coupé ; c'est affiché en dehors de la boite (valeur par défaut)
 - ▶ `"hidden"` : ce qui déborde est coupé (donc, invisible)
 - ▶ `"scroll"` : un ascenseur (toujours visible) permet de visualiser ce qui déborde
 - ▶ `"auto"` : un ascenseur est présent si nécessaire
 - ▶ `"inherit"`

```
p { max-height:75px; max-width:250px;
padding:5px; margin:10px;
border:1px solid black;
}
p:first-child { overflow:hidden;
color:red; }
p:nth-child(2) { overflow:scroll;
color:blue; }
p:nth-child(3) { color:green; }

<body>
  <p> Le comportement de ces boites ...
    ... la propriété overflow. </p>
  <p> Le comportement de ces boites ...
    ... la propriété overflow. </p>
  <p> Le comportement de ces boites ...
    ... la propriété overflow. </p>
  <p> Un simple paragraphe qui est
    chevauché. </p>
```

Le comportement de ces boîtes n'est pas la même car nous jouons avec la propriété overflow. Le comportement de

Le comportement de ces boîtes n'est pas la même car nous jouons avec la propriété `overflow`. Le comportement

Le comportement de ces boîtes n'est pas la même car nous jouons avec la propriété overflow. Le comportement de ces boîtes n'est pas la même car un simple paragraphe overflow.

Les propriétés suivantes vous permettent de spécifier comment le fond de la fenêtre complète du navigateur ou simplement d'une boîte individuelle doit apparaître.

Propriété	Objet
background-color	Spécifie une couleur pour le fond de la page ou de la boîte
background-image	Fixe une image en fond de page ou de boîte
background-repeat	Indique si l'image de fond doit être répétée sur la page ou boîte
background-attachment	Indique si une image pour le fond doit rester à la même position quand l'utilisateur scrolle
background-position	Indique où une image doit être positionnée dans soit la fenêtre, soit la boîte contenante
background	Raccourci pour spécifier plusieurs propriétés à la fois

Contrôler le fond

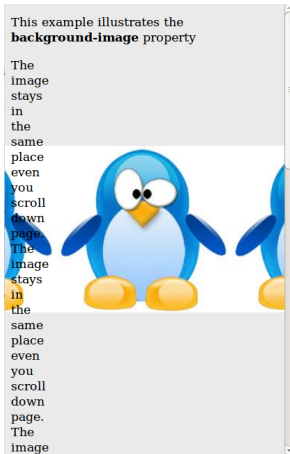
- `background-image` : permet d'utiliser une image comme fond de boîte ; la syntaxe est : `background-image:url(imageFilename)`
Cette propriété prend le pas sur la propriété `background-color`. Par défaut, la propriété `background-image` répète une image à la fois verticalement et horizontalement.
- `background-repeat` : ses valeurs possibles sont "repeat", "repeat-x", "repeat-y", et "no-repeat"
- `background-position` : ses valeurs possibles sont "x% y%", "x y", "center", "top", ...
- `background-attachment` : permet de spécifier que l'image doit rester à la même position même quand la page est scrollée ; ses valeurs possibles sont "scroll", "fixed", et "inherit"
- `background` : permet de spécifier ces propriétés dans n'importe quel ordre

Exemple

```
background-image:url("images/penguin.jpeg");  
background-attachment:fixed; background-position:center;  
background-repeat:repeat-x; background-color:#eaeaea;
```




```
body {  
  background-image:url("images/penguin.jpeg");  
  background-attachment:fixed;  
  background-position:center;  
  background-repeat:repeat-x;  
  background-color:#eaeaea;  
}  
p.narrow { width:20px; }  
  
<body>  
  <p>This example illustrates the <strong>  
background-image</strong> property </p>  
  <p class="narrow">  
    The image stays in the same place  
    even you scroll down page.  
    ...  
    The image stays in the same place  
    even you scroll down page.  
  </p>  
</body>
```



Les propriétés pour les liste sont indiquées dans la table suivante :

Propriété	Objet
list-style-type	permet de contrôler la forme ou apparence du marqueur de liste (puce ou nombre)
list-style-position	Spécifie si une entrée longue prenant plus d'une ligne de texte doit s'aligner sur la première ligne ou sur le marqueur
list-style-image	Spécifie une image pour le marqueur
list-style	Raccourci pour les propriétés précédentes
marker-offset	Spécifie la distance entre un marqueur et le texte

- `list-style-type` : permet de contrôler la forme du marqueur :

Valeur	Marqueur	Exemple
Pour les listes non ordonnées (<code>ul</code>)		
<code>none</code>	rien	
<code>disc</code>	cercle plein	●
<code>circle</code>	cercle vide	○
<code>square</code>	carré plein	■
Pour les listes ordonnées (<code>ol</code>)		
<code>decimal</code>	nombre	1,2,3
<code>decimal-leading-zero</code>	nombre précédé de 0	01, 02, 03
<code>lower-alpha</code>	caractère alphanumérique en minuscule	a, b, c
<code>upper-alpha</code>	caractère alphanumérique en majuscule	A, B, C
<code>lower-roman</code>	nombre romain en minuscule	i, ii, iii
<code>upper-roman</code>	nombre romain en majuscule	I, II, III

- `list-style-position` : "inside", "outside" ou "inherit"
- `list-style-image` : une URL, "none" ou "inherit"



```
li:nth-child(1) { list-style:none; }  
li:nth-child(2) { list-style:disc; }  
li:nth-child(3) { list-style:circle; }  
li:nth-child(4) { list-style:square; }  
li:nth-child(5) { list-style:decimal; }  
li:nth-child(6) { list-style:lower-alpha; }  
li:nth-child(7) { list-style:upper-alpha; }  
li:nth-child(8) { list-style:lower-roman; }  
li:nth-child(9) { list-style:upper-roman; }  
li:nth-child(10) { list-style:upper-roman;  
    list-style-position:inside; }
```

```
<ul>  
  <li>article de liste </li>  
  <li>article de liste </li>  
  ...  
  <li>article de liste </li>  
</ul>
```

- article de liste
- article de liste
- article de liste
- article de liste
- 5. article de liste
- f. article de liste
- G. article de liste
- viii. article de liste
- IX. article de liste
- X. article de liste
- article de liste

Propriété	Objet
<code>border-collapse</code>	Indique si le navigateur doit contrôler l'apparence des bordures adjacentes ou si chaque cellule doit maintenir son style
<code>border-spacing</code>	Spécifie la largeur entre les cellules
<code>caption-side</code>	Spécifie sur quel côté la légende doit apparaître
<code>empty-cells</code>	Spécifie si la bordure doit être affichée lorsque la cellule est vide
<code>table-layout</code>	Permet aux navigateurs d'accélérer la mise en forme de la table en utilisant les premières propriétés chargées pour une colonne (plutôt que d'avoir à charger la table complète avant de l'afficher)

Remarque

Si vous voulez cacher explicitement ou montrer explicitement les bordures pour les cellules vides, utiliser la propriété `empty-cells` car les navigateurs traitent/affichent différemment les cellules vides.

Les Outlines sont attachés aux boîtes et sont utiles pour mettre en valeur certains aspects d'une page. Un outline ne prend pas de place ; c'est comme si le style outline se positionne au dessus de la page lorsqu'il est affiché.

Propriété	Objet
outline-width	Spécifie la largeur du outline
outline-style	Spécifie le style de ligne pour le outline
outline-color	Spécifie la couleur du outline
outline	Raccourci pour les propriétés ci-dessus

Remarque

Un outline est toujours identique sur chaque coté.

Propriétés diverses

- `cursor` : spécifie le type du curseur de souris à afficher lorsqu'un élément est pointé.
- `display` : spécifie le type de boîte à générer, c'est-à-dire soit comme `"inline"` soit comme `"block"` ; cela permet de forcer un élément à être d'un type différent ; `"none"` peut être utilisé pour ignorer l'élément (et ainsi ne pas l'afficher).
- `visibility` : permet de cacher une boîte, mais affecte la mise en page ; les deux valeurs possibles sont `"visible"` et `"hidden"`

Remarque

On peut vouloir changer le curseur (de pointeur à main) lorsque une image ou un bouton est survolé.

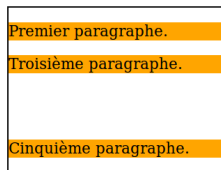
Attention

Ne pas utiliser la propriété `visibility` pour cacher des informations confidentielles.



```
body { cursor:move;
  border:solid 2px black; }
p { background-color:orange; }
p.invisible { visibility:hidden; }
p.ignored { display:none; }

<p>Premier paragraphe.</p>
<p class="ignored">Second paragraphe
  totalement ignoré.</p>
<p>Troisième paragraphe.</p>
<p class="invisible">Quatrième paragraphe
  invisible.</p>
<p>Cinquième paragraphe.</p>
```





Beaucoup de propriétés CSS, lorsqu'appliquées à un élément, sont héritées par les éléments enfant. Quand une règle redéfinit une propriété héritée, elle prend le dessus.

Exemple

```
body { font-size:16pt; }  
td { font-size:10pt; }
```

Remarque

L'héritage permet d'éviter d'écrire des règles pour chaque élément, et permet donc de réduire la taille de la feuille de styles.

Attention

Certaines propriétés ne peuvent pas être héritées.

Plan

- 1 Introduction
- 2 Sélecteurs CSS
- 3 Propriétés CSS
- 4 Positionnement**
- 5 Modules CSS3

Par défaut, les éléments bloc au sein d'une page HTML seront placés de haut en bas et les éléments en ligne de gauche à droite (au sein d'un bloc). Cela est appelé le *flux normal*. Si vous voulez contrôler la position des éléments, vous devez utiliser les propriétés suivantes :

- `position` qui permet de spécifier un schéma de positionnement pour une boîte
- les propriétés de décalage de boîte `top`, `right`, `bottom` et `left`
- `z-index` qui établit l'ordre d'empilement des éléments
- `float` qui spécifie si une boîte (un élément) doit flotter ou non (i.e., être hors du flux normal)
- `clear` qui est utilisé pour le contrôle des boîtes qui apparaissent après une boîte flottante

Remarque

Eviter d'utiliser des tables pour le positionnement.

Propriétés position et de décalage de boîte

Les valeurs pour `position` sont :

- `static` : utilisé pour le flux normal (valeur par défaut)
- `relative` : l'élément est positionné relativement à sa position normale (en considérant le flux normal)
- `absolute` : l'élément est positionné relativement au premier élément ancêtre qui a une position autre que `static` ; si il n'y en a pas, c'est relativement à `<html>`
- `fixed` : l'élément est positionné relativement à un point fixe (e.g., le coin supérieur gauche de la fenêtre du navigateur) ; il ne bougera pas même si la fenêtre est scrollée

Quand une boîte a la propriété `position` avec la valeur "`relative`", "`absolute`" ou "`fixed`", les propriétés de décalage de boîte, dont les valeurs sont une longueur, un pourcentage ou "`auto`", sont utilisées pour indiquer où il doit être positionné. Ce sont :

- `top` : décalage par rapport au sommet de l'élément parent
- `right` : décalage par rapport à la droite de l'élément parent
- `bottom` : décalage par rapport à la base de l'élément parent
- `left` : décalage par rapport à la gauche de l'élément parent

Positionnement relatif et absolu

Le positionnement relatif positionne un élément par rapport à ce que serait sa position en flux normal. Il est déplacé en prenant en compte les valeurs données par les propriétés de décalage de boîte.

Le positionnement absolu écarte un élément du flux normal. Il est déplacé d'un élément référence (premier élément ancêtre static ou `<html>`) en prenant en compte les valeurs données par les propriétés de décalage de boîte.

Exemple

```
p.two { position:relative; left:40px; top:-40px; }  
p.twoBis { position:absolute; left:50px; top:-25px; }
```

Vous devez spécifier seulement un décalage gauche ou droite, et seulement un décalage haut ou bas.

Attention

Sauf si vous utilisez une couleur de fond pour une boîte, elle sera transparente par défaut. Aussi, tout chevauchement inapproprié sera illisible.



```
p { border:solid 2px black; padding:5px;
  width:200px; background-color:#FFFFFF; }
p.two { position:relative;
  left:40px; top:-40px; }
div.page { position:absolute;
  left:50px; top: 200px;
  border:solid 2px black; padding:20px;
}
p.twoBis { position:absolute;
  left:50px; top:-25px; }
```

```
<p>Un paragraphe qui sera en haut de la page.</p>
<p class="two">Un paragraphe qui sera
  au dessus du premier.</p>
<p>Un pagagraphe qui sera au milieu
  de la page.</p>
<div class="page">
  <p>Un paragraphe qui va se balader
    dans la page.</p>
  <p class="twoBis">Encore un autre pagrapahe
    placé dans le div.</p>
  <p>Et pour finir un dernier paragraphe
    qui sera en bas.</p>
</div>
```

Un paragraphe qui sera en haut de la page.
Un paragraphe qui sera au dessus du premier.

Un pagagraphe qui sera au milieu de la page.

Encore un autre pagrapahe placé dans le div.

Un paragraphe qui va se balader dans la page.

Et pour finir un dernier paragraphe qui sera en bas.

Le positionnement fixe écarte un élément du flux normal, et en outre, la boîte ne bouge pas lorsque la page est scrollée.

Lorsque vous avez des boîtes positionnées en utilisant `relative`, `absolute` ou `fixed`, vous pouvez contrôler quelles boîtes apparaissent au premier plan en utilisant la propriété `z-index`. La valeur de la propriété `z-index` est un nombre, et plus ce nombre est élevé, plus l'élément est affiché sur un plan proche.

Exemple

```
p.two { position:absolute; left:30px; top:120px; z-index:3; }  
p.three { position:absolute; left:10px; top:140px; z-index:2; }
```



```
header {
  position:fixed; top:0px; left:0px;
  font-size:28px;
  color:white; background-color:#666666;
  border:solid 2px black; padding:20px; }
p { width:200px;
  background-color:white;
  border:solid 2px black; padding:5px; }
p:first-child { margin-top:120px; }
p:nth-child(2) { position:absolute;
  left:30px; top:120px; z-index:3; }
p:nth-child(3) { position:absolute;
  left:40px; top:140px; z-index:2; }

<header>CSS</header>
<main>
  <p>Ceci est un premier paragraphe de la
    page.</p>
  <p>ceci est le deuxième paragraphe qui se
    situe quelque part.</p>
  <p>ceci est le troisième paragraphe de la
    page qui est sous le second paragraphe.</p>
  <p>Cette page contient de nombreux paragraphes
    ...
    CSS 3
```

CSS

Ceci est un premier
paragraphe qui se situe
quelque part.

Ceci est sous le second
paragraphe.

Ceci est sous le second
paragraphe.
positionnement fixe.

Cette page contient de
nombreux paragraphes
qui vous permettent de
constater le
positionnement fixe.

Cette page contient de
nombreux paragraphes
qui vous permettent de
constater le
positionnement fixe.

La propriété float

La propriété `float` permet de considérer un élément hors du flux normal et de le placer aussi loin que possible sur la gauche ou la droite de la boîte parent. La boîte flottante sera alignée sur le haut de la boîte parent.

- `left` : la boîte flotte sur la gauche de l'élément parent et le contenu de l'élément parent se positionne sur sa droite.
- `right` : la boîte flotte sur la droite de l'élément parent et le contenu de l'élément parent se positionne sur sa gauche.
- `none` : la boîte ne flotte pas et se positionne dans le flux normal.
- `inherit` : la boîte prend la même valeur de propriété que l'élément parent.

Remarque

Les éléments bloc et les éléments en-ligne peuvent être rendus flottants.

Attention

Lorsque vous spécifiez la propriété `float`, vous devez aussi utiliser la propriété `width` pour indiquer la largeur de la boîte parent que la boîte flottante doit occuper ; autrement, la boîte flottante occupera automatiquement toute la place.



```
body { width:350px; }
h1 { text-align:center; border:solid 2px; }
p { border:solid 1px; padding:5px; }
q { float:right; width:150px; quotes: " " " ";
padding:5px; margin:5px; border:solid 1px;
background-color:#f1f1f1;
}
```

```
<body>
  <h1>Floating</h1>
  <p>
    <q>Saint-Mittre, un saint provençal fort
      honoré dans la contrée.</q>
    L'aire Saint-Mittre est un carré ...
    ... seuls traversent.
  </p>
</body>
```

Floating

L'aire Saint-Mittre est un carré long, d'une certaine étendue, qui s'allonge au ras du trottoir de la route, dont une simple bande d'herbe usée la sépare. D'un côté, à droite, une ruelle, qui va se terminer en cul-de-sac, la borde d'une rangée de masures ; à gauche et au fond, elle est close par deux pans de muraille rongés de mousse, au-dessus desquels on aperçoit les branches hautes des mûriers du Jas-Meiffren, grande propriété qui a son entrée plus bas dans le faubourg. Ainsi fermée de trois côtés, l'aire est comme une place qui ne conduit nulle part et que les promeneurs seuls traversent.

"Saint-Mittre, un saint provençal fort honoré dans la contrée."

Pour contrôler les éléments qui peuvent se trouver autour du contenu des éléments flottants, utiliser la propriété `clear`. Ses valeurs possibles sont :

- `left` : le contenu de l'élément avec la propriété `clear` ne peut apparaître à la gauche d'un élément flottant.
- `right` : le contenu de l'élément avec la propriété `clear` ne peut apparaître à la droite d'un élément flottant.
- `both` : le contenu de l'élément avec la propriété `clear` ne peut apparaître sur aucun des deux cotés d'un élément flottant.
- `none` : permet les éléments flottants des deux cotés.



```
body { width:350px; }
h1 { text-align:center; border:solid 2px; }
p { clear:right;
    border:solid 1px; padding:5px; }
q { float:right; width:150px; quotes: " " " ";
    padding:5px; margin:5px; border:solid 1px;
    background-color:#f1f1f1; display:block; }

<body>
  <h1>Clear</h1>
  <q>Saint-Mittre, un saint provençal fort
    honoré dans la contrée.</q>
  <p> L'aire Saint-Mittre est un carré long ...
    ... seuls traversent.
  </p>
</body>
```

Clear

“Saint-Mittre, un
saint provençal
fort honoré dans
la contrée.”

L'aire Saint-Mittre est un carré long, d'une certaine étendue, qui s'allonge au ras du trottoir de la route, dont une simple bande d'herbe usée la sépare. D'un côté, à droite, une ruelle, qui va se terminer en cul-de-sac, la borde d'une rangée de masures ; à gauche et au fond, elle est close par deux pans de muraille rongés de mousse, au-dessus desquels on aperçoit les branches hautes des mûriers du Jas-Meiffren, grande propriété qui a son entrée plus bas dans le faubourg. Ainsi fermée de trois côtés, l'aire est comme une place qui ne conduit nulle part et que les promeneurs seuls traversent.

En considérant les statistiques concernant les navigateurs et la résolution d'écran :

- <http://www.w3counter.com/globalstats.php>
- http://www.w3schools.com/browsers/browsers_stats.asp
- http://www.w3schools.com/browsers/browsers_display.asp

C'est une bonne idée de concevoir des pages web dont le contenu tient sur environ 980 pixels de large tandis que les informations importantes tiennent en hauteur sur les 550 premiers pixels.

Il y a plusieurs façons d'agencer les pages web :

- la conception largeur-fixe force les pages à une certaine hauteur et largeur
- la conception fluide (responsive design)



Il suffit de spécifier explicitement les dimensions de la page.

Exemple

```
#page {  
  width:800px; margin-left:auto; margin-right:auto;  
  background-color:#ffffff;  
  border-style:1px solid #666666; padding:20px;  
  font-size:12px;  
}
```

```
<body>  
  <div id="page">  
    <!-- content of page goes here -->  
  </div>  
</body>
```

Par exemple, une colonne et quatre lignes (divisions) : la première contient le nom de l'institution et le logo, la seconde contient les éléments de navigation (dans la page), la troisième présente le contenu et la dernière le footer.

Exemple

```
<body>
  <div class="page">
    <header> <!-- COMPANY NAME/LOGO goes here --> </header>
    <nav> <!-- NAVIGATION goes here --> </nav>
    <main> <!-- PAGE CONTENT goes here --> </main>
    <footer> <!-- FOOTER goes here --> </footer>
  </div>
</body>
```



```
body { background-color:#d6d6d6; }
.page {
  width:500px; margin:0 auto;
  background-color:white;
  border:solid 1px #666666; }
header { background-color:#eee; padding:3px; }
nav { font-weight:bold;
  background-color:#e3e3e3; padding:5px; }
main { padding:10px; }
footer { text-align:center; }
```

```
<body>
  <div class="page">
    <header><h1>My Company Name</h1></header>
    <nav><a href="">Home</a> | ... </nav>
    <main>
      <h2>Sample Web Page</h2>
      <p>Lorem ipsum ... </p>
      <p>Mauris quis urna ... </p>
    </main>
    <footer>
      Toto SA <br /> <address> ... </address>
    </footer>
  </div>
</body>
```

My Company Name

[Home](#) | [Products](#) | [Services](#) | [About Us](#) | [Contact Us](#)

Sample Web Page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc eleifend, erat id commodo placerat, nulla purus bibendum justo, in dictum orci mi vitae nulla. Nullam semper viverra nulla. Sed lacinia feugiat eros. Maecenas ullamcorper ligula quis odio. Donec pede massa, pharetra sit amet, accumsan a, iaculis egestas, lectus. Etiam ullamcorper elementum wisi. Etiam et felis aliquet dui tempus sagittis. Donec dapibus ipsum id leo. Integer est ante, imperdiet non, suscipit sit amet, varius a, sem. Integer lobortis wisi id erat. Nullam aliquet augue ac elit. Nulla facilisi. Vivamus ligula tortor, molestie at, accumsan quis, semper vitae, augue. Praesent pede neque, sollicitudin non, facilisis sed, viverra a, pede. Cras nec urna. Curabitur ut metus. Curabitur erat lacus, tempus vitae, elementum nec, pulvinar vel, leo. Sed a velit. Proin erat. Donec sem.

Mauris quis urna. Donec sodales, risus non dictum ultricies, nisl lorem imperdiet sapien, vel euismod orci risus non lorem. Nam tempus, pede nec tincidunt lacinia, lectus mauris malesuada quam, a tincidunt elit justo a ligula. Nunc cursus nonummy nulla. Quisque adipiscing. Donec nisl elit, viverra in, elementum eu, auctor id, eros. Sed sagittis, neque vel blandit tempor, justo odio posuere nulla, at condimentum lorem nibh suscipit arcu. Mauris rhoncus, nunc vel hendrerit aliquet, purus velit iaculis mauris, aliquam eleifend ante augue ut velit. Pellentesque sed turpis vel odio varius posuere. Ut urna mi, ultricies ut, lobortis in, varius porta, nulla.

TOTO SA
CONTACT: TOTO@GLOBULE.FR

Format double-colonne

Conservons le même code, mais changeons les règles CSS.

Exemple

```
.page {  
    ...  
    background-image:url(images/2columnbackground.gif);  
    background-repeat:repeat-y;  
    ...  
}  
nav { float:left; width:100px; ... }  
main { margin-left:100px; ... }  
...
```

Remarque

L'image doit faire un pixel de haut.



```
body { background-color:#d6d6d6; }
.page {
  width:500px; margin:0 auto;
  background-image:url(2columnbackground.gif);
  background-repeat:repeat-y;
  border:solid 1px #666666; }
header {
  background-color:#f3f3f3;
  padding:3px; }
nav {
  font-weight:bold;
  float:left; width:100px; padding:5px; }
main {
  padding:10px; margin-left:100px; }
a { display:block; margin:6px; }
footer {
  text-align:center;
  font-variant:small-caps; }

<body>
...
</body>
```

My Company Name

[Home](#)
[Products](#)
[Services](#)
[About Us](#)
[Contact](#)

Sample Web Page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc eleifend, erat id commodo placerat, nulla purus bibendum justo, in dictum orci mi vitae nulla. Nullam semper viverra nulla. Sed lacinia feugiat eros. Maecenas ullamcorper ligula quis odio. Donec pede massa, pharetra sit amet, accumsan a, iaculis egestas, lectus. Etiam ullamcorper elementum wisi. Etiam et felis aliquet dui tempus sagittis. Donec dapibus ipsum id leo. Integer est ante, imperdiet non, suscipit sit amet, varius a, sem. Integer lobortis wisi id erat. Nullam aliquet augue ac elit. Nulla facilisi. Vivamus ligula tortor, molestie at, accumsan quis, semper vitae, augue. Praesent pede neque, sollicitudin non, facilisis sed, viverra a, pede. Cras nec urna. Curabitur ut metus. Curabitur erat lacus, tempus vitae, elementum nec, pulvinar vel, leo. Sed a velit. Proin erat. Donec sem.

Mauris quis urna. Donec sodales, risus non dictum ultricies, nisi lorem imperdiet sapien, vel euismod orci risus non lorem. Nam tempus, pede nec tincidunt lacinia, lectus mauris malesuada quam, a tincidunt elit justo a ligula. Nunc cursus nonummy nulla. Quisque adipiscing. Donec nisi elit, viverra in, elementum eu, auctor id, eros. Sed sagittis, neque vel blandit tempor, justo odio posuere nulla, at condimentum lorem nibh suscipit arcu. Mauris rhoncus, nunc vel hendrerit aliquet, purus velit iaculis mauris, aliquam eleifend ante augue ut velit. Pellentesque sed turpis vel odio varius posuere. Ut urna mi, ultricies ut, lobortis in, varius porta, nulla.

Toto SA
CONTACT: TOTO@GLOBULE.FR

Conception fluide

```
<body>
  <div class="gauche">
    ...
  </div>
  <div class="droite">
    ...
  </div>
</body>
```

Fixed Layout

```
body { margin: 0px; }
.gauche {
  width: 275px; float: left;
}
.droite {
  width: 685px; float: left;
}
```

Fluid Layout

```
body { margin: 0px; }
.gauche {
  width: 28.6%; float: left;
}
.droite {
  width: 71.4%; float: left;
}
```

Conception fluide (Responsive Design)

Le *Responsive Design* nécessite d'utiliser des tailles proportionnelles pour la plupart des éléments.

Par exemple, en gardant 5% pour les marges :

Example

```
body { margin: 0px; }  
.gauche {  
  width: 27%;  
  margin-left: 1.66%;  
  margin-right: 1.66%;  
  float: left;  
}  
.droite {  
  width: 68%;  
  margin-right: 1.66%;  
  float: left;  
}
```

Il ne faut plus utiliser de valeurs absolues représentant la taille de l'image.

```
#image1 { max-width: 100%; }
```

Remarque

Avec une marge autour d'une image, il faut s'assurer que la somme des valeurs pour `margin-left`, `margin-right`, et `max-width` totalise 100%.

Remarque

Sur le site <http://mobiletest.me>, vous pouvez tester l'apparence de vos pages sur différents types d'appareil.

Attention

Ne pas utiliser des pixels et des points. Utiliser des pourcentages, ou des unités em et rem.

```
body {  
  font-size: 100%  
}  
p {  
  font-size: 0.9em  
}  
h1 {  
  font-size: 2em  
}
```

CSS

```
.gauche {  
    width: 28.6%; float: left;  
}  
.droite {  
    width: 71.4%; float: left;  
}  
.gaucheContent {  
    border: 0.07em solid gray;  
    margin: 0.3em;  
    padding: 0.2em 0.3em  
           0.4em 0.4em;  
}
```

HTML

```
<body>  
    <div class="gauche">  
        <div class="gaucheContent">  
            ...  
        </div>  
    </div>  
    <div class="droite">  
        ...  
    </div>  
</body>
```

Viewports

Apple a introduit le concept de viewport.

Pour un site compatible "responsive design", vous devez désactiver l'effet viewport, avec l'élément suivant dans le <head>

```
<meta content="initial-scale=1.0" name="viewport">
```



Plan

- 1 Introduction
- 2 Sélecteurs CSS
- 3 Propriétés CSS
- 4 Positionnement
- 5 Modules CSS3**

Parmi les extensions CSS3 (sous forme de modules) les plus abouties, on trouve :

- de nouveaux sélecteurs puissants (voir en particulier les pseudo-classes dans la section sur les sélecteurs)
- de nouvelles méthodes avancées pour définir les couleurs (voir la section sur les propriétés)
- des possibilités d'arrondis de coins et d'ombrage
- des possibilités de transitions (et animations)
- des possibilités de transformations des éléments
- le multi-colonne
- les media queries
- des dégradés linéaires ou radiaux de couleurs

La propriété `text-shadow` permet de créer une ombre sur un texte. Les paramètres sont dans l'ordre :

- décalage horizontal de l'ombre (vers la droite)
- décalage vertical de l'ombre (vers le bas)
- rayon du flou, correspondant à l'étendu du dégradé de la couleur opaque vers le transparent (optionnel)
- couleur de l'ombre

La propriété `box-shadow` permet de créer une ombre sur une boîte. Les paramètres sont dans l'ordre :

- décalage horizontal de l'ombre (vers la droite)
- décalage vertical de l'ombre (vers le bas)
- rayon du flou (optionnel)
- taille de l'ombre (optionnel)
- couleur de l'ombre

La propriété `border-radius` permet d'arrondir les coins des bordures en spécifiant l'étendue de l'arrondi. Elle prend :

- soit 4 valeurs pour les quatre coins (la première valeur correspond au coin supérieur gauche, puis on tourne dans le sens des aiguilles d'une montre) ;
- soit 2 valeurs (correspondant aux coins opposés) ;
- soit 1 valeur pour un même arrondi sur les 4 coins.

Attention

`box-shadow` et `border-radius` font partie de ces propriétés qui peuvent nécessiter une déclaration redondante sur la base de préfixes dits vendeurs (`-webkit-`, `-moz-`, `-ms-`, `-o-` selon les versions). Par exemple :

```
-webkit-border-radius:14px;  
-moz-border-radius:14px;  
-o-border-radius:14px;  
border-radius:14px;
```



```
p { padding:10px; margin:10px;
width:100px;
}
p:first-child {
text-shadow:2px 2px black;
}
p:nth-child(2) {
box-shadow:0px 0px 10px 3px red;
}
p:last-child {
border:solid 2px green;
border-radius:12px;
}
```

```
<p>Ceci est un premier paragraphe
qui bénéficie de text-shadow</p>
<p> Ceci est un second paragraphe
qui bénéficie de box-shadow</p>
<p> Ceci est un troisième paragraphe
qui bénéficie de border-radius</p>
```

**Ceci est un
premier
paragraphe
qui
bénéficie de
text-shadow**

Ceci est un
second
paragraphe
qui
bénéficie de
box-shadow

Ceci est un
troisième
paragraphe
qui bénéficie
de border-
radius

Le module “2D transforms” introduit la propriété transform qui permet la rotation, translation et scaling d’éléments (sans changer leurs places dans le flux du document). Sa syntaxe est :

```
transform:function(value);  
ou transform:function1(value1) function2(value2) ... ;
```

Les valeurs possibles pour les fonctions sont :

- translate(x,y), translateX(x), translateY(y),
- rotate(deg),
- scale(val), scale(val1,val2), scaleX(val), scaleY(val),
- skewX(deg), skewY(deg),
- matrix(...)



```
p {  
  width:150px; border:2px solid black;  
  padding:10px; margin:10px;  
  background-color:#EEEEEE;  
}  
p:nth-child(2) { transform:translateX(20px); }  
p:nth-child(3) { transform:rotate(45deg); }  
p:nth-child(4) { transform:scale(0.75); }  
p:nth-child(5) {  
  transform:skewX(10deg) skewY(10deg);  
}
```

```
<p> Ceci est un premier pagraphe</p>  
<p> Ceci est un second pagraphe</p>  
<p> Ceci est un troisième pagraphe</p>  
<p> Ceci est un quatrième pagraphe</p>  
<p> Ceci est un cinquième pagraphe</p>
```

Ceci est un
premier pagraphe

Ceci est un
second pagraphe

Ceci est un
troisième
pagraphe

Ceci est un
quatrième
pagraphe

Ceci est un
cinquième
pagraphe

Le principe est de permettre une transition douce entre l'ancienne valeur et la nouvelle valeur d'une propriété CSS lorsqu'un événement est déclenché :

- soit via une pseudo-classe telles que `:hover` ou `:focus`
- soit via JavaScript

Pour définir une transition, il est nécessaire de préciser au minimum :

- La ou les propriété(s) à animer
- La durée de l'animation

Exemple

```
p {  
  background-color:green;  
  padding: 5px;  
  /* Pour webkit */  
  -webkit-transition-property:background-color;  
  -webkit-transition-duration:2s;  
  /* Pour Firefox */  
  -moz-transition-property:background-color;  
  -moz-transition-duration:2s;  
  /* a terme uniquement */  
  transition-property: background-color;  
  transition-duration:2s;  
}  
p:hover, p:focus {  
  background-color:red;  
}
```

<p> Ceci est un premier pagraphe</p>

<p> Ceci est un second pagraphe</p>

<p> Ceci est un troisième pagraphe</p>

Le mode multi-colonne est possible maintenant, avec les propriétés :

- `column-count` qui indique le nombre de colonnes
- `column-width` qui permet de spécifier la largeur de chaque colonne
- `column-gap` qui indique l'écart entre deux colonnes
- `column-rule` pour le style de l'inter-colonne

Remarques

- Si `column-width` est fixé, alors `column-count` représente le nombre maximal de colonnes.
- La propriété `columns` est un raccourci pour `column-width` et `column-count`.
- Il est possible d'empêcher ou au contraire de forcer des sauts de colonnes avant, après ou au sein des éléments via les propriétés `break-before`, `break-inside` et `break-after`.



```
body { width:400px; }  
p {  
  height:200px; border: solid 2px black;  
  column-gap:20px;  
  column-rule: 3px dotted red;  
}  
p:first-child { width:300px; column-count:2; }  
p:last-child { width:100%; column-width:70px; }  
  
<p> Lorsqu'on sort de Plassans ...  
... d'aire Saint-Mittre. </p>  
<p> L'aire Saint-Mittre est un carré ...  
... de mousse. </p>
```

Lorsqu'on sort de Plassans par la porte de Rome, située au sud de la ville, on trouve, à droite de la route de Nice, après avoir dépassé les	premières maisons du faubourg, un terrain vague désigné dans le pays sous le nom d'aire Saint-Mittre.
--	---

L'aire Saint-Mittre est un carré long, d'une certaine étendue, qui s'allonge	au ras du trottoir de la route, dont une simple bande d'herbe usée la sépare. D'un côté,	à droite, une ruelle, qui va se terminer en cul-de-sac, la borde d'une rangée de mesures ;	à gauche et au fond, elle est close par deux pans de muraille rongés de mousse.
--	--	--	---

Les `media queries` étendent les règles `@media`, et aussi l'attribut `media`, en permettant de spécifier des restrictions d'usage en fonction de critères telle que la taille de l'écran ou le support de couleur par exemple. Le but est de cibler finement les périphériques de destination en fonction de leurs capacités intrinsèques.

Pour spécifier les restrictions, on utilise une expression booléenne sur la base des opérateurs logiques suivants :

- `and` (pour la conjonction, et logique),
- `only`
- `not` (pour la négation logique),
- `,` (pour la disjonction, ou logique)

Remarque

Chaque sous-expression est toujours entre parenthèses.

Example

```
<body>
  <div id="page">
    ...
  </div>
</body>

#page { width:1200px; color:black }

@media screen and (max-width:1000px) {
  #page { width:900px; color:red; }
}

@media screen and (max-width:800px) {
  #page { width:700px; color:blue; }
}

@media screen and (min-width:200px) and (max-width:640px) {
  #page { width:100%; color:yellow; }
}
```

La plupart des critères peuvent être préfixés par `min-` et `max-` lorsqu'elles acceptent des valeurs numériques pour définir des valeurs minimales ou maximales à respecter.

- `color` pour le nombre de bits par couleur et `color-index` pour le nombre de couleurs que le périphérique peut gérer.
- `device-aspect-ratio` pour le ratio largeur/hauteur du périphérique, et `aspect-ratio` pour la ratio de la zone d'affichage (par exemple 16/9)
- `device-height` et `device-width` pour la hauteur et la largeur du périphérique
- `height` et `width` pour la largeur de la zone d'affichage du périphérique
- `orientation`, qui peut être `portrait` ou `landscape`, et `resolution` pour la densité de pixels (en `dpi`, `dppx`, ou `dpcm`)
- `monochrome`, `scan` et `grid`

Voir [Alsacreation](#)s

Exclusion mutuelle des règles

Avec @media

```
/* Normal styles here */
@media (min-width: 600px) and (max-width: 700px) {
    /* Override the styles for 600-700 pixel windows. */
}
@media (min-width: 400px) and (max-width: 599.99px) {
    /* Override the styles for 400-600 pixel windows. */
}
@media (max-width: 399.99px) {
    /* Override the styles for sub-400 pixel windows. */
}
```

Avec <link> et l'attribut media

```
<link rel="stylesheet" media="(min-width: 568.01px)"
      href="standard.css">
<link rel="stylesheet" media="(max-width: 568px)"
      href="small_styles.css">
```

Pour l'impression

```
<link rel="stylesheet" href="NormalPrintStyles.css"
      media="print and (min-width: 25cm)" >
<link rel="stylesheet" href="NarrowPrintStyles.css"
      media="print and (not min-width: 25cm)" >
```

Pour l'Ipad

```
<link rel="stylesheet" href="iPad_portrait.css"
      media="(max-device-width: 768px) and (orientation: portrait)">
<link rel="stylesheet" href="iPad_landscape.css"
      media="(max-device-width: 768px) and (orientation: landscape)">
```

Pour la video

```
<video controls width="400" height="300">
  <source src="butterfly_mobile.mp4" type="video/mp4"
    media="(max-device-width: 480px)">
  <source src="butterfly.mp4" type="video/mp4">
  <source src="butterfly.ogv" type="video/ogg">
</video>
```


Consulter : <http://www.mydevice.io/devices/>

Appareil	Largeur	Hauteur
Apple iPhone 6	750	1334
Apple iPhone 5	640	1136
Apple iPod Touch	640	1136
Samsung Galaxy Note 3	1080	1920
Samsung Galaxy S4	1080	1920
Samsung Galaxy S4 mini	540	960
Samsung Galaxy S3	720	1280

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

La syntaxe est :

```
background: linear-gradient(color-stop1, color-stop2);  
background: linear-gradient(direction, color-stop1, color-stop2);  
background: linear-gradient(angle, color-stop1, color-stop2);
```

Remarque

Il est possible de spécifier plus de deux couleurs.



```
p:first-child {  
    background:linear-gradient(red,yellow); }  
p:nth-child(2) {  
    background:linear-gradient(to right,  
                                red,yellow); }  
p:nth-child(3) {  
    background:linear-gradient(to bottom right,  
                                red,yellow); }  
p:nth-child(4) {  
    background:linear-gradient(15deg,red,yellow); }  
p:nth-child(5) {  
    background:linear-gradient(red,yellow,green); }  
p:nth-child(6) {  
    background: linear-gradient(to right,red,  
                                orange,yellow,green,blue,indigo,violet); }  
  
<p>Voici un premier pagaraphe. </p>  
<p>Voici un second paragraphe. </p>  
<p>Voici un troisième paragraphe. </p>  
<p>Voici un pagaraphe. </p>  
<p>Voici un paragraphe. </p>  
<p>Voici un paragraphe. </p>
```

Voici un premier
pagaraphe.

Voici un second
paragraphe.

Voici un troisième
paragraphe.

Voici un
pagaraphe.

Voici un
paragraphe.

Voici un
paragraphe.

Un dégradé radial est défini par son centre.

La syntaxe est :

```
background: radial-gradient(color-stop1, color-stop2);  
background: radial-gradient(shape, color-stop1, color-stop2);  
background: repeating-radial-gradient(color-stop1, color-stop2);  
...
```



```
p:first-child {  
  background:linear-gradient(to right,  
    rgba(255,0,0,0), rgba(255,0,0,1)); }  
p:nth-child(2) {  
  background:repeating-linear-gradient(red,  
    yellow 10%, green 20%); }  
p:nth-child(3) {  
  background: radial-gradient(red, yellow,  
    green); }  
p:nth-child(4) {  
  background: radial-gradient(red 5%,  
    yellow 15%, green 60%); }  
p:nth-child(5) {  
  background: radial-gradient(closest-side  
    at 60% 55%,red,yellow,green,black); }  
p:nth-child(6) {  
  background: repeating-radial-gradient(red,  
    yellow 10%, green 15%); }
```

Voici un premier
paragraphe.

Voici un second
paragraphe.

Voici un troisième
paragraphe.

Voici un
paragraphe.

Voici un
paragraphe.

Voici un
paragraphe.