

Créer son site Web

HTML et CSS



Page 1 : Introduction

Page 2 : Language HTML

Page 7 : Le CSS

Page 14 : MementoHTML5 et CSS3

Introduction

Les langages HTML et CSS sont à la base du fonctionnement de tous les sites web. Quand vous consultez un site avec votre navigateur, il faut savoir que, en coulisses, des rouages s'activent pour permettre au site web de s'afficher. L'ordinateur se base sur ce qu'on lui a expliqué en HTML et CSS pour savoir ce qu'il doit afficher.

Langages HTML et CSS



Traduction par
l'ordinateur



Résultat visible à l'écran



Pour expliquer à l'ordinateur ce que vous voulez faire, il va falloir utiliser un langage qu'il comprend donc apprendre deux langages qui ont des rôles différents :

- **HTML (HyperText Markup Language)** : il a fait son apparition dès 1991 lors du lancement du Web. Son rôle est de gérer et organiser le contenu. C'est donc en HTML que vous écrirez ce qui doit être affiché sur la page : du texte, des liens, des images...
- **CSS (Cascading Style Sheets, aussi appelées Feuilles de style)** : le rôle du CSS est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...). Ce langage est venu compléter le HTML en 1996.

HTML (pas de CSS)



HTML + CSS



On peut ajouter également des scripts en langage JavaScript pour obtenir un comportement dynamique ou du PHP.

Attention : Les langages HTML et CSS font partie des langages de description (contrairement à JavaScript qui est un langage de programmation).

Dans la suite de ce cours, nous utiliserons la version HTML5.(actuellement, la version la plus utilisée)

Les langages HTML et XHTML sont largement compatibles, XHTML possédant une syntaxe plus rigoureuse.

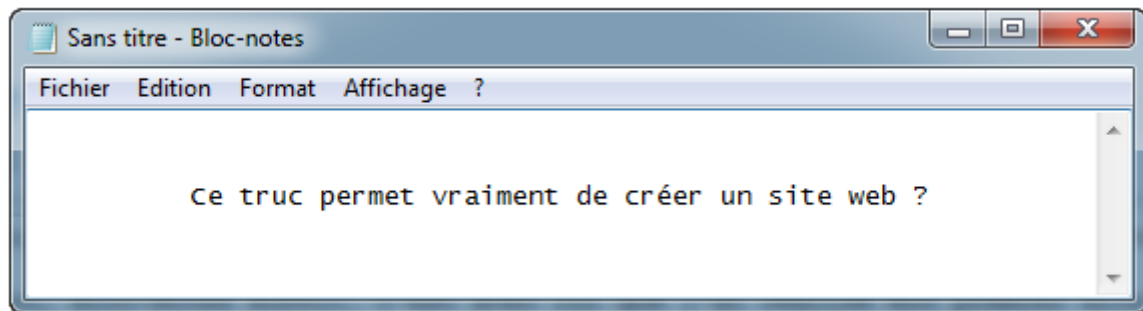
Langage HTML

- Les outils pour coder du HTML

Le code d'une page HTML est stocké dans un fichier texte avec l'extension `.htm` ou `.html`

Un éditeur de texte est donc nécessaire pour écrire le code source : On pourra se servir entre autre, d'un éditeur de texte comme le **Bloc-notes** de Windows ou **NotePad++** .

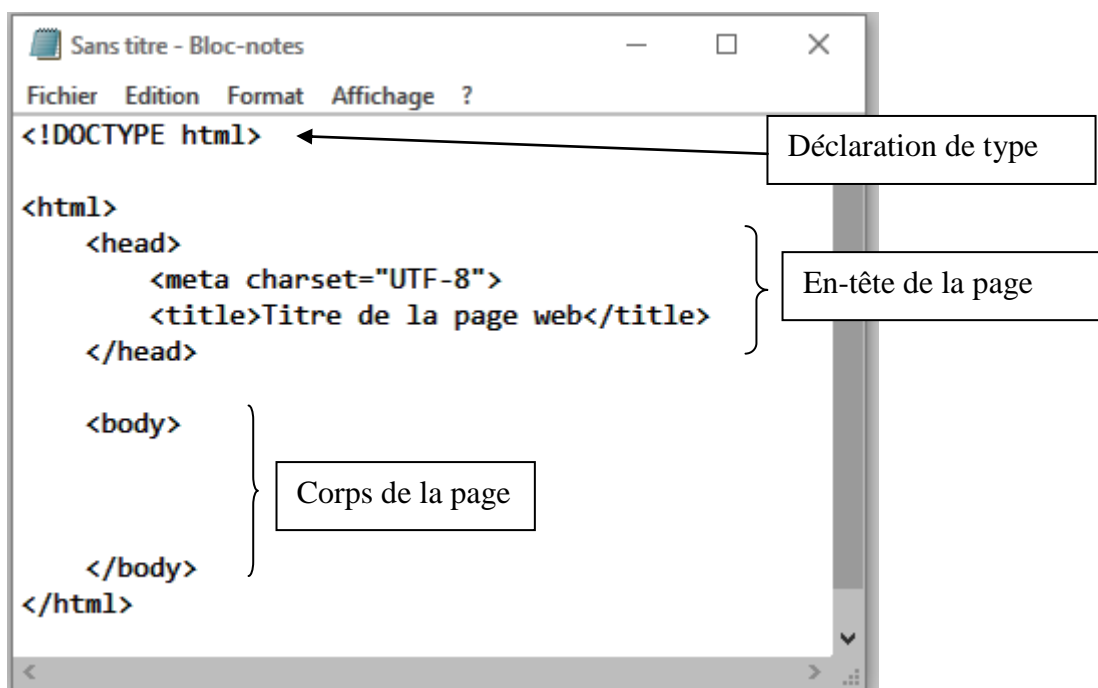
Pour interpréter le code HTML, il faut simplement un navigateur Web (c'est-à-dire un client HTTP HyperText Transfer Protocol).



- Structure minimum d'un document HTML

En HTML, la structure matérielle d'un document est décrite à l'aide de **balises**.

Un document HTML doit comporter une en-tête (balise `<head>`), un corps (balise `<body>`) et avant tout une déclaration de type de document précisant la version de la norme HTML utilisée.



Le Doctype

La toute première ligne s'appelle le doctype. Elle est indispensable car c'est elle qui indique qu'il s'agit bien d'une page web HTML et elle en précise la version. Ce n'est pas vraiment une balise comme les autres (elle commence par un point d'exclamation).

Quand vous voyez la balise doctype (`<!DOCTYPE html>`), cela signifie que la page est écrite en HTML5.

La balise <html>

C'est la balise principale du code. Elle englobe tout le contenu de la page. Comme on peut le voir, la balise fermante `</html>` se trouve tout à la fin du code !

L'en-tête <head>

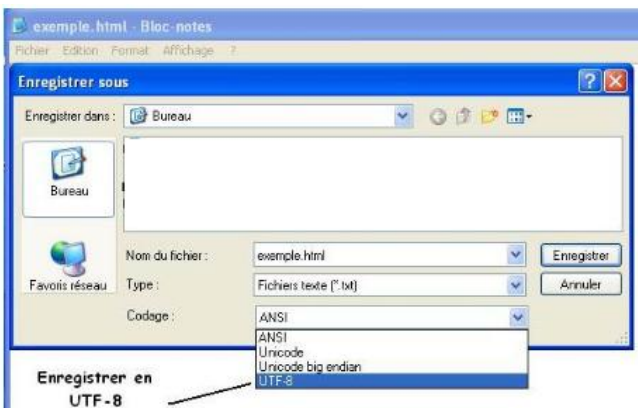
Cette section donne quelques informations générales sur la page comme son titre (balise `<title>`), l'encodage (pour la gestion des caractères spéciaux), etc..



Les informations que contient l'en-tête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes !

⇒ L'encodage (charset) :

La balise `<meta charset="utf-8" />` indique l'encodage utilisé dans le fichier.html.



Il y a plusieurs techniques d'encodage en fonction des langues : ISO-8859-1, OEM 775, Windows-1253... Une seule cependant devrait être utilisée aujourd'hui autant que possible : UTF-8. Cette méthode d'encodage permet d'afficher sans aucun problème pratiquement tous les symboles de toutes les langues de notre planète ! (accents, idéogrammes chinois et japonais, caractères arabes, etc.).

Il faut aussi que le fichier soit bien enregistré en UTF-8. C'est le cas le plus souvent sous Linux par défaut mais, sous Windows, il faut généralement le dire au logiciel.

⇒ Le titre principal de la page :

C'est le titre de la page, probablement l'élément le plus important ! Toute page doit avoir un titre qui décrit ce qu'elle contient. Il est conseillé de garder le titre assez court (moins de 100 caractères en général).



Le titre ne s'affiche pas dans la page mais en haut de celle-ci (souvent dans l'onglet du navigateur).

Le corps <body>

C'est là que se trouve la partie principale de la page. Tout ce qui est écrit ici sera affiché à l'écran. C'est à l'intérieur du corps que l'on écrit la majeure partie du code.

• Les balises des éléments de bloc

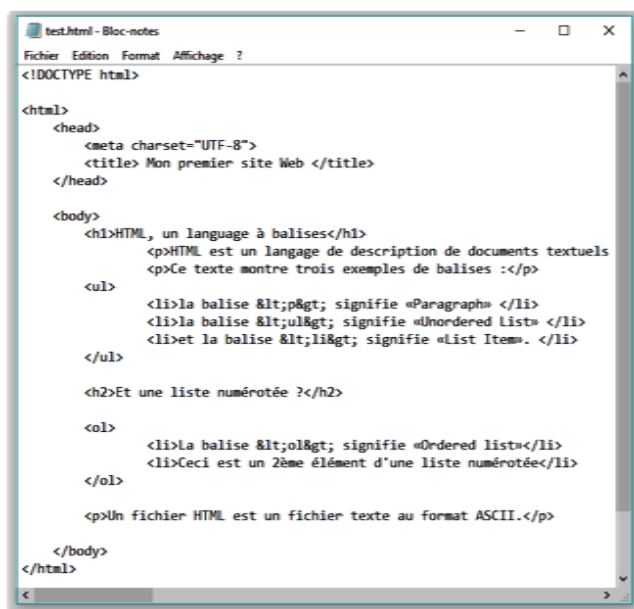
De nombreuses balises permettent de spécifier dans quelle catégorie de mise en page se trouve un texte (est-ce qu'il s'agit d'un titre, d'un paragraphe ou d'une liste... ?). Ces balises délimitent les éléments dits « de bloc ».

Tout texte ou élément présent dans une page HTML doit être dans une balise d'élément de bloc. Ces balises doivent être ouvertes et fermées. Les blocs ainsi générés par ces balises sont indépendants et structurent les pages web. Il y a un saut de ligne avant et après leur contenu. Par défaut, deux éléments de bloc sont l'un en dessous de l'autre.

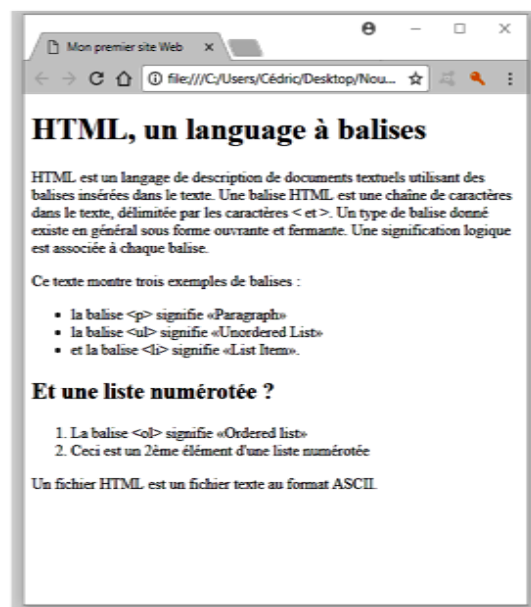
Exemples de balises HTML délimitant des éléments de bloc :

```
<h1>Ceci est un titre de niveau 1</h1>
<h2>Ceci est un titre de niveau 2</h2>
<h3>Ceci est un titre de niveau 3</h3>
<p>Ceci est un paragraphe</p>
<ul>
  <li>Ceci est un élément d'une liste non numérotée</li>
  <li>Ceci est un 2ème élément d'une liste non numérotée</li>
</ul>
<ol>
  <li>Ceci est un élément d'une liste numérotée</li>
  <li>Ceci est un 2ème élément d'une liste numérotée</li>
</ol>
```

Language HTML



Interprétation du code dans un navigateur Web



Un fichier HTML est un fichier texte structuré. La structure est obtenue grâce à l'utilisation de **balises** (markup) du langage. On parlera d'**éléments** :

L'élément p est utilisé pour marquer un paragraphe.

L'élément ul pour commencer une liste d'item non numérotés.

L'élément li introduit un item de liste.

Grâce à ces éléments de structuration, un logiciel (en général des navigateurs, mais aussi les moteurs de recherche) pourra plus facilement traiter le texte.

Dans le fichier HTML, les caractères < et > qui appartiennent au texte de base mais qui ne sont pas utilisés pour des balises ont dû être remplacés par des séquences de caractères appelées « **entités HTML** » de sorte qu'ils ne soient pas interprétés par les logiciels comme des éléments du langage HTML.

Une entité en HTML commence par le caractère '&' (esperluette) et termine par le caractère ';'.

&lt;	(less than)	représente <
&gt;	(greater than)	représente >
&amp;	(ampersand)	représente &



Pratiquez : Réalisez une première page HTML simple avec des titres, quelques paragraphes, etc....

- Les balises des éléments en-ligne (inline)

Une balise de type inline se trouve obligatoirement à l'intérieur d'une balise de bloc. Une balise inline ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne (c'est pour cela que l'on parle de balise « en ligne »).

Les balises inline ont plusieurs fonctions comme préciser la mise en forme de parties du texte (mise en valeur, en exposant, en indice, en citation ...), la création de liens hypertextes ou encore l'insertion d'images.

Exemples :

Gras	[Bold]	...	Début et fin de zone en gras
		...	
Italique	[Italic]	<I>...</I>	Début et fin de zone en italique
		...	
A la ligne	[Line break]	 	Aller à la ligne
Commentaires	[Comments]	<!-- *** -->	Ne pas afficher
Centrage	[Center]	<CENTER></CENTER>	Centrer



Pratiquez : Ouvrir l'éditeur de texte et taper ce qui suit.

```
test2.html - Bloc-notes
Fichier Edition Format Affichage ?
<!DOCTYPE html>

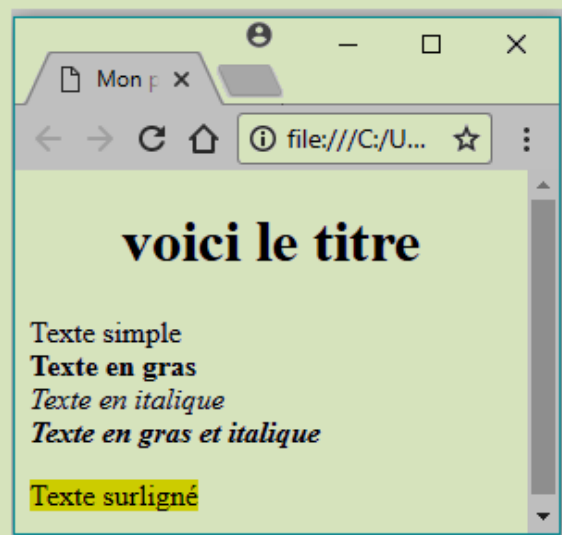
<html>
  <head>
    <meta charset="UTF-8">
    <title> Mon premier site Web </title>
  </head>

  <body>
    <h1><center>voici le titre</center></h1>
    <p>Texte simple<br>
    <b>Texte en gras</b><br>
    <i>Texte en italique</i><br>
    <i><b>Texte en gras et italique</b></i><br></p>

    <!-- Ceci est un autre paragraphe -->
    <p><mark>Texte surligné</mark></p>

  </body>
</html>
```

Avec un navigateur Web, Vous pouvez lire ceci :



En fait, c'est le navigateur qui choisit comment afficher les mots. On lui dit que les mots sont assez importants et, pour faire ressortir cette information, il change l'apparence du texte en utilisant l'italique par exemple. On pourra surtout modifier cette apparence avec le CSS.



Utiliser le *mémento HTML* dans lequel sont répertoriées un plus grand nombre de balises. (page 14)

• Insérer une image

La balise qui permet d'insérer une image est la balise orpheline ``. Elle doit être accompagnée de deux attributs obligatoires :

- `src` : il permet d'indiquer où se trouve l'image que l'on veut insérer. On peut soit mettre un chemin absolu (ex. : `http://www.site.com/fleur.jpg`), soit mettre le chemin en relatif (ce qu'on fait le plus souvent). Ainsi, si l'image est dans un sous-dossier images, on écrit : `src="images/fleur.jpg"`.
- `alt` : cela signifie « texte alternatif ». On doit toujours indiquer un texte alternatif à l'image, c'est-à-dire un court texte qui décrit ce que contient l'image. Ce texte sera affiché à la place de l'image si celle-ci ne peut pas être téléchargée (cela arrive), ou dans les navigateurs de personnes handicapées. Cela aide aussi les robots des moteurs de recherche pour les recherches d'images. Pour la fleur, on mettrait par exemple : `alt="Une fleur"`.

On a alors la syntaxe : ``

• Lien hypertexte

Pour faire un lien, la balise à utiliser est : `<a>`. Il faut cependant lui ajouter un attribut, `href`, pour indiquer vers quelle page le lien doit conduire.

Par exemple, pour créer un lien renvoyant à la page Wikipedia sur le HTML5 lorsqu'on clique sur le mot HTML5, on écrira :

` HTML5 `

Pour faire un lien vers un autre site, il suffit donc de copier son adresse (ou URL) dans la balise. On parle alors de **lien absolu** car l'adresse donnée est l'adresse complète.



Par défaut, le lien s'affiche en bleu souligné. Si on a déjà ouvert la page, le lien s'affiche en violet. Cette apparence peut être modifiée avec le CSS.

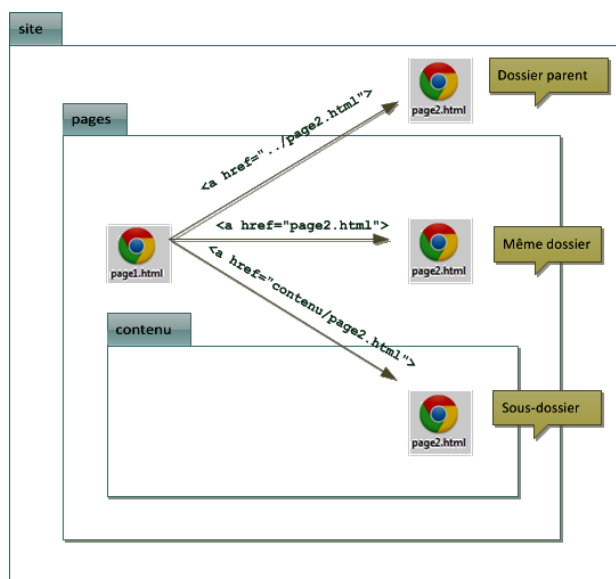
Pour créer un lien entre différentes pages de son propre site, il suffit de créer 2 pages dans un même dossier (page1.html et page2.html par exemple), puis dans la page 1, on crée le lien suivant :

`<p> la page 2</p>`

Dans le cas où les 2 pages ne sont pas dans le même dossier, on indiquera le lien comme sur le schéma ci-contre :



La page principale d'un site est toujours nommée *index* par convention



• Outil de vérification syntaxique

Pour vérifier que votre page Web est conforme aux spécifications HTML5, rendez vous sur le site du W3C (World Wide Web Consortium) : `http://validator.w3.org`

Pour une page Web locale (pas encore publiée sur le Web), il faut choisir :

Validate by File Upload → cliquer sur Parcourir et rechercher le fichier à tester puis → Check

S'il y a des erreurs, elles vous seront indiquées, avec des explications (en anglais, of course).

Conseil : vérifier et corriger systématiquement vos pages Web avec cet outil. C'est contraignant au début, mais cela vous fera prendre rapidement de bonnes habitudes.



Pratiquez : Dans un dossier nommé « Mon_site », réalisez une page html nommée « index.html » dans laquelle doit figurer un titre de niveau 1, du texte, une photo et un lien vers la page d'accueil de GOOGLE et un lien vers une autre page nommée « page2.html » que vous développerez.

Le CSS

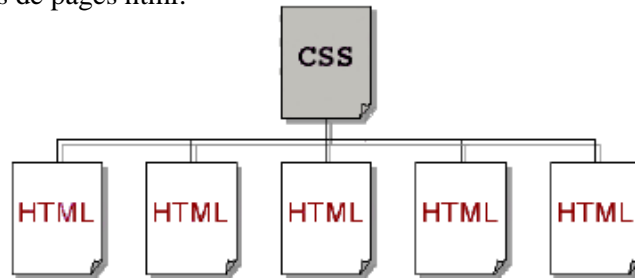
• Introduction

Nous avons appris les principales balises du HTML, ce qui nous permet de créer une page web "brute" avec des liens mais dépouillée, sans aucune décoration ...donc plutôt laide!

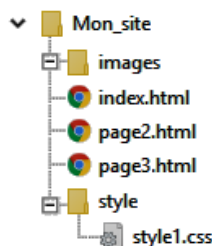
La mise en forme du texte se fait à l'aide des "feuilles de style" et du langage css. (Cascade Style Sheets)

Il y a deux façons de procéder :

1. écrire les instructions de style dans la page html, avec la balise <style> (déconseillé)
2. traiter les styles dans un fichier séparé, ce qui est de loin préférable, car avec un seul fichier on peut modifier la présentation de centaines de pages html.



⇒ On crée un nouveau fichier dans l'éditeur Bloc-notes ou Notepad++ :



Par exemple, notre feuille de style se nomme « **style1.css** » et se trouve dans le dossier appelé « **style** » lui-même dans le dossier « Mon_site » dans lequel se trouvent les autres éléments du site Web comme les pages html.

⇒ On ajoute le lien vers la feuille de style entre les balises <head> et </head> de chaque pages html :

`<link rel="stylesheet" href="style/style1.css" />`

• Principe du code CSS

La syntaxe de base du code CSS se compose de 3 parties :

```
selector {property: value;}
    ↑           ↑           ↑
    À quelle(s)  La propriété, par  La valeur de la
    balise(s) HTML  exemple, la couleur  propriété de couleur
    la propriété s'applique  d'arrière-plan, par  d'arrière-plan, par
    (p.ex. "body")  exemple, la couleur  exemple, le rouge
                    d'arrière-plan  ("#FF0000")
                    {"background-color"}
```

• Couleurs et arrière-plans

⇒ Couleur de l'arrière-plan [background-color] :

```
body {background-color: yellow;}
```

Le style appliqué à la balise <body> est écrit entre accolades, avec un point-virgule à la fin de chaque instruction. Il est agréable de pratiquer une indentation, et de passer à la ligne après chaque instruction, mais ce n'est pas obligatoire!

Il existe des couleurs prédéfinies (red, black, ...) et on peut aussi utiliser le code rgb (rouge, vert bleu):

```
body {background-color : rgb(250,120,0);}
```


⇒ Image d'arrière-plan [background-image] :

On peut aussi utiliser une image de fond.

Attention : l'image doit être dans le même dossier que le fichier css, ou dans un sous dossier auquel cas il faut indiquer le chemin (exemple : "images/image.jpg") :

```
body {background-image : url("images/image.jpg ");}
```

⇒ Répéter l'image d'arrière-plan [background-repeat] :

Cette propriété permet de répéter ou non une image d'arrière-plan (en général de petite taille), pour faire par exemple un effet mosaïque.

```
body {  
    background-image: url("image.jpg");  
    background-repeat : repeat ;  
}
```

Le tableau suivant décrit les différentes valeurs de cette propriété :

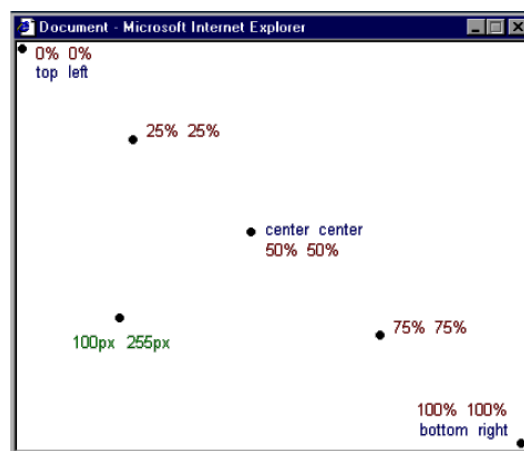
Valeur	Description
background-repeat: repeat-x	L'image se répète horizontalement
background-repeat: repeat-y	L'image se répète verticalement
background-repeat: repeat	L'image se répète horizontalement et verticalement
background-repeat: no-repeat	L'image ne se répète pas

⇒ Positionner une image d'arrière-plan [background-position] :

Par défaut, l'image d'arrière-plan se positionnera dans le coin supérieur gauche de l'écran. La propriété *background-position* permet de changer cette position prédéfinie et de placer l'image d'arrière-plan n'importe où à l'écran.

Il y a plusieurs méthodes pour fixer la valeur de *background-position*. Par contre, elles se présentent toutes sous la forme d'un jeu de coordonnées. Par exemple, la valeur '100px 200px' positionne l'image d'arrière-plan à 100px depuis la gauche et à 200px depuis le haut de la fenêtre du navigateur.

Les coordonnées peuvent s'exprimer en pourcentages de la largeur de l'écran, ou en unités fixes (pixels, centimètres, etc.), ou on peut utiliser les mots-clés "top", "bottom", "center", "left" ou "right". Le modèle suivant illustre ce système :



Le tableau suivant fournit quelques exemples de valeur :

Valeur	Description
background-position: 2cm 2cm	L'image est positionnée à 2 cm de la gauche et à 2 cm du haut de la page
background-position: 50% 25%	L'image est positionnée au centre et à un quart de la page vers le bas
background-position: top right	L'image est positionnée au coin supérieur droit de la page

Exemple : Ce code positionne l'image d'arrière-plan dans le coin inférieur droit de la page :

```
body {  
    background-color: yellow;  
    background-image: url("image.jpg");  
    background-repeat : no-repeat ;  
    background-position : right bottom;  
}
```



On peut obtenir la même chose en utilisant la concision :

```
body { background: yellow url("image.jpg") no-repeat right bottom;}
```

- Mise en forme du texte

⇒ La couleur des textes [color] :

Exemple : Tous les titres h1 seront en rouge.

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

On peut aussi agir sur la couleur d'arrière plan :

```
h1 {color : red; background-color : blue ;}
```

⇒ Polices d'écriture [font-family] :

font-family permet d'indiquer, par ordre de préférence, la police désirée.

Puisque tous les utilisateurs n'ont pas forcément les mêmes polices que nous, il faut toujours que la liste contienne des polices classiques, et même une police "générique" (serif ou sans-serif).

Exemple : Tous les titres h1 seront écrits en arial, sinon en verdana, :

```
h1 {font-family : arial, verdana, sans-serif ;}
```

Times New Roman Garamond Georgia	Ces trois familles de polices appartiennent au genre serif . Elles se caractérisent par leurs « empattements »
Trebuchet Arial Verdana	Ces trois familles de polices appartiennent au genre sans serif . Elles se caractérisent par l'absence d'« empattements »
Courier Courier New Andale Mono	Ces trois familles de polices appartiennent au genre monospace . Elles se caractérisent par une chasse fixe

⇒ Mise en valeur du texte [font] :

- **font-style** possède 2 options : *normal* et *italic*
- **font-weight** possède 2 options : *normal* et *bold* (écriture en gras)
- **font-size** permet de préciser la taille de la police.

Cette taille peut être exprimée de plusieurs façons : en px, pt, % et em. px et pt sont des mesures absolues, il vaut mieux choisir % ou em qui indiquent une taille relative à la taille normale choisie par l'utilisateur.

Exemple : Les caractères des paragraphes ont une taille égale à 120% de la taille normale, les caractères des liens sont doublés

```
p {font-size:120%;}  
a {font-size : 2em;}
```



On peut obtenir la même chose en utilisant la concision :

```
h1 {font : italic bold 30px verdana,sans-serif;}
```

⇒ Compléments de mise en forme [text] :

- **text-align** permet de préciser la position du texte, les valeurs étant *left*, *right*, *center*, *justify*.
- **text-decoration** a 3 options : *underline*, *overline* et *line-through*.

Exemple : Les titres h2 seront centrés et soulignés

```
h2 {  
  text-align:center ;  
  text-decoration:underline;  
}
```



Pratiquez : Dans le dossier « mon site », créer un fichier nommé « style1.css » dans lequel vous pouvez modifier la couleur de fond ainsi que la couleur du titre de niveau 1 des pages html réalisées précédemment.

- Mise en forme des liens

Le CSS permet de définir des propriétés différemment, selon que le lien est visité, non visité, activé, ou si le curseur le survole. Cela permet d'ajouter des effets plaisants et utiles à vos sites Web. Pour contrôler ces effets, on utilise ce qu'on appelle des pseudo-classes

Une pseudo-classe permet de tenir compte de conditions et événements différents pour la définition d'une propriété sur une balise HTML

- La pseudo-classe : **link** est utilisée pour les liens menant aux pages que l'utilisateur n'a pas visitées
- La pseudo-classe : **visited** est utilisée pour les liens menant aux pages que l'utilisateur a visitées
- La pseudo-classe : **active** est utilisée pour les liens qui sont activés
- La pseudo-classe : **hover** est utilisée lorsque le pointeur de la souris survole un lien

Exemple : Voici un code css qui écrit les liens en rouge, en vert surligné sur fond blanc lorsque la souris passe dessus, en noir lorsqu'ils ont été visités.

```
a {color : red;}
a:hover{ text-decoration: overline; color: green; background-color : white;}
a:visited {color : black;}
```

- Les identifiants

Parfois, on veut appliquer un style spécial à un élément ou à un groupe d'éléments particuliers. Il existe 2 types d'identifiants pour appliquer un style à une partie précise d'un document :

- l'identifiant **id**, qui ne concerne qu'un élément unique,
- l'identifiant **class**, qui peut s'appliquer à plusieurs éléments.

Exemple : On veut écrire en gras, plus gros et en bleu le paragraphe :

<p>On peut faire ressortir une équation: $-3x + 5 = 9$ </p>

- Dans le code html, dans la balise ouvrante de paragraphe on ajoute id="equation" :

```
<p id="equation">On peut faire ressortir une belle équation:  $-3x + 5 = 9$ </p>
```

- Dans le fichier css on ajoute :

```
#equation{color:blue; font-weight:bold;}
```

Remarques : dans le fichier css :

- Devant le nom de l'**id** on ajoute un **dièse** (#)
- Devant le nom de la **class** on ajoute un **point** (.)

Voici un code qui va mettre une partie des liens en vert sur fond blanc lorsqu'ils sont inactifs, en blanc sur fond vert lorsqu'ils sont survolés par la souris, en noir sur fond blanc lorsqu'ils ont été visités.

- Code html : Cliquer ici

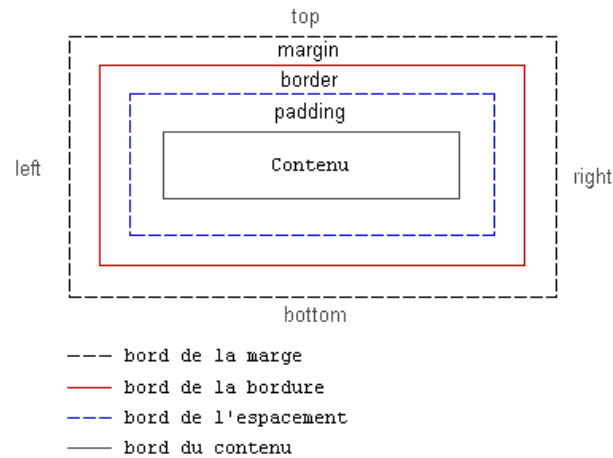
- Code CSS :

```
. lien_C_Escoute {color: green;background-color:white ;}
. lien_C_Escoute:hover {color:white;background-color:green;}
. lien_C_Escoute:visited {color:black;background-color:white;}
```

- Les boîtes

Le modèle des boîtes de CSS décrit les boîtes générées pour les éléments HTML. Il contient également des options détaillées pour l'ajustement des marges, des bordures, des espacements et du contenu de chaque élément.

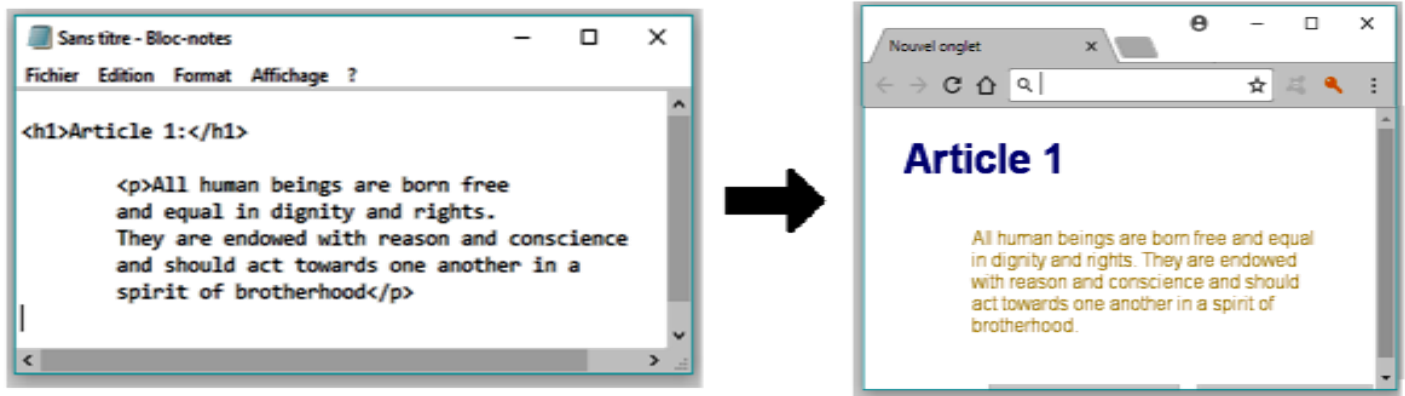
Le diagramme suivant montre la structure du modèle des boîtes :



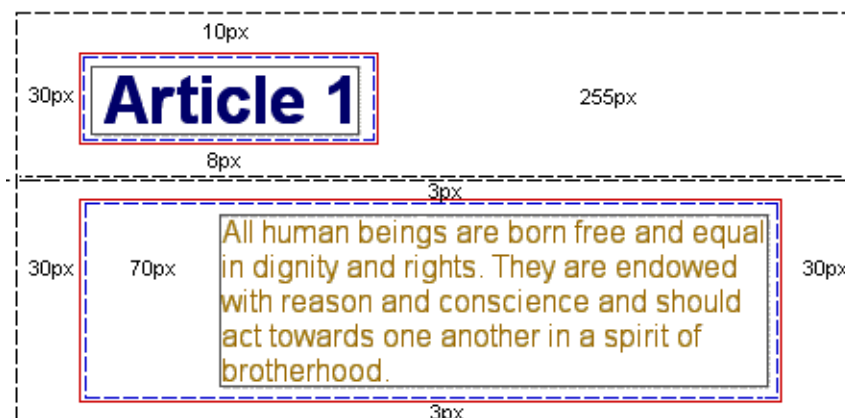
Exemple : utilisation des boîtes dans un cas réel avec un titre et du texte :

Le code HTML est le suivant :

En ajoutant de la couleur et des informations de police, l'exemple pourrait se présenter comme cela :



L'exemple contient deux éléments : `<h1>` et `<p>`. On peut illustrer le modèle de boîtes des deux éléments de la façon suivante :



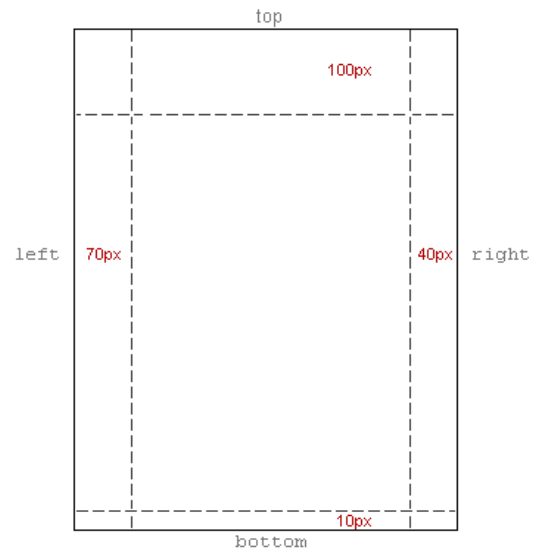
Les propriétés qui régissent les différentes boîtes sont padding, margin et border.

⇒ Marge d'un élément [margin] :

Un élément a quatre côtés : droit ("right"), gauche ("left"), supérieur ("top") et inférieur ("bottom"). La marge est la distance entre chaque côté et l'élément avoisinant

Exemple : Nous voulons définir les marges du document même, c'est-à-dire de l'élément <body>.

L'illustration ci-contre montre les marges voulues pour les pages.



Le code CSS serait le suivant :

```
body {  
    margin-top: 100px;  
    margin-right: 40px;  
    margin-bottom: 10px;  
    margin-left: 70px;  
}
```

Ou plus simplement :

```
body {margin: 100px 40px 10px 70px;}
```

Vous pouvez fixer pareillement les marges sur presque tous les éléments. Par exemple, on peut choisir de définir les marges de tous les paragraphes de texte balisés avec <p> :

```
body {margin: 100px 40px 10px 70px;}  
  
p {margin: 5px 50px 5px 50px;}
```

⇒ Espacement dans un élément [padding] :

L'espacement peut aussi être assimilé à un « remplissage ». Cela peut se comprendre dans la mesure où l'espacement n'affecte pas la distance de l'élément aux autres éléments mais définit seulement la distance interne entre la bordure et le contenu de l'élément.

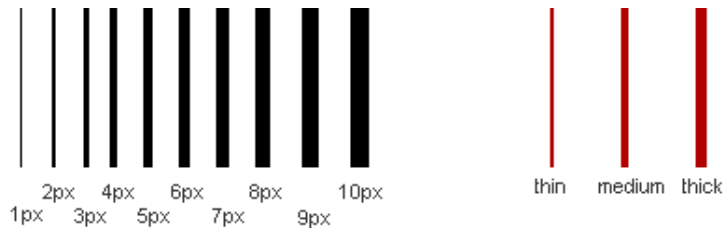
Exemple : En définissant l'espacement des titres, on change le remplissage autour du texte dans chaque titre :

```
h1 {  
    background: yellow;  
    padding: 20px 20px 20px 80px;  
}  
  
h2 {  
    background: orange;  
    padding-left: 120px;  
}
```

⇒ Les bordures [border] :

- L'épaisseur des bordures : **border-width**

L'épaisseur des bordures est définie par la propriété `border-width`, dont les valeurs peuvent être "thin", "medium" et "thick", ou une valeur numérique en pixels. La figure ci-dessous illustre le système :



- Couleur des bordures : **border-color**



La propriété `border-color` définit la couleur d'une bordure.

Des valeurs de couleur normales sont, par exemple : "#123456", "rgb(123,123,123)" ou "yellow" .

- Types de bordures : **border-style**

On peut choisir parmi plusieurs types de bordures. Voici huit types de bordures différents tels qu'Internet Explorer 5.5 les interprète.



Toutes les bordures ont la couleur "gold" et une épaisseur valant "thick" mais elles peuvent naturellement s'afficher dans d'autres couleurs et épaisseurs.

On peut utiliser les valeurs "none" ou "hidden" si on ne veut aucune bordure.

Exemple : Les trois propriétés décrites précédemment peuvent être réunies pour chaque élément et donc produire des bordures différentes.

Pour illustrer cela, voyons un document où nous définissons des bordures différentes pour `<h1>`, `<p>`.`` .

```
h1 {  
    border-width: thick;  
    border-style: dotted;  
    border-color: gold;  
}  
  
p {  
    border-width: 1px;  
    border-style: dashed;  
    border-color: blue;  
}  
  
ul {  
    border-width: thin;  
    border-style: solid;  
    border-color: orange;  
}
```



Utiliser le **mémento CSS** dans lequel sont répertoriées un plus grand nombre de codes. (*page 16*)



Pratiquez : faites votre site Web à présent

BASE DU DOCUMENT

Balise	Description
<html>	Balise principale
<head>	En-tête de la page
<body>	Corps de la page

EN TÊTE (entre <head> et </head>)

<link />	Liaison avec une feuille de style
<meta />	Métadonnées de la page web (charset, mots-clés, etc.)
<script>	Code JavaScript
<style>	Code CSS
<title>	Titre de la page

EXEMPLE DE CODE MINIMAL AVEC CSS

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8" />
  <title>Le titre de la page</title>
  <link rel="stylesheet" href="mon_style.css"/>
</head>
<body>
  .....
  .....
</body>
</html>
```

Remarque: pour ne pas avoir de problèmes avec les caractères spéciaux et les accents, choisir le jeu de caractères (charset) **utf-8**, sans oublier d'enregistrer le fichier html en utf-8 sans BOM.

STRUCTURATION DU TEXTE

<abbr>	Abréviation
<blockquote>	Citation (longue)
<cite>	Citation du titre d'une œuvre ou d'un événement
<q>	Citation (courte)
<sup>	Exposant
<sub>	Indice
	Mise en valeur forte
	Mise en valeur normale
<mark>	Mise en valeur visuelle
<h1> à <h6>	Titres de niveau 1 à 6
<h2>	Titre de niveau 2
<h3>	Titre de niveau 3
<h4>	Titre de niveau 4
<h5>	Titre de niveau 5
<h6>	Titre de niveau 6
	Image

<figure>	Figure (image, code, etc.)
<figcaption>	Description de la figure
<audio>	Son
<video>	Vidéo
<source>	Format source pour les balises <audio> et <video>
<a>	Lien hypertexte
 	Retour à la ligne
<p>	Paragraphe
<hr />	Ligne de séparation horizontale
<address>	Adresse de contact
	Texte supprimé
<ins>	Texte inséré
<dfn>	Définition
<kbd>	Saisie clavier
<pre>	Affichage formaté (pour les codes sources)
<progress>	Barre de progression
<time>	Date ou heure

LISTES

TABLEAUX

	Liste à puces, non numérotée
	Liste numérotée
	Élément de la liste à puces
<dl>	Liste de définitions
<dt>	Terme à définir
<dd>	Définition du terme

<table>	Tableau
<caption>	Titre du tableau
<tr>	Ligne de tableau
<th>	Cellule d'en-tête
<td>	Cellule
<thead>	Section de l'en-tête du tableau
<tbody>	Section du corps du tableau
<tfoot>	Section du pied du tableau

FORMULAIRES

<form>	Formulaire
<fieldset>	Groupe de champs
<legend>	Titre d'un groupe de champs
<label>	Libellé d'un champ
<input />	Champ de formulaire (texte, mot de passe, case à cocher, bouton, etc.)
<textarea>	Zone de saisie multiligne
<select>	Liste déroulante
<option>	Élément d'une liste déroulante
<optgroup>	Groupe d'éléments d'une liste déroulante

BALISE DE SECTIONS

Ces balises permettent de construire le squelette d'un site Web.

<header>	En-tête
<nav>	Liens principaux de navigation
<footer>	Pied de page
<section>	Section de page
<article>	Article (contenu autonome)
<aside>	Informations complémentaires

BALISE GÉNÉRIQUES

Les balises génériques sont des balises qui n'ont pas de sens sémantiques.

En effet, toutes les autres balises ont un sens : <p> signifie « paragraphe », <h2> signifie « sous-titre », etc...

Parfois, on a besoin d'utiliser des balises génériques (aussi appelées balises universelles) car aucune des autres ne convient. On utilise le plus souvent des balises génériques pour construire son design.

	Balise générique de type inline
<div>	Balise générique de type block

Elles ont un intérêt uniquement si vous associez un attribut class, id ou style :

- **Class** : Indique un nom de classe CSS à utiliser
- **Id** : Donne un nom à la balise. Ce nom doit être unique sur toute la page car il permet d'identifier la balise. Vous pouvez vous servir de l'ID pour de nombreuses choses, par exemple pour créer un lien vers une ancre, pour un style CSS de type ID, pour des manipulations en JavaScript, etc...
- **Style** : Cet attribut vous permet d'identifier directement le code CSS à appliquer. Vous n'êtes donc pas obligés d'avoir une feuille de style à part, vous pouvez mettre directement les attributs CSS.

TEXTES

Propriété	Valeurs (exemples)	Description
font-family	<i>police1, police2, police3, serif, sans-serif, monospace</i>	Nom de police
@font-face	<i>Nom et source de la police</i>	Police personnalisée
font-size	1.3em, 16px, 120%...	Taille du texte
font-weight	bold, normal	Gras
font-style	italic, oblique, normal	Italique
text-decoration	underline, overline, line-through, blink, none	Soulignement, ligne au-dessus, barré ou clignotant
font-variant	small-caps, normal	Petites capitales
text-transform	capitalize, lowercase, uppercase	Capitales
font	-	Super propriété de police. Combine : font-weight, font-style, font-size, font-variant, font-family.
text-align	left, center, right, justify	Alignement horizontal
vertical-align	baseline, middle, sub, super, top, bottom	Alignement vertical (cellules de tableau ou éléments inline-block uniquement)
line-height	18px, 120%, normal...	Hauteur de ligne
text-indent	25px	Alinéa
white-space	pre, nowrap, normal	Césure
word-wrap	break-word, normal	Césure forcée
text-shadow	5px 5px 2px blue (horizontale, verticale, fondu, couleur)	Ombre de texte

COULEURS

color	<i>nom, rgb(rouge,vert,bleu), rgba(rouge,vert,bleu,transparence), #CF1A20...</i>	Couleur du texte
background-color	<i>Identique à color</i>	Couleur de fond
background-image	<i>url('image.png')</i>	Image de fond
background-attachment	fixed, scroll	Fond fixe
background-repeat	repeat-x, repeat-y, no-repeat, repeat	Répétition du fond
background-position	(x y), top, center, bottom, left, right	Position du fond
background	-	Super propriété du fond. Combine : background-image, background-repeat, background-attachment, background-position
opacity	0.5	Transparence

TABLEAUX

border-collapse	collapse, separate	Fusion des bordures
empty-cells	hide, show	Affichage des cellules vides
caption-side	bottom, top	Position du titre du tableau

CURSEUR SOURIS

cursor	crosshair, default, help, move, pointer, progress, text, wait, e-resize, ne-resize, auto...	Curseur de souris
--------	---	-------------------

BOITES

width	150px, 80%...	Largeur
height	150px, 80%...	Hauteur
min-width	150px, 80%...	Largeur minimale
max-width	150px, 80%...	Largeur maximale
min-height	150px, 80%...	Hauteur minimale
max-height	150px, 80%...	Hauteur maximale
margin-top	23px	Marge en haut
margin-left	23px	Marge à gauche
margin-right	23px	Marge à droite
margin-bottom	23px	Marge en bas
margin	23px 5px 23px 5px (haut, droite, bas, gauche)	Super-propriété de marge. Combine : margin-top, margin-right, margin-bottom, margin-left.
padding-left	23px	Marge intérieure à gauche
padding-right	23px	Marge intérieure à droite
padding-bottom	23px	Marge intérieure en bas
padding-top	23px	Marge intérieure en haut
padding	23px 5px 23px 5px (haut, droite, bas, gauche)	Super-propriété de marge intérieure. Combine : padding-top, padding-right, padding-bottom, padding-left.
border-width	3px	Épaisseur de bordure
border-color	nom, rgb(rouge,vert,bleu), rgba(rouge,vert,bleu,transparence), #CF1A20...	Couleur de bordure
border-style	solid, dotted, dashed, double, groove, ridge, inset, outset	Type de bordure
border	3px solid black	Super-propriété de bordure. Combine border-width, border-color, border-style. Existe aussi en version border-top, border-right, border-bottom, border-left.
border-radius	5px	Bordure arrondie
box-shadow	6px 6px 0px black (horizontale, verticale, fondu, couleur)	Ombre de boîte

POSITION

display	block, inline, inline-block, table, table-cell, none...	Type d'élément (block, inline, inline-block, none...)
visibility	visible, hidden	Visibilité
clip	rect (0px, 60px, 30px, 0px) <i>rect (haut, droite, bas, gauche)</i>	Affichage d'une partie de l'élément
overflow	auto, scroll, visible, hidden	Comportement en cas de dépassement
float	left, right, none	Flottant
clear	left, right, both, none	Arrêt d'un flottant
position	relative, absolute, static	Positionnement
top	20px	Position par rapport au haut
bottom	20px	Position par rapport au bas
left	20px	Position par rapport à la gauche
right	20px	Position par rapport à la droite
z-index	10	Ordre d'affichage en cas de superposition. La plus grande valeur est affichée par-dessus les autres.

LISTES

list-style-type	disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none	Type de liste
list-style-position	inside, outside	Position en retrait
list-style-image	url('puce.png')	Puce personnalisée
list-style	-	Super-propriété de liste. Combine list-style-type, list-style-position, list-style-image.