

Initiation aux langages du Web :

CSS

Table des matières

1.Introduction	4
2.Notion d'héritage	4
3.Syntaxe	5
Exemple 2: Application d'un style à plusieurs balises	5
4.Inclusion de style	5
a)Dans l'entête	5
b)Dans un fichier externe	6
c)Dans une balise	6
5.Les sélecteurs	6
a)Le sélecteur « class »	6
b)Le sélecteur « id »	7
6.Mettre en forme un texte	7
a)Introduction	7
b)Polices génériques	7
Famille Serif	8
Famille Sans-serif	9
c)Taille des caractères	9
d)Choix des couleurs	11
e)Attributs utiles	11
7.Les marges	13
a)Aperçu	13
b)Spécification des tailles	13
c)Le modèle des boîtes	14
8.Les bordures	15
a)Styles de bordure	15
b)Encadrer du texte	16
9.Mise en page	16
a)Le principe	16
b)Code HTML	17
c)Code CSS	17
10.Image de fond	18
a)Quelques écueils à éviter	18
b)Code CSS	18
c)Autre paramètres	20
11.Quelques effets	20
a)Rollover	20
b)Les tooltips	21

c)Menu déroulant	23
Annexe	25
1.Index des exemples	25
2.Index des illustrations	26

1. Introduction

Le design d'un site évolue toujours et le problème, lorsqu'on n'utilise pas de feuilles de style, c'est qu'il faut reprendre toutes les pages une à une pour modifier une police de caractère ou une couleur de fond. En séparant style et contenu grâce au CSS, ce lourd handicap est résolu.

On peut déclarer les styles à différents endroits, et selon ces endroits ils seront plus ou moins prioritaires. On obtient donc une cascade de styles.

- « fichier », déclaration des styles dans une **feuille de style externe** : c'est de loin la meilleure chose à faire et la plus pratique à maintenir, mais c'est celle qui a le moins de poids ;
- « entête », déclaration des **styles en interne**, dans l'en-tête de la page : à ne faire qu'avec des styles particuliers à une page. Les styles déclarés auront plus de poids que ceux de la feuille de style externe et donc l'emporteront en cas de conflits ;
- Déclaration des **styles en attributs des éléments HTML** : à faire pour des mises en forme ponctuelles. Ces styles l'emporteront sur tous les autres.

2. Notion d'héritage

La notion d'héritage est présente par défaut en HTML / XHTML et nous l'avons déjà abordée pour la notion de taille.

Nous avons dit que la taille d'un élément spécifique, lorsqu'elle n'est pas précisée, correspond à celle de l'élément parent. Il en est de même pour le style.

Par exemple, lorsque l'on attache un style particulier à l'élément « body », c'est le style de toute la page qui est impacté. Cependant, il est possible d'attacher un style à un élément « fils » de « body » pour que son style soit redéfini.

3. Syntaxe

Pour chaque déclaration, la structure est toujours la même :

```
sélecteur {  
    propriété: valeur;  
}
```

Exemple 1: Syntaxe CSS

- le sélecteur, c'est la balise (eg. « body », « h1 », ...), l'identifiant (valeur de l'attribut « id ») ou la classe (valeur de l'attribut « class ») ;
- la propriété, c'est l'attribut qu'on veut appliquer (eg. « font », « margin », ...) et qui dépend de la balise ;
- la valeur précise les caractéristiques de la propriété.

Il est possible d'appliquer un style à plusieurs balises en séparant les noms des balises par des virgules :

```
h1, h2, h3 {  
    font-size: 100%;  
}
```

Exemple 2: Application d'un style à plusieurs balises

4. Inclusion de style

a) Dans l'entête

Pour inclure du CSS dans l'en-tête d'une page HTML, il faut utiliser la balise « style » à l'intérieur de la balise « head » :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">  
<head>  
    <title>Titre de la page</title>  
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />  
    <meta http-equiv="Content-Language" content="fr" />  
    <style>  
        * {padding:0px;margin:0px;}  
        body {background-color:#ffffff;}  
    </style>  
</head>  
<body>  
    <div>  
        ici, mon texte.  
    </div>  
</body>  
</html>
```

Exemple 3: Insertion de CSS dans l'en-tête

b) Dans un fichier externe

Pour inclure un fichier CSS externe, il faut utiliser la balise « link » à l'intérieur de la balise « head » :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
  <title>Titre de la page</title>
  <meta http-equiv="Content-Type" content="text/html;charset=iso-8859-1" />
  <meta http-equiv="Content-Language" content="fr" />
  <link href="style.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <div>
    ici, mon texte.
  </div>
</body>
</html>
```

Exemple 4: Inclusion d'un fichier CSS

c) Dans une balise

Pour définir un style CSS directement dans une balise, il faut utiliser l'attribut « style » :

```
<p style="font-color=#000000">Texte de couleur noir</p>
```

Exemple 5: Définition du style au niveau d'une balise

5. Les sélecteurs

Il existe deux types de sélecteurs en CSS : « class » et « id ».

a) Le sélecteur « class »

Ce sélecteur est utilisé lorsque l'on veut reproduire un style sur **plusieurs** éléments HTML. Dans une feuille de style, le nom du sélecteur « class » est toujours précédé d'un « . » :

```
.droite {
  text-align:right;
}
```

Exemple 6: Utilisation du sélecteur « class »

Pour appeler ce style au niveau d'un élément HTML, il suffit d'utiliser l'attribut « class » :

```
<p class="droite">
  bla bla
</p>
```

Exemple 7: Application d'un style avec l'attribut « class »

b) Le sélecteur « id »

Ce sélecteur est utilisé lorsque l'on veut attacher un style à **un seul** élément HTML. Dans une feuille de style, le nom du sélecteur « id » est toujours précédé d'un « # » :

```
#paragraphe {  
    text-align:right;  
}
```

Exemple 8: Utilisation du sélecteur « id »

Pour appeler ce style au niveau d'un élément HTML, il suffit d'utiliser l'attribut « id » :

```
<p id="paragraphe">  
    bla bla  
</p>
```

Exemple 9: Application d'un style avec l'attribut « id »

6. Mettre en forme un texte

a) Introduction

Le choix de la ou les polices de caractères doit se faire en connaissance de cause car elle est sujette aux contraintes suivantes :

- Toutes les polices de caractères n'existent pas sur tous les ordinateurs et si l'ordinateur de votre visiteur ne possède pas la police désirée, une autre police s'affichera modifiant votre mise en page ;
- Mac et PC ne possèdent pas toujours les mêmes noms de polices, ni la même résolution et vous devrez donc vérifier les équivalences entre ces deux plateformes ;
- Tous les navigateurs ne gèrent pas le rendu des polices de la même façon et il ne faut pas espérer le même rendu partout.

b) Polices génériques

Il existe 5 familles de polices de caractères dites "génériques" :

- Serif ;
- Sans-serif ;
- Monospace ;
- Cursive ;
- Fantasy ;

Ci-dessous un rendu de ces polices sur quelques navigateurs :

Firefox	IE Explorer	Opera	Safari
<ul style="list-style-type: none"> • Serif • Sans-serif • Monospace • Cursive • Fantasy 	<ul style="list-style-type: none"> • Serif • Sans-serif • Monospace • Cursive • <i>Fantasy</i> 	<ul style="list-style-type: none"> • Serif • Sans-serif • Monospace • Cursive • Fantasy 	<ul style="list-style-type: none"> • Serif • Sans-serif • Monospace • Cursive • Fantasy

On s'aperçoit que les polices serif et sans-serif sont celles qui subissent le moins de modifications et nous allons nous y intéresser.

Famille Serif

Les fontes de caractères suivantes, sont assez courantes :

- Times new roman (PC) ;
- Times (Mac) ;
- Georgia (Mac/PC) ;
- Palatino Linotype (PC) ;
- Palatino (Mac).

Les associations à respecter pour une homogénéité de rendu pourraient être :

- times new roman

```
body {
    font-family:"times new roman", times, serif;
}
```

Exemple 10: Police Serif Times new roman

- Georgia

```
body {
    font-family:georgia, serif;
}
```

Exemple 11: Police Serif Georgia

- Palatino Linotype

```
body {
    font-family:"palatino linotype", palatino, serif;
}
```

Exemple 12: Police Serif Palatino Linotype

Famille Sans-serif

Les fontes de caractères suivantes, sont assez courantes :

- Verdana (Mac/PC) ;
- Arial (Mac/PC) ;
- Trebuchet (PC) ;
- Helvetica (Mac) ;
- Tahoma (PC) ;
- Geneva (Mac).

Les associations à respecter pour une homogénéité de rendu pourraient être :

- Arial

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

Exemple 13: Police Sans-serif Arial

- Verdana

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

Exemple 14: Police Sans-serif Verdana

- Helvetica

```
body {  
    font-family:trebuchet, helvetica, sans-serif;  
}
```

Exemple 15: Police Sans-serif Helvetica

- Tahoma

```
body {  
    font-family:tahoma, geneva, sans-serif;  
}
```

Exemple 16: Police Sans-serif Tahoma

c) Taille des caractères

Il existe deux types d'unités pour régler la taille des caractères : fixes et proportionnelles.

Il est préférable de ne pas utiliser de tailles de caractères absolues comme les points (*pt*), les picas (*pc*) ou les centimètres (*cm*) car celles-ci ne sont pas redimensionnables à l'écran. Il faut également éviter d'utiliser les pixels (*px*) car Internet Explorer les considère comme une unité de taille de caractère fixe.

S'il est important de pouvoir redimensionner les caractères, c'est pour laisser la possibilité au visiteur de zoomer.

Il faut donc utiliser des unités relatives, telles que les *em* ou les pourcentages (%) car ces unités sont proportionnelles à la taille en pixels déclarée dans le navigateur qui par défaut est réglée à 16px.

La taille des caractères se déclare par l'attribut « font-size ».

```
body {
  font-family:tahoma, geneva, sans-serif;
  font-size:100%;
}
```

Exemple 17: Spécification d'une taille de police de caractères en CSS

d) Choix des couleurs

De manière générale, évitez les associations de couleurs pas assez contrastées comme, par exemple, le rouge et le vert ou encore le blanc et le gris.

Les couleurs se déclarent grâce à l'attribut « color » en utilisant des codes hexadécimaux ou RVB. Il est préférable d'éviter les noms de couleur (eg. red, green, ...) car elles n'ont pas la même interprétation selon les navigateurs.

```
body {
  font-family:tahoma, geneva, sans-serif;
  font-size:100%;
  color:#000000 ; /* code hexadecimal du noir */
}
```

Exemple 18: Spécification des couleurs en CSS

e) Attributs utiles

Ci-dessous un tableau regroupant certains attributs utilisés pour modifier la mise en forme :

Attribut	Valeur	Description
font-style	normal italic	italique
font-variant	normal small-caps	petites capitales
font-weight	normal bold	gras
text-align	left right center justify	aligné à gauche aligné à droite centré justifié
text-decoration	none underline overline line-through blink	rien souligné surligné rayé clignotant
text-transform	none capitalize uppercase lowercase	rien 1 ^{ère} lettre des mots en majuscules majuscules minuscules
line-height	normal number %	Normale taille en pixels taille en

		pourcents
--	--	-----------

Pour tester le rendu de police de caractères, vous pouvez vous aider du site suivant :

<http://www.typtester.org/>

7. Les marges

a) Aperçu

Pour affecter des marges aux différents éléments d'une page web, les propriétés CSS à utiliser sont « margin », pour les marges extérieures, et « padding » pour les marges intérieures.

Pour chaque élément HTML, on peut donc définir l'espacement qui le séparera des autres éléments (« margin ») et les espacements intérieurs dont il peut bénéficier (« padding »).

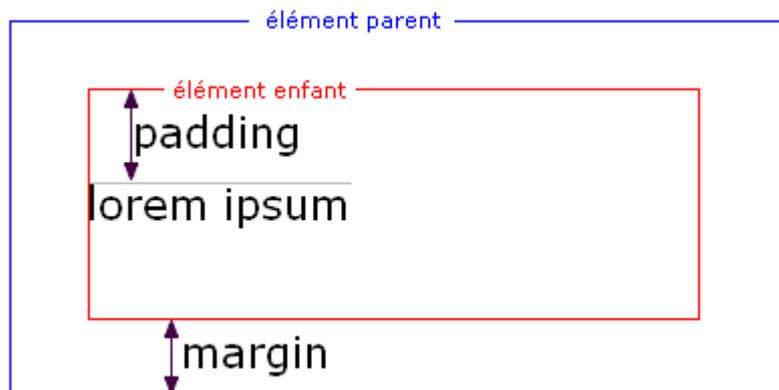


Illustration 1: Définition de la notion de marge en CSS

b) Spécification des tailles

On peut détailler les tailles des marges à l'aide des attributs « margin-top », « margin-right », « margin-bottom », « margin-left », ou synthétiser les quatre d'un seul coup (la première valeur étant celle du haut, puis on tourne dans le sens des aiguilles d'une montre).

On aura donc :

```
p {  
    margin: 2px 5px 2em 0;  
}
```

Exemple 19: Déclaration de marge implicite

Qui équivaut à :

```
p {  
    margin-top: 2px;  
    margin-right: 5px;  
    margin-bottom: 2em;  
    margin-left: 0;  
}
```

Exemple 20: Déclaration de marge explicite

Si on ne met que deux valeurs, la première s'appliquera à « margin-top » et « margin-bottom », la seconde à « margin-right » et « margin-left ».

On aura donc :

```
p {  
    margin: 2px 5px;  
}
```

Exemple 21: Déclaration de marge implicite à deux valeurs

Qui équivaut à :

```
p {  
    margin-top: 2px;  
    margin-right: 5px;  
    margin-bottom: 2px;  
    margin-left: 5px;  
}
```

Exemple 22: Déclaration de marge explicite à deux valeurs

c) Le modèle des boîtes

D'après le « *box model* », lorsqu'on attribue une taille à un élément de type block à l'aide des attributs « width » ou « height », les marges viennent s'ajouter à cette taille.

Si l'on prend le code CSS suivant :

```
p {  
    width: 200px;  
    padding: 40px;  
    margin: auto;  
}
```

Exemple 23: Déclaration du style pour un paragraphe

Le paragraphe aura une largeur totale de $200 + 2 \times 40 = 280$ pixels.

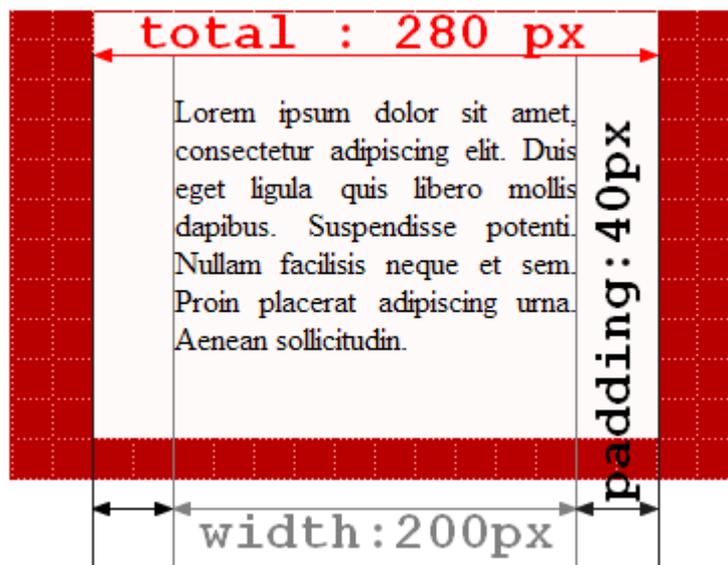


Illustration 2: Modèle des boîtes

Vous aurez remarqué l'attribut « margin » avec sa valeur « auto » qui permet, dans le cas d'un bloc muni d'une largeur, de centrer horizontalement un élément.

8. Les bordures

Les bordures permettent, par exemple, de délimiter une zone, mettre en relief une information ou encore entourer les cellules d'un tableau.

La propriété permettant d'obtenir une bordure en CSS est tout simplement « border ». On peut associer une épaisseur de bordure avec « border-width », un style avec « border-style » et une couleur avec « border-color ».

Comme pour les marges, il est possible de spécifier de manière implicite les paramètres :

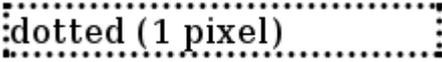
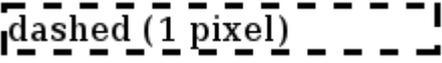
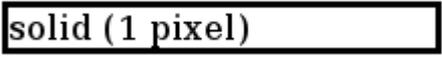
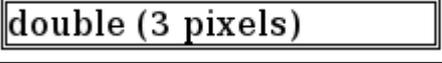
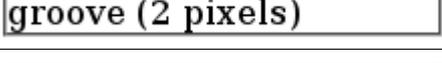
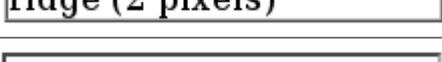
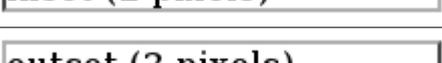
```
p {
    border: width style color;
}
```

Exemple 24: Spécification implicite des paramètres de bordure

L'épaisseur des bordures (« border-width ») est généralement exprimée en pixel (px).

a) Styles de bordure

Le tableau ci-dessous résume les huit styles de bordures différentes avec l'épaisseur minimum pour un rendu optimal :

Style	Rendu
dotted (pointillé)	
dashed (tirets)	
solid (solide)	
double (double)	
groove (rainurée)	
ridge (relief)	
inset (relief intérieur)	
outset (relief extérieur)	

Pour obtenir une bordure en pointillés d'une épaisseur de 1 pixel et de couleur noire on

peut utiliser le CSS suivant :

```
p {  
    border-width: 1px;  
    border-style: dotted;  
    border-color: black;  
}
```

Exemple 25: Déclaration d'une bordure pointillée noire explicite

Ou plus simplement :

```
p {  
    border: 1px dotted black;  
}
```

Exemple 26: Déclaration d'une bordure pointillée noire implicite

b) Encadrer du texte

Pour encadrer tout ou partie d'un texte, on peut utiliser la balise « span » :

```
<p>Exemple d'encadrement de texte avec un <span>span</span></p>
```

Exemple 27: Encadrement avec un span (HTML)

Il n'y a plus qu'à définir le style de l'élément span dans une feuille de style :

```
span {  
    border: 1px solid red;  
    padding: 3px;  
}
```

Exemple 28: Encadrement avec un span (CSS)

9. Mise en page

La majorité des sites sont constitués des éléments suivants :

- un **header** qui affiche la bannière et caractérise l'identité du site ;
- un **footer** qui affiche certains liens ;
- un **menu** qui permet la navigation ;
- une **zone** centrale qui affiche le contenu.

Même s'il est possible de procéder à une telle mise en page grâce aux tableaux, ils ne sont pas prévus pour cela.

a) Le principe

Nous allons « découper » le site en plusieurs parties grâce à des éléments « div » que nous allons identifier de manière unique grâce à l'attribut « id ». Une fois identifiés, il ne reste plus qu'à attacher un style aux différentes « div » pour procéder à la mise en page.

b) Code HTML

```
<div id="header">
  Header
</div>
<div id="main">
  <div id="menu">
    Menu
  </div>
  <div id="contenu">
    Contenu
  </div>
</div>
<div id="footer">
  Footer
</div>
```

Exemple 29: Site segmenté par des éléments « div »

Le site exposé possède les quatre grandes sections décrites précédemment. Cependant, on remarquera que la section « menu » et « contenu » sont incluses dans une section « main ».

Cela va nous permettre de centrer ces deux sections et surtout de les rendre **redimensionnables**.

c) Code CSS

```
/* permet de fusionner les marges */
#header, #menu, #contenu, #footer {
padding: 1px 0;
}
#header {
text-align:center;
}
#main {
/* permet d'émuler un tableau */
display:table;
/* la valeur auto permet de centrer automatiquement le contenu */
margin:auto;
}
#menu {
/* permet d'émuler une cellule d'un tableau */
display:table-cell;
width:180px;
}
#contenu {
/* permet d'émuler une cellule d'un tableau */
display:table-cell;
}
#footer {
text-align:center;
}
```

Exemple 30: CSS permettant une mise en page de type tableau

La « div » « main » est émulée en tableau grâce à « display:table » et les « div » « menu »

et « contenu » deviennent des cellules de ce tableau grâce à « display:table-cell ».

Le contenu est aligné par défaut en haut, mais on peut modifier l'alignement en jouant sur le paramètre « vertical-align » qui prend les valeurs « middle » (centrer) ou « bottom » (en bas).

10. Image de fond

L'image de fond permet de casser avec les fonds unicolores traditionnels et donne, si elle est bien choisie, une véritable identité au site.

a) Quelques écueils à éviter

- L'image de fond doit être **légère** et ne pas dépasser les 20 Ko sous peine de ralentir le chargement du site ;
- Elle ne doit pas être trop contrastée, sous peine d'entraîner des problèmes de lisibilité ;
- La couleur de fond doit être différente de celle de la police de caractère pour ne pas pénaliser la lisibilité du contenu pendant le chargement de l'image de fond ;
- L'image de fond ne doit pas servir à afficher du contenu. Les images affichant du contenu, sont insérées par le biais de la balise « img » (sans oublier l'attribut « alt »).

b) Code CSS

L'image de fond s'attache généralement à la balise « body » mais il est possible de l'attacher à n'importe quel élément de type bloc. Dans l'exemple suivant, l'image est attachée au corps de la page (« body ») :

```
body {  
    color:black;  
    background-color:white;  
    background-image:url(images/fond.png);  
}
```

Exemple 31: Attachement d'une image de fond

Les attributs :

- « color » renseigne la couleur de la police de caractère ;
- « background-color » renseigne la couleur de fond de page ;
- « background-image:url() » renseigne l'URL de l'image choisie pour le fond de

page.

c) *Autre paramètres*

Ci-dessous un tableau récapitulant les principaux attributs et leurs valeurs :

Attribut	Valeur	Description
background-attachment	fixed	Fixe l'image de fond au lieu de la faire défiler avec le texte
background-position	[right center left] [top center bottom] pourcentage	La première valeur fixe la verticalité et le second l'horizontalité : « right top » ou « 100% 0% »
background-repeat	no-repeat repeat-x repeat-y repeat	Permet de répéter l'image de fond selon l'axe horizontal, vertical, les deux ou aucun.

Il est également possible de cumuler toutes ces informations de manière implicite :

```
body {  
    background: white url(fond.png) no-repeat right bottom;  
}
```

11. Quelques effets

a) *Rollover*

L'effet de *rollover* est très utilisé pour rendre un site plus ergonomique. Par exemple, lorsque la souris survole un lien, celui-ci peut changer de couleur pour montrer à l'utilisateur que la souris est bien positionnée.

De tels effets sont déclenchés grâce à l'événement « hover ».

L'exemple ci-dessous montre l'événement « hover » attaché à une ancre :

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

Exemple 32: Événement « hover » sur une ancre

Lorsque la souris passe sur une ancre, le texte devient rouge et le soulignement est supprimé. On peut aller plus loin en changeant la couleur de fond (« background-color ») ou encore ajouter une image (« background-image »).

L'élément « a » possède plusieurs événements :

- « link » lien non visité ;
- « visited » lien visité.
- « hover » lien survolé par la souris ;
- « active » lien actif ;

L'ordre dans lequel sont énoncés ces événements doit être le même que celui dans lequel ils apparaissent dans la feuille de style.

b) Les tooltips

Le but recherché est d'avoir une zone de la page qui réagit au survol de la souris par un tooltip.

Il y a donc deux zones à identifier dans la page :

- La zone qui déclenche le tooltip ;
- La zone qui contient le texte à afficher.

Pour la première nous pouvons avoir le choix (« p », « img », ...) alors que pour la deuxième il est préférable d'utiliser l'élément « span » (étudié pour le texte). Ci-dessous un exemple avec un paragraphe et une image :

```
<span class="bulle">  
    
  <span>Voici un popup sur une image</span>  
</span>  
<br/>  
<p class="bulle">  
  Bla bla  
  <span>Voici un popup sur un paragraphe</span >  
</p>
```

Exemple 33: Utilisation des pop-ups (HTML)

On peut remarquer que :

- L'attribut « class » des zones qui déclenche le tooltip ont tous la valeur « bulle » ;
- L'image est encadrée d'un élément « span » ;

Le fait de positionner l'attribut class des éléments qui déclenche le tooltip permet de les identifier et ainsi on peut masquer le texte du tooltip.

Pour ce qui est de l'image, il ne faut pas oublier que c'est un élément de type en ligne et donc ne peut contenir un autre élément. Le fait d'utiliser un élément « span » permet de palier à ce problème.

```

.bulle {
    cursor: pointer;
}
.bulle span {
    display: none;
}
.bulle:hover span {
    padding: 3px;
    border: 1px black solid;
    display: block;
    position: absolute;
    background-color: #E4E4E4;
}

```

Exemple 34: Utilisation des pop-ups (CSS)

Quelques explications sur le code CSS :

- Le premier bloc « .bulle » a pour objectif de remplacer le pointeur de la souris par une main.
- Le deuxième bloc « .bulle span » permet de masquer tous les éléments « span » (le texte du popup) contenus dans les éléments de classe « bulle » indépendamment du type de l'élément ;
- Le troisième et dernier bloc déclenche l'affichage du pop-up.

c) *Menu déroulant*

Cet effet ne sera supporté que sur des navigateurs récents et il faudra plutôt privilégier Javascript pour que cela fonctionne sur tous les navigateurs. L'avantage majeur de CSS est sa légèreté et c'est pourquoi le sujet mérite d'être abordé.

Le contexte est simple, on a deux listes imbriquées, une qui affiche les entêtes des menus et une autre qui affiche les rubriques :

```

<div id="menu">
<ul>
<li><a href="#">Menu 1</a></li>
<li><a href="#">Menu 2</a>
<ul>
<li><a href="#">sous-menu 1</a></li>
<li><a href="#">sous-menu 2</a></li>
</ul>
</li>
</ul>
</div>

```

Exemple 35: Menu déroulant (HTML)

Ci-dessous le code CSS permettant de gérer les animations :

```
#menu ul {
    margin:0;
    padding:0;
    list-style-type:none;
    text-align:center;
}
#menu li {
    float:left;
    margin:auto;
    padding:2px;
    background-color:#E4E4E4;
}
#menu ul li ul {
    display:none;
}
#menu ul li:hover ul {
    display:block;
}
#menu li:hover ul li {
    float:none;
}
#menu li ul {
    position:absolute;
}
```

Exemple 36: Code CSS gérant les animations du menu

Le premier bloc sert à modifier le style des listes pour enlever les puces (« list-style-type:none »), centrer le texte (« text-align:center ») et éliminer les espacements entre éléments.

Le deuxième bloc sert à positionner le menu à gauche (« float:left ») ainsi qu'à centrer la deuxième liste en dessous de la première (« margin:auto »).

Le troisième bloc sert à masquer la deuxième liste.

Le quatrième bloc gère l'effet d'apparition de la liste imbriquée.

Le cinquième bloc permet de faire apparaître les éléments de la liste imbriquée les uns sous les autres (et non les uns à côté des autres).

Le dernier bloc permet d'aligner les sous-menus sur l'entête de la section pour ne pas la décentrer.

Annexe

1. Index des exemples

Index des exemples

Exemple 1: Syntaxe CSS.....	5
Exemple 2: Application d'un style à plusieurs balises.....	5
Exemple 3: Insertion de CSS dans l'en-tête.....	5
Exemple 4: Inclusion d'un fichier CSS.....	6
Exemple 5: Définition du style au niveau d'une balise.....	6
Exemple 6: Utilisation du sélecteur « class ».....	6
Exemple 7: Application d'un style avec l'attribut « class ».....	6
Exemple 8: Utilisation du sélecteur « id ».....	7
Exemple 9: Application d'un style avec l'attribut « id ».....	7
Exemple 10: Police Serif Times new roman.....	8
Exemple 11: Police Serif Georgia.....	8
Exemple 12: Police Serif Palatino Linotype.....	8
Exemple 13: Police Sans-serif Arial.....	9
Exemple 14: Police Sans-serif Verdana.....	9
Exemple 15: Police Sans-serif Helvetica.....	9
Exemple 16: Police Sans-serif Tahoma.....	9
Exemple 17: Spécification d'une taille de police de caractères en CSS.....	11
Exemple 18: Spécification des couleurs en CSS.....	11
Exemple 19: Déclaration de marge implicite.....	13
Exemple 20: Déclaration de marge explicite.....	13
Exemple 21: Déclaration de marge implicite à deux valeurs.....	14
Exemple 22: Déclaration de marge explicite à deux valeurs.....	14
Exemple 23: Déclaration du style pour un paragraphe.....	14
Exemple 24: Spécification implicite des paramètres de bordure.....	15
Exemple 25: Déclaration d'une bordure pointillée noire explicite.....	16
Exemple 26: Déclaration d'une bordure pointillée noire implicite.....	16
Exemple 27: Encadrement avec un span (HTML).....	16
Exemple 28: Encadrement avec un span (CSS).....	16
Exemple 29: Site segmenté par des éléments « div ».....	17
Exemple 30: CSS permettant une mise en page de type tableau.....	17
Exemple 31: Attachement d'une image de fond.....	18
Exemple 32: Événement « hover » sur une ancre.....	20
Exemple 33: Utilisation des pop-ups (HTML).....	21
Exemple 34: Utilisation des pop-ups (CSS).....	23

Exemple 35: Menu déroulant (HTML).....	23
Exemple 36: Code CSS gérant les animations du menu.....	24

2. Index des illustrations

Index des exemples

Illustration 1:Définition de la notion de marge en CSS.....	13
Illustration 2:Modèle des boîtes.....	14