

# Cascading Style Sheets

Code: css-intro

## Originaux

[url: http://tecfa.unige.ch/guides/tie/html/css-into/css-intro.html](http://tecfa.unige.ch/guides/tie/html/css-into/css-intro.html)

[url: http://tecfa.unige.ch/guides/tie/pdf/files/css-intro.pdf](http://tecfa.unige.ch/guides/tie/pdf/files/css-intro.pdf)

## Auteurs et version

- [Daniel K. Schneider](#) - [David Ott](#) - [Vivian Synteta](#) - [Natascha Michel](#)
- Version: 1.1 (modifié le 18/9/07 par DKS)

## Prérequis

*Module technique précédent:* [html-intro](#)

*Module technique précédent:* [xhtml-intro](#)

*Module technique précédent:* [xml-tech](#) (pour les chapitres sur XML seulement)

## Autres modules

*Module technique suppl.:* [xml-xslt](#)

## **Abstract**

Petite introduction à CSS

## **Objectifs**

- Savoir faire de simples style sheets HTML avec CSS
- Faire des pages emboîtées sans tables
- XML avec CSS
- (Pas de DHTML pour le moment, ce se fera dans un autre module je pense)

## **A faire**

- Améliorer les explications, ce document est difficile à comprendre en auto-formation .....

# 1. Table des matières détaillée

1. Table des matières détaillée	3
2. Introduction aux "Cascading Style Sheets"	5
2.1 Motivation	5
2.2 Principe de base	6
2.3 Ressources	7
2.4 Conseils HTML	7
3. Association d'une feuille de style à une page HTML	8
3.1 Styles "inline"	8
3.2 Feuille de style imbriquée dans une page	9
3.3 Feuille de style externe associé à des pages HTML	10
3.4 Multiples feuilles de style	11
3.5 Importation	12
3.6 Feuilles de style client-side	12
4. La notion de sélecteurs	13
4.1 Simple sélecteurs pour les balises	13
4.2 Class et attributs ID	14
A.Les attributs et sélecteurs "class" 14	
B.Attribut ID 16	
C."wildcard" 16	
4.3 Les balises HTML <div> et <span>	17
A.<div> 17	
B.Définition "inline" d'un style 17	
C.Définition externe d'un style 18	
4.4 Sélecteurs contextuels	19
4.5 Pseudo sélecteurs (élément et classes)	20
A.Pseudo-éléments 20	
B.Pseudo-classes 20	
5. Les déclarations CSS	21
5.1 Types d'éléments et défauts HTML	22

<b>6. Attributs du texte</b>	<b>23</b>
<b>6.1 Attributs pour la gestion des propriétés d'écriture</b>	<b>23</b>
<b>6.2 Alignement du texte</b>	<b>24</b>
<b>7. Les boites CSS et leur positionnement</b>	<b>25</b>
<b>7.1 Anatomie des boites</b>	<b>25</b>
<b>7.2 Les bords, les cadres et le couleur</b>	<b>26</b>
<b>7.3 Le positionnement</b>	<b>29</b>
<b>8. XML avec CSS</b>	<b>31</b>
<b>8.1 Association d'une feuille de style</b>	<b>31</b>
<b>8.2 Selecteurs CSS2 pour XML et HTML</b>	<b>32</b>
<b>8.3 Premières opérations à faire</b>	<b>34</b>

## 2. Introduction aux "Cascading Style Sheets"

### 2.1 Motivation

#### Avantages

- Mises en page HTML et XML sophistiquées
- DHTML (changement de positionnement d'éléments sur x,y et z par exemple)
- Séparation de contenu et de style:
  - permet de servir une page à plusieurs "sauces" (HTML riche, HTML pauvre mais efficace, HTML vocal, ....), donc variantes pour imprimer, le PDA, etc.
  - permet de gérer centralement le "look" de pleines de pages, donc diminue le coût de maintenance
  - rends une page plus "lisible" et plus rapide à télécharger

#### Désavantages

- Dans CSS1 / CSS2: peu de capacités pour transformer un texte (voir XSLT)
- Implémentations CSS1: bonnes dans Mozilla/Firefox et potables dans IE 6
- Implémentations CSS2: toujours lacunaires.
- Il existe des tables de compatibilités

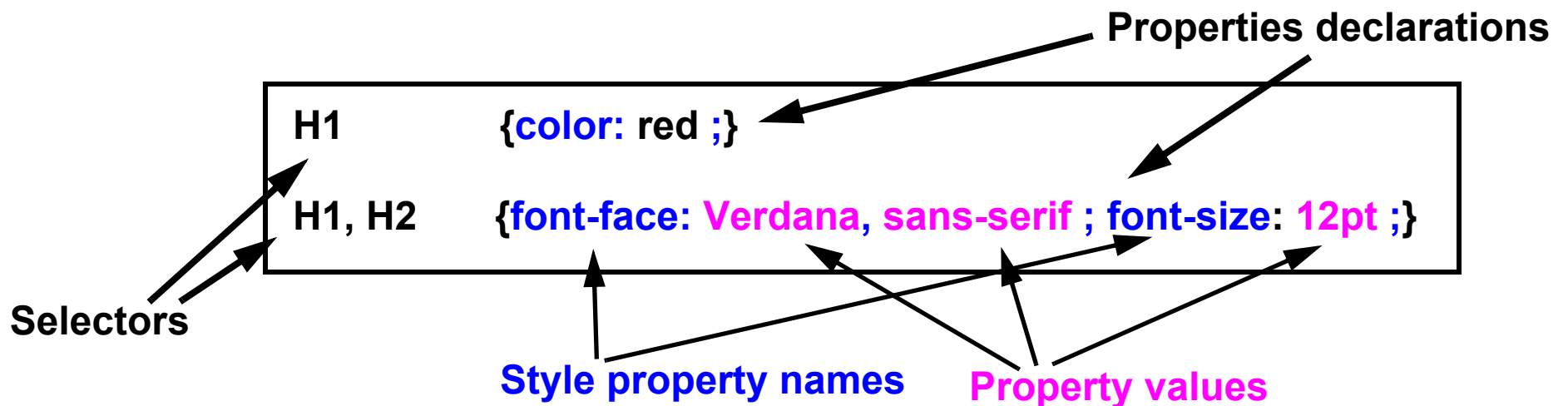
*url:* (voir <http://tecfa.unige.ch/guides/css/pointers.html>)

## 2.2 Principe de base

- Feuille de style = jeux de règles qui précise l'affichage d'un élément HTML ou XML
- Chaque règle est composée:
  - d'un **sélecteur** (qui indique à quel type d'élément la règle s'applique)
  - une **déclaration** (qui comprend une ou plusieurs instructions de mise en page)

### Exemples (de sensibilisation)

```
H1 { color: red ; }
P  { font-face: Verdana, sans-serif ; font-size: 12pt }
H1, H2, H3 { color : blue }
H1.ChapterTOC, H2.PeriodeTOC, H2.ExerciceTOC, H2.SectionTOC {
  display: block;text-indent: 30pt;
  text-align: left; font-size: 14.000000pt;
  font-weight: Bold; font-family: "Times";
}
```



## 2.3 Ressources

[url: http://tecfa.unige.ch/guides/css/pointers.html](http://tecfa.unige.ch/guides/css/pointers.html) (voir la section links!!)

[url: http://www.w3.org/Style/CSS/](http://www.w3.org/Style/CSS/) (CSS page c/o W3C)

## 2.4 Conseils HTML

- Mettez des balises fermantes (<p> .....</p>, <li> .... </li>, etc)
- Testez avec Mozilla/Firefox, ajustez ensuite pour IE Explorer (et autres navigateurs déficients).

### **Pour être compatible:**

- Pour gérer les anciens browsers (NS 4 et IE 5) comprenant mal CSS il faut écrire des scripts de filtrage (disponibles sur le "web master's sites").
- Pour gérer les Netscape 3 etc. pas de problème, ils ignorent les CSS

## 3. Association d'une feuille de style à une page HTML

Il existe plusieurs méthodes

### 3.1 Styles "inline"

- On peut définir un attribut style pour chaque élément HTML

```
<p style="color: red; font: 12pt times; margin-left: 2em">
```

On peut définir un style pour chaque balise.

Ce paragraphe contient son propre style, mais ce n'est pas très efficace ...

```
</p>
```

**[url: http://tecfa.unige.ch/guides/css/ex/simple-css2.html](http://tecfa.unige.ch/guides/css/ex/simple-css2.html)**

- Il s'agit ici d'une technique qu'il faut mieux éviter, car elle ne respecte pas très bien le principe de la séparation de contenu et de présentation
- Voir aussi `<span>` qui permet de définir un style à l'intérieur d'un paragraphe.

```
<p>
```

On peut définir `<span style="color: red;">` une phrase toute rouge `</span>`. Et revenir à la normale plus tard ...

```
</p>
```



## 3.2 Feuille de style imbriquée dans une page

- Déclaration d'une feuille de style avec la balise <style>

### Exemple 3-1: Simples feuilles CSS imbriquées

[url: http://tecfa.unige.ch/guides/css/ex/simple-css.html](http://tecfa.unige.ch/guides/css/ex/simple-css.html)

[url: http://tecfa.unige.ch/guides/css/ex/simple-css2.html](http://tecfa.unige.ch/guides/css/ex/simple-css2.html) (exemple plus compliqué)

### Utilisation:

- La déclaration doit se faire dans le <head> (sinon il peut y avoir des problèmes, par ex. la couleur background du body qui ne s'affiche pas)
- Pour rester compatible avec les anciens navigateurs, il faut commenter les règles CSS comme dans l'exemple ci-dessous !

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.2//EN">
<html> <head>
  <title>Simple CSS démo</title>
  <STYLE type="text/css">
    <!--
    body {background: white; font-family: Helvetica, Arial, sans-serif;}
    H2, H3{font-family: Helvetica, Arial, sans-serif;}
    P.intro {color: blue; margin-left: 4em; margin-right: 2em;}
    .default {margin-left: 2em;}
    -->
  </STYLE>
</head>
```

### 3.3 Feuille de style externe associé à des pages HTML

- Solution efficace pour gérer le look des plusieurs (ou pleins!) de pages
- La balise <link> doit se trouver dans le <head>
- Si vous utilisez des caractères non standards dans le texte ("é", "ö", etc.) il faudrait déclarer l'attribut CHARSET dans le HTML (à vérifier)
- Pensez à créer un répertoire central pour vos styles ! (A tecfa: /web/styles)
- Vous pouvez regarder le source d'une page HTML et ensuite manuellement entrer l'URL pour le CSS et le regarder (s'il en existe un)

#### Exemple 3-2: Simple feuille CSS externe

[url: http://tecfa.unige.ch/guides/css/ex/simple-css3.html](http://tecfa.unige.ch/guides/css/ex/simple-css3.html)

```
<html>
  <head>
    <link rel="stylesheet" href="simple-css3.css" charset="ISO-8859-1" type="text/css"/>

    <title>Simple CSS démo III</title>
  </head>
  <body>
```

**ATTENTION:** Dans le fichier - feuille de style vous mettez *juste* les déclarations CSS, rien d'autre: pas d'entêtes, rien d'autre !!

## 3.4 Multiples feuilles de style

- On peut laisser l'utilisateur choisir !
  - dans Mozilla en tout cas (Menu View->Use Style)
  - on peut aussi écrire un programme JavaScript qui le fait
- [url: http://www.alistapart.com/stories/alternate/test.html](http://www.alistapart.com/stories/alternate/test.html)
- Pensez à faire une version spéciale pour l'impression

Exemple de la home page de DKS:

```
<style type="text/css" media="all">@import "/tecfa-people/dks-normal.css";</style>
```

```
<link rel="alternate stylesheet" type="text/css" media="screen" title="Friendly  
fonts" href="/tecfa-people/dks-friendly.css" />
```

```
<link rel="alternate stylesheet" type="text/css" media="screen" title="bigtype" href="/  
tecfa-people/dks-big.css" />
```

```
<link rel="alternate stylesheet" type="text/css" media="screen" title="Print with URLs"  
href="/tecfa-people/dks-print-url.css" />
```

```
<link rel="alternate stylesheet" type="text/css" media="screen" title="Print" href="/  
tecfa-people/dks-print.css" />
```

```
<link rel="stylesheet" type="text/css" media="print" href="/tecfa-people/dks-  
print.css" />
```

## 3.5 Importation

- Il s'agit d'une alternative au "linking" ci-dessus

[url: http://tecfa.unige.ch/guides/css/ex/simple-css4.html](http://tecfa.unige.ch/guides/css/ex/simple-css4.html)

```
<style>
  <!--
    @import ("simple-css3.css") ;
    p.default {margin-right: 3em; margin-left: 3em}
  -->
</style>
```

### Attention:

- Marche mal avec certains Explorer 4
- Ne marche pas avec Netscape (4.x) ! On peut utiliser cet fait pour filter ces navigateurs ....

## 3.6 Feuilles de style client-side

- Marche uniquement dans Mozilla et c'est très cool:
- On peut censurer tout ce qu'on aime pas, adapter la page à la vue etc. :)

## 4. La notion de sélecteurs

- Un sélecteur est un "nom" qui indique à quels éléments d'une page HTML on desire appliquer une règle.
- Chaque règle CSS commence nécessairement par un sélecteur.

### **Rappel de la syntaxe pour une règle CSS:**

```
Syntaxe: selecteur(s) { propriété: valeur ; propriété: valeur1,  
valeur2, valeur3 ; ..... }
```

Il existe plusieurs méthodes pour identifier les éléments, par exemple:

- on indique le nom d'une balise (X)HTML
- on indique la classe d'une balise (X)HTML
- on indique l'identificateur d'une balise XHTML
- on indique le nom d'une balise plus sa classe, etc.

### 4.1 Simple sélecteurs pour les balises

- On indique le nom d'une balise, ou encore une liste séparés par des virgules.

```
H1 {color: green}
```

```
H2 {color: green}
```

est équivalent à:

```
H1, H2 {color: green}
```

- Dans ce cas on utilise CSS pour changer l'affichage par défaut des différentes balises
- Typiquement, on modifie la couleur de certains éléments, la taille des fontes, les marges, etc.

## 4.2 Class et attributs ID

### A. Les attributs et sélecteurs "class"

- Mécanisme puissant pour:
  - définir de classes de contenu (indépendamment des balises)
  - pour différencier plusieurs variantes d'une même balise
- Principe: On associe une valeur à l'attribut "class"
- On peut aussi définir le style des enfants de la balise qui a l'attribut "class" ; mais il faut indiquer le chemin complet de l'arbre (*dans l'ex., default ul li*).
- il est possible d'avoir plusieurs classes par éléments (*dans l'ex., le 2ème <p>*).
- Lorsque deux styles définissent la même propriété, c'est celui qui est le plus L de l'élément qui "gagne" (*dans l'ex., font-size*)

## Exemple 4-1: Classes dans une simple feuille de style

[url: http://tecfa.unige.ch/guides/css/ex/simple-css2.html](http://tecfa.unige.ch/guides/css/ex/simple-css2.html)

```
<STYLE type="text/css">
  P.intro {color: blue; margin-left: 4em; margin-right: 2em;}
  .default ul li {font-size: 0.8em;}
  .default {margin-left: 2em; font-size:1em;}
  .bleu {color:red;}
</STYLE>

.....
<p class="intro">
  Après des années de bricolage infâme ....
</p>
<p class="default bleu">
  bla bla
</p>
<div class="default">
<ul>
  <li>item 1</li>
  <li>item 2</li>
</ul>
<pre>
  notre classe default marche partout, même pour un para préformaté
</pre>
</div>
```

Le style `.default` s'appliquera à tous les éléments dont la classe est "default" alors que le style `.intro` seulement aux balises `<p>`.

## B. Attribut ID

- Parfois plus économique d'utiliser l'attribut ID (p.ex. lorsqu'on a un script qui implique des id)
- on peut aussi combiner les deux (définir un style pour la classe et l'id d'une même balise)
- Le selecteur s'écrit avec un # devant
- Si on veut définir les éléments enfants d'un éléments avec l'attribut id ayant une classe définie, on peut le faire sans passer par les éléments intermédiaire
  - (pour l'ex. ci-dessus, sans ul)

## C. "wildcard"

- "Wildcard" \* (s'applique à tous le éléments) :  

```
* { font-size: 12pt; }
```
- pratique pour définir une police par défaut pour tous les éléments sans la définir à chaque fois (y compris `<pre>` et `<button>`).



## 4.3 Les balises HTML <div> et <span>

- ont été créés spécialement pour un usage avec les feuilles de style

### A. <div>

- peut contenir toutes les autres balises HTML et donc servir à mettre en forme toute une section d'une texte (mais attention aux priorités! <div> est un bloc alors que <span> est un élément in-line, cf 5.1 "Types d'éléments et défauts HTML" [22] Il sera pas possible de définir une largeur de 100% -toute la page- pour un span)
- <div> avec l'attribut class peut servir à faire des "custom tags" (voir plus loin)
- <span> sert à changer une séquence de caractères à l'intérieur d'une balise

### B. Définition "inline" d'un style

#### Exemple 4-2: Changer la couleur d'un partie du texte

[url: http://tecfa.unige.ch/guides/css/ex/simple-div-span.html](http://tecfa.unige.ch/guides/css/ex/simple-div-span.html)

```
<div style="color: blue">
  <h1>L'influence d'un div tag qui aime le bleu</h1>
  <P>
    bla bla bla
  </P>
</div>
```

## C. Définition externe d'un style

### Exemple 4-3: Faire des backgrounds avec une définition externe

[url: http://tecfa.unige.ch/guides/css/ex/simple-div2.html](http://tecfa.unige.ch/guides/css/ex/simple-div2.html)

```
<style>
  div.important {
    background: rgb(204,204,255);
    padding: 0.5em; border: none;
  }
</style>
.....
<body>
  <div class="important">
    <h1>Une section importante</h1>

    <h2>Un sous-titre</h2>

    <P>
      Tout le div a son joli background à lui.
      bla bla bla
    </P>
  </div>
```

## 4.4 Sélecteurs contextuels

### Exemple 4-4: Simple utilisation d'un sélecteur contextuel

[url: http://tecfa.unige.ch/guides/css/ex/simple-css5.html](http://tecfa.unige.ch/guides/css/ex/simple-css5.html)

```
<STYLE type="text/css">
  <!--
  P strong {color: red;}
  -->
</STYLE>
```

.....

```
<P>
```

Ceci est un paragraph avec un **<strong>** strong modifié **</strong>**.

En d'autres terme cela permet de redéfinir de façon simple le rendering de certaines balises "logiques" comme `&lt;em&gt;` ou `&lt;strong&gt;`.

```
</P>
```

## 4.5 Pseudo sélecteurs (élément et classes)

- Un pseudo sélecteur identifie un élément par des critères autres que balise et classe
- CSS2 en définit pleins, ci-dessous les éléments de CSS1.
- Attention: certains navigateurs implémentent mal ces fonctionnalités CSS1.

### A. Pseudo-éléments

- permettent d'identifier des éléments non-balisés (comme la première lettre ou la première ligne)
- Marche avec les navigateurs IE 5.5 (à confirmer) et Mozilla

### Exemple 4-5: Pseudo-sélecteurs

[url: http://tecfa.unige.ch/guides/css/ex/pseudo-selectors.html](http://tecfa.unige.ch/guides/css/ex/pseudo-selectors.html)

```
<style>
  P:first-letter { font-size: 500%; color: green }
  P:first-line { color: green }
</style>
```

### B. Pseudo-classes

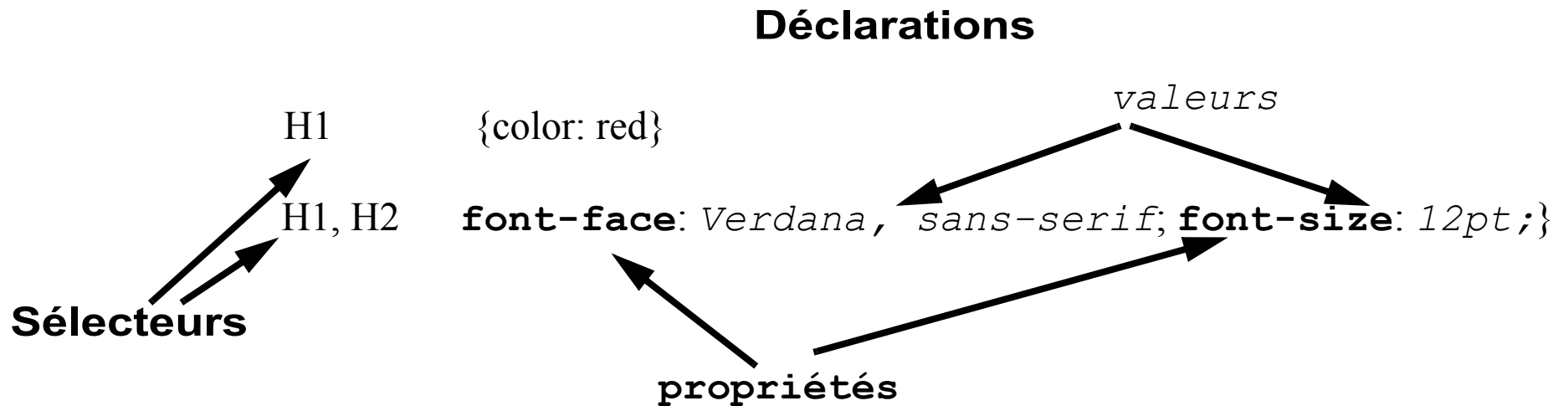
- Pour rendre plus difficile la lecture des liens :)

```
A:link { color : white }
A:visited { color : yellow }
A:active {color : red }
```

## 5. Les déclarations CSS

### Rappel de la syntaxe pour une règle CSS:

selecteur(s) { propriété: valeur ; propriété: valeur1, valeur2, valeur3 ; ..... }



### Attention:

- Les valeurs des propriétés sont séparées par des ","
- Les pairs "propriétés: valeurs" sont séparées par des ";"

[ reste = à développer, voir d'autres documentations en attendant]

- Types
- Boîtes
- Positionnement.....

## 5.1 Types d'éléments et défauts HTML

Avant d'utiliser un style, voir quel est l'élément html qui s'en rapproche le plus. Par exemple, pour faire des titres, se baser sur `<h1>`, `<h2>`, `<h3>`... plutôt que sur `<p>` ou `<div>`.

En typographie on distingue entre 2-3 types d'éléments de base:

1. les blocs (blocks), c.a.d. des éléments qui commencent un nouveau paragraphe. Exemples HTML: `<p>`, `<h2>`, `<div>`
  2. Les listes et leurs éléments sont des blocs spéciaux: `<li>`
  3. les "in-line", c.a.d. s'insérant dans un paragraphe. Exemples HTML: `<b>`, `<strong>`, `<span>`  
`display: block;`  
`display: inline;`  
`display: list-item;`
- Pour chaque élément "bloc" il existe un jeu de règles pour définir les marges, la position, les couleurs du texte et de l'arrière-plan. On peut même définir leurs positions absolues sur la page. Donc un bloc est une boîte qu'on remplit avec un contenu selon les règles que l'on choisit.
  - Evidemment, il est possible de changer de type, par exemple afficher une liste à puce comme une suite de phrases.

### Exemple 5-1: CSS positioning - `<li>` inline

[url: http://tecfa.unige.ch/guides/css/ex/simple-inline.html](http://tecfa.unige.ch/guides/css/ex/simple-inline.html)

Voir aussi 7. "Les boîtes CSS et leur positionnement" [25]

## 6. Attributs du texte

Les attributs les plus utilisés sont les suivants (pleins d'autres !) (voir les pointeurs tecfa)

### 6.1 Attributs pour la gestion des propriétés d'écriture

Attributs	Valeurs	se charge de	exemple
<b>font-family</b>	nom de l'écriture	police	<code>font-family: Helvetica;</code>
	serif	famille de police, écriture de type serif	<code>font-family: Times, serif;</code>
<b>font-size</b>	pt, cm	taille de police	<code>font-size: 14pt;</code>
<b>font-style</b>		forme d'écriture	
	italic	italique	<code>font-style: italique;</code>
<b>font-weight</b>	de 100 à 900	épaisseur	<code>font-weight: 500;</code>
	normal	correspond à la valeur 400	<code>font-weight: normal;</code>
	bold	correspond à la valeur 700	<code>font-weight: bold;</code>

## 6.2 Alignement du texte

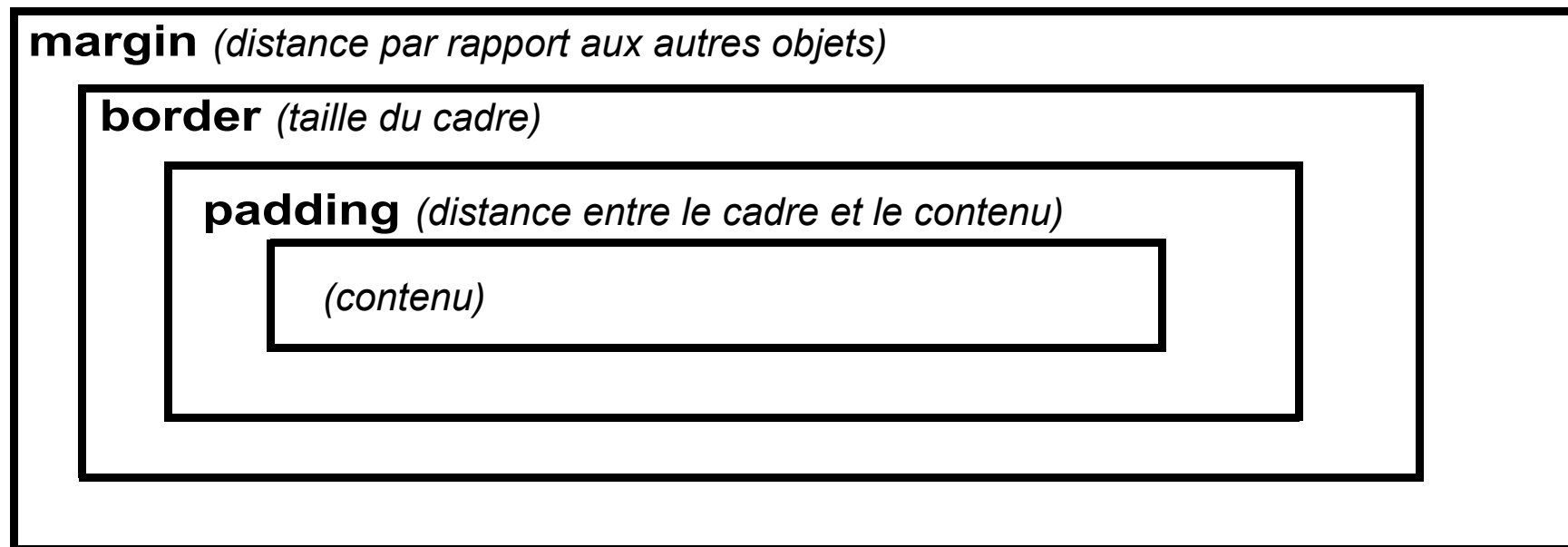
Attributs	Valeurs	se charge de	exemple
<b>text-align</b>		alignement des paragraphes	
	left	aligné à gauche, valeur par défaut	<code>text-align: left;</code>
	center	centré	<code>text-align: center;</code>
	right	aligné à droite	<code>text-align: right;</code>
	justify	justifié en forme de bloc	<code>text-align: justify;</code>
<b>text-indent</b>	pt, cm	retrait de la première ligne	<code>text-indent: 1cm;</code>
<b>line-height</b>	pt, cm	hauteur des lignes	<code>line-height: 14pt;</code>
	valeur relative	écart entre les lignes par rapport à la taille de l'écriture	<code>line-height: 1.2;</code>



# 7. Les boîtes CSS et leur positionnement

## 7.1 Anatomie des boîtes

Une boîte (chaque élément "bloc" sauf les tables et autres exceptions qui ont des priorités à part) possède l'anatomie suivante:



Chaque élément correspond à un selecteur CSS qui permet de définir la largeur générale (des 4 cotés) ou encore la largeur de chaque côté. On peut aussi définir dessin et couleur du cadre.

## 7.2 Les bords, les cadres et le couleur

Attributs	Valeurs	se charge de	exemple
<b>margin</b>	pt, px, cm, %	4 marges	<code>body {margin:1cm;}</code>
<b>margin-top</b>		marge en haut	<code>p {margin-top:10px;}</code>
<b>margin-bottom</b>		marge en bas	<code>h3 {margin-bottom:3pt;}</code>
<b>margin-left</b>		marge à gauche	<code>img {margin-left:50px;}</code>
<b>margin-right</b>		marge à droite	<code>p.citation {margin-right:10pt;}</code>
<b>border</b>	pt,px, cm, %	largeur du cadre	<code>p {border:5px;}</code>
<b>border-top</b>			<code>h1 {border-top:0.2cm;}</code>
<b>etc ...</b>			
<b>border-style</b>		style de cadre	
	solid	ligne simple	<code>p {border-style:solid;}</code>
	double	ligne double	<code>h1 {border-style:double;}</code>
<b>padding</b>	pt,px,cm,%,etc	marge intérieures	<code>p {padding: 5px;}</code>
<b>color</b>	valeur hexa / nom	couleur d'un élément	<code>#menu {color:#000000;}</code> <code>body {color:blue;}</code>
<b>background</b>	aussi	couleur de l'arrière-plan	<code>h1, h2 {background:silver;}</code>

- Ceci n'est qu'un aperçu !
- Il existe notamment des "shortcuts" qu'on montre sur la page suivante.

## Les bords, les cadres et la couleur (shortcuts)

Exemple:

```
#content {  
border-right:2px dotted black;  
border-bottom:2px dotted black;  
color: #000;  
background-color:#ffffcc;  
padding: 5px 5px 5px 5px;    /* haut, droite, bas, gauche */  
margin:5px 15px 5px 5px;  
}
```

```
H1 {  
    /* 1 cm en haut et en bas, 2cm sur les côtés */  
margin: 1cm 2cm;  
    /* 3em en haut, 20% sur les côtés, et 2em en bas */  
padding: 3em 20% 2em;  
}
```

## Exemple 7-1: Plusieurs exemples simples avec des boîtes

[url: http://tecfa.unige.ch/guides/css/ex/boxing0.html](http://tecfa.unige.ch/guides/css/ex/boxing0.html)

- la boîte pour le "body" a des lignes pointillés
- la boîte pour un <p> a une petite marge, un bord solide, et une couleur en arrière-plan
- une partie de ce même paragraphe est mis dans une boîte avec un "span"

```
<body style="border-style:dotted">
<h1>Boîtes</h1>
<p>The dashed box is for the "body", the big thick box is for a "p" tag, and the thin
box is used within a "span" tag. (View the source).</p>
<p style="background-color:#d0d0d0;border-style:solid;border-width:4px;
padding:1em;margin:0.1cm;">
  Tout est une boîte - DKS nov 2003
  <span style="background-color:#d0d0d0;border-style:solid;border-width:1px;
padding:1pt;margin:1pt;width:50%;position:relative">
    <a href="http://validator.w3.org/check/referer">XHTML validation</a> -
    <a href="http://jigsaw.w3.org/css-validator/check/referer">CSS validation<a>
  </span>
</p>
<p>Voir <a href="boxing1.html">boxing1.html</a> pour plus ... </p>
</body>
```

## 7.3 Le positionnement

- Par défaut les éléments d'une page HTML sont affichés séquentiellement un après l'autre
- Mais on peut les positionner selon des coordonnées absolues ou relatives n'importe où sur une page (voir les exemples qui suivent)

```
#content{/* les coordonnees selon l'horloge: 12, 3, 6, 9 */
  float:left;
  width:80%;
  color: #000;
  background-color:#ffffcc;
  padding: 5px 5px 5px 5px;
  margin:5px 5px 5px 5px;
}
#menu{ /* qui sera mis à droite */
  position: absolute;
  right: 0;
  width: 17%;
  font:10px/14px verdana, sans-serif;
  color:black;
  margin:5px 5px 5px 5px;
  background-color: #ffffff;
}
```

- Voir le prochain slide pour une série d'exemples

## **Exemple 7-2: Positionnement relatif de boîtes flottantes**

[url: http://tecfa.unige.ch/guides/css/ex/boxing1.html](http://tecfa.unige.ch/guides/css/ex/boxing1.html)

## **Exemple 7-3: Positionnement relatif horizontal + vertical**

[url: http://tecfa.unige.ch/guides/css/ex/boxing2.html](http://tecfa.unige.ch/guides/css/ex/boxing2.html)

## **Exemple 7-4: Un menu: "ul/li" comme boutons**

[url: http://tecfa.unige.ch/guides/css/ex/boxing3.html](http://tecfa.unige.ch/guides/css/ex/boxing3.html)

## **Exemple 7-5: Positions absolues**

[url: http://tecfa.unige.ch/guides/css/ex/boxing4.html](http://tecfa.unige.ch/guides/css/ex/boxing4.html)

## **Exemple 7-6: Layout 3-colonnes**

[url: http://tecfa.unige.ch/guides/css/ex/boxing5.html](http://tecfa.unige.ch/guides/css/ex/boxing5.html)

## 8. XML avec CSS

### 8.1 Association d'une feuille de style

- Il est conseillé d'utiliser une feuille externe et ensuite l'importer:

```
<?xml-stylesheet type="text/css" href="feuille.css"?>
```

- note: pas de feuilles de styles internes comme pour HTML !

Voici le début d'un fichier:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<?xml-stylesheet href="stepbystep.css" type="text/css"?>
```

```
<!DOCTYPE Stepbystep SYSTEM "stepbystep-ex.dtd">
```

```
<Stepbystep xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<Doctitle>ATTENTION: Il ne s'agit ici que d'une démo XML + CSS. L'original de ce fichier  
se trouve  
ailleurs et a été modifié sans doute depuis ....
```

```
</Doctitle>
```

```
<Info>
```

```
<Author Email="Stephane.morand@tecfa.unige.ch">
```

```
</Author>
```

```
<Version>Version 0.2</Version>
```

```
<Para>installation brute de postnuke sur le serveur tecfaseed
```

```
</Para>
```

```
</Info>
```

## 8.2 Selecteurs CSS2 pour XML et HTML

- XML nécessite un navigateur qui supporte CSS2 (au moins en partie)
- A part la notation `Balise.classe` les sélecteur XML et HTML sont les mêmes !

### **sélection d'un élément**

Syntaxe: `nom_de_l'élément`

Exemple:

```
Step {  
    display: list-item;  
    list-style-type: decimal;  
}
```

### **sélection d'un élément qui est l'enfant direct d'un élément**

Syntaxe: `élément_mère > élément`

Exemple:

```
Step > Title { .... }
```

### **sélection d'un élément qui est le descendant d'un élément**

Syntaxe: `élément_mère élément`

Exemple:

```
Step Title { .... }
```

Dans l'exemple suivant P est un descendant de LI, LI doit être un enfant direct de OL. OL est dans DIV.

```
DIV OL>LI P
```



## **sélection d'un élément qui est le frère d'un élément**

Syntaxe: `élément_frère + élément`

Exemple:

```
H1 + H2 { margin-top: -5mm }
```

(on réduit la distance)

## **sélection d'un élément qui possède un attribut**

Syntaxe: `élément[attribut]`

Exemple:

```
Title[status] { color: blue; }
```

(tous les titres qui un attribut "status" sont peints en bleu )

## **sélection d'un élément qui possède un attribut avec une valeur**

Syntaxe: `élément[attribut="valeur"]`

Exemple:

```
Title[status="brouillon"] { color: red; }
```

(tous les titres qui un attribut "status" avec valeur "brouillon" sont peints en rouge )

Note: au lieu de "=", on a aussi ~=" et != (voir la documentation)

## 8.3 Premières opérations à faire

- Il faut d'abord indiquer pour chaque élément s'il est un "block" ou "inline"
- Faire sortir les titres
- Gérer les listes

### Exemples

```
/* title et para sont des éléments "block", ils ont une petite marge
title, para {display: block; margin: 0.5em;}
/* les title sont un peu plus grands */
title {font-size: 1.5em;}
/* les item sont des list-item de type bullet */
item {display: list-item;list-style-type: bullet;}
/* strong est un élément inline, rendering on italic et bleu */
strong {display: inline; font-style: italic; color: rgb(000,000,128);}
```

### Exemple 8-1: Exemple "Stepbystep"

[url: http://tecfa.unige.ch/guides/css/ex-xml/stepbystep/](http://tecfa.unige.ch/guides/css/ex-xml/stepbystep/)

(répertoire avec tous les fichiers)

- Le DTD Stebbystep permet de formater des instructions de type "pas par pas", par exemple comment installer un logiciel.

### Exemple 8-2: Récit

[url: http://tecfa.unige.ch/guides/xml/examples/recit/](http://tecfa.unige.ch/guides/xml/examples/recit/)

( répertoire avec une solution CSS et une solution XSLT )

- La DTD "RECIT" permet d'écrire des simples récits avec une grammaire génératrice.