Cascading Style Sheets

Code: css-intro

Originaux

url: **http://tecfa.unige.ch/guides/tie/html/css-into/css-intro.html**

url: **http://tecfa.unige.ch/guides/tie/pdf/files/css-intro.pdf**

Auteurs et version

- Daniel K. Schneider - David Ott - Vivian Synteta

- Version: 0.4 (modifié le 15/8/01 par DKS)

Prérequis

Module technique précédent: html-intro

Module technique précédent: xml-tech (pour les chapitres sur XML seulement)

Autres modules

Module technique suppl.: xml-xslt

© TECFA 15/8/01

Abstract

Petite introduction à CSS

Attention: sans description des propriétés (consultez un autre document !)

Objectifs

- Savoir faire de simples style sheets HTML avec CSS
- XML avec CSS
- (Pas de DHTML pour le moment, ce sera fera dans un autre module je pense)

© TECFA 15/8/01

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 (section à déplacer) 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 Importation 11
3.5 Feuille de style client-side 11
4. Sélecteurs pour HTML 12
4.1 Simple selecteurs pour les balises 12
4.2 Les balises et 13
4.3 Selecteurs contextuels 14
4.4 Pseudo sélecteurs (élément et classes) 15
A.Pseudo-éléments 15
B.Pseudo-classes 15
4.5 Class et attributs ID 16
A.Les selecteurs "class" 16
5. Les déclarations 17
6. XML avec CSS 18
6.1 Association d'une feuille de style 18
6.2 Selecteurs CSS2 pour XML et HTML 18

 © TECFA 15/8/01

6.3 Premiers pas **20**

© TECFA 15/8/01

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, ...)
  - permet de gérér centralement le "look" de pleins de pages, donc diminue le coût de maintenance
  - rends un page plus "lisible" et plus rapide à télécharger

Désavantages

- Mauvaise implémentation avec Netscape 4
- Médiocre en NS 4.7 et IE 5 (suffisante pour mises en pages simples)
- Potable dans IE 5.5 et NS 6, presque bonne dans Mozilla 0.9

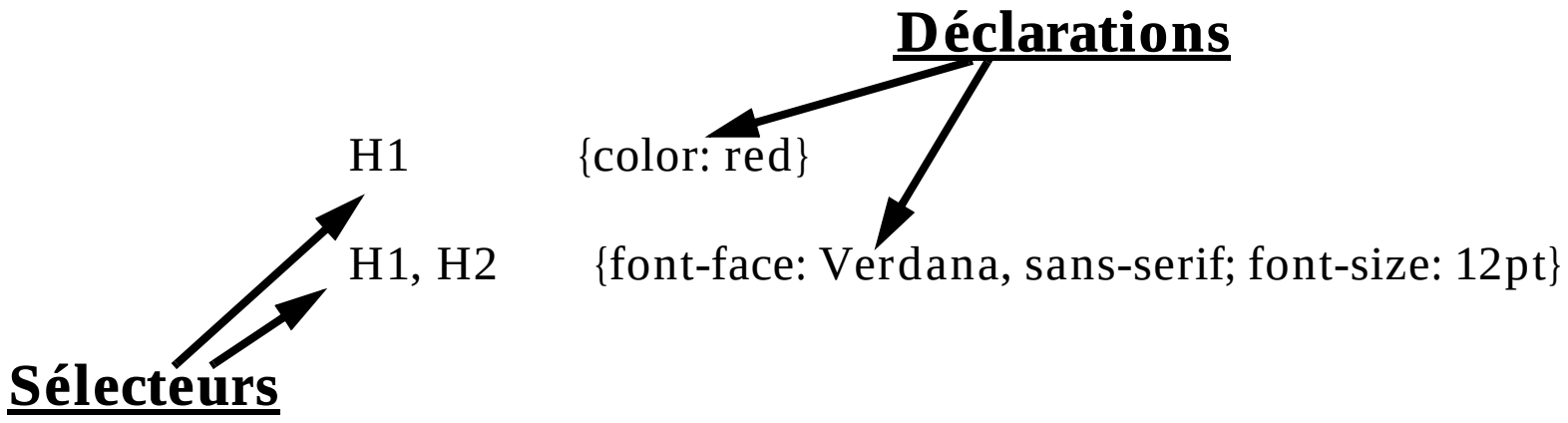
Il existe des tables de compatibilités, par exemple:
url: **http://webreview.com/wr/pub/guides/style/mastergrid.html**

2.2 Principe de base

- Feuille de style = jeux de règles qui précise l'affichage d'un élément HTML
- 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**

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

url: **http://style.webreview.com/** *(style at webreference)*

url: **http://webreview.com/wr/pub/guides/style/glossary.html** *(overview of the CSS spec)*

2.4 Conseils (section à déplacer)

- Mettez des balises fermantes (..... , <li> .... , etc)
- Testez avec Mozilla pour voir si votre code est juste, ajustez ensuite pour Netscape et IE Explorer.

Pour être compatible:

- Manipulations sophistiqués uniquement avec et
- Sinon, utilisez des classes
- Pour gérer les anciens browsers (NS 4 et IE 5) comprénant 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

© TECFA 15/8/01

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**
- 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 présentation
- Je pense qu'il s'agit aussi d'une tentative pour abolir la balise <font> problématique dans le versions futurs de HTML (XHTML)
  - voir

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**
url: **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 absolument 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 gérér 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 (à vérifier)
- Pensez à créer un répertoire central pour vos styles !
  - A tecfa: /web/styles

Exemple 3-2: Simple feuille CSS externe

url: **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>
```

3.4 Importation

- Il s'agit d'une alternative au "linking" ci-dessus
  url: **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) !

3.5 Feuille 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. :)

© TECFA 15/8/01

4. Sélecteurs pour HTML

- Ici, on ne discute que les sélecteurs, c.a.d. la définition des éléments à qui on applique une règle.
- [à faire: priorités !]

Rappel de la syntaxe pour une règle CSS:

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

- Il existe plusieurs méthodes pour identifier les éléments

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}
```


© TECFA 15/8/01

4.2 Les balises et

- ont été créés spécialement pour un usage avec les feuilles de style
- peut contenir toutes les autres balises HTML et donc servir à mettre en forme toute une section d'une texte (mais attention aux priorités!)
- avec l'attribut class peut servir à faire des "custom tags" (voir plus loin)
- sert à changer une séquence de caractères à l'intérieur d'une balise

Exemple 4-1: Quelques usages de div

url: **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>
```

Exemple 4-2: Faire des backgrounds

url: **http://tecfa.unige.ch/guides/css/ex/simple-div2.html**

```
    <style>
      div.important {
        background: rgb(204,204,255);
        padding: 0.5em; border: none;
      }
    </style>
```

4.3 Sélecteurs contextuels

Exemple 4-3: Simple utilisation d'un sélecteur contextuel

url: **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>
```

4.4 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 très mal ces fonctionalités 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 4-4: Pseudo-sélecteurs

url: **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 les liens:

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

4.5 Class et attributs ID

A. Les sélecteurs "class"

- Mécanisme puissant pour:
  - définir de classes de contenu (indépendamment des balises)
  - pour différencier plusieurs variantes d'une même balise

Exemple 4-5: Classes dans une simple feuille de style

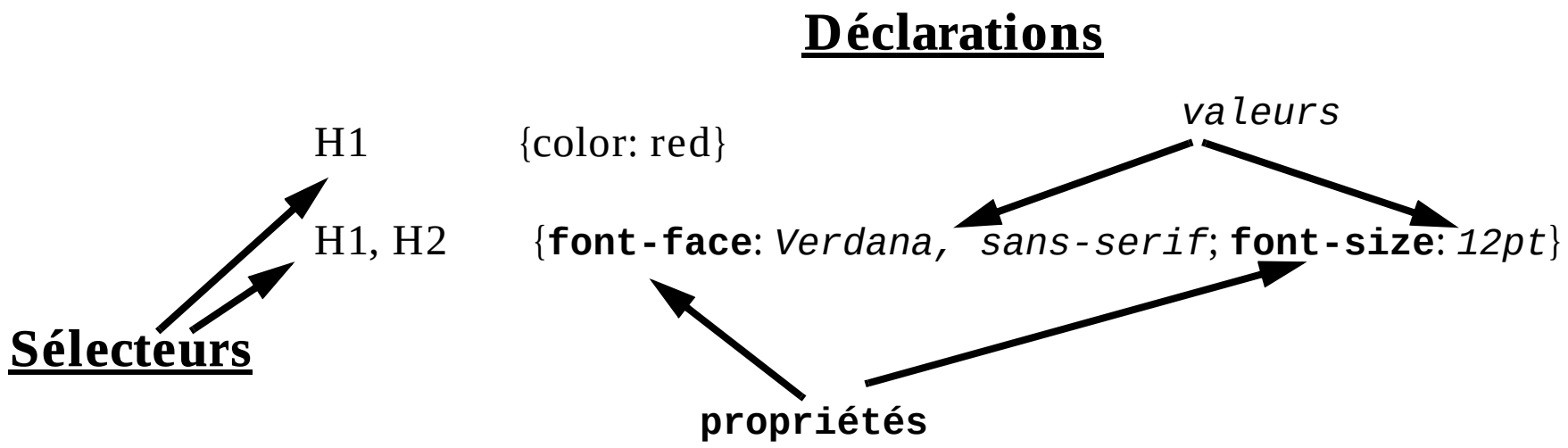
url: **http://tecfa.unige.ch/guides/css/ex/simple-css2.html**

```
 <STYLE type="text/css">
   P.intro {color: blue; margin-left: 4em; margin-right: 2em;}
   .default  {margin-left: 2em;}
  </STYLE>
......
 <p class="intro">
      Après des années de bricolage infâme ....
    </p>
<p class="default">
   bla bla
</p>
<pre class="default">
   notre classe default marche partout, même pour un para préformaté
</pre>
```

5. Les déclarations

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 "propriés: valeurs" sont séparées par des ";"

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

- Types
- Boites
- Positionnement.....

6. XML avec CSS

6.1 Association d'une feuille de style

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

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

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

6.2 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 !

"Wildcard" (s'applique à tous le éléments)

```
Syntaxe: *
* {font-size: 12pt; }
```

sélection d'un élément

```
Syntaxe: nom_de_l'élément
Step {
   display: list-item;
   list-style-type: decimal;
}
```

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

```
Syntaxe: élément_mère  > élément
Step > Title { .... }
```

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

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

(on réduit la distance)

sélection d'un élément qui possède un attribut

```
Syntaxe: élément[attribut]
Title[status] { color: blue; }
```

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

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

```
Syntaxe: élément[attribut="valeur"]
Title[status="brouillon"] { color: red; }
```

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

© TECFA 15/8/01

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

6.3 Premiers pas

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

Exemples

```
/* strong est un élément inline, rendering on italic et bleu   */
strong {display: inline; font-style: italic; color: rgb(000,000,128);}

/* 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;}
```


© TECFA 15/8/01