

Cascading Style Sheets

Feuilles de style en cascade

CSS 2

Francis Drailard - www.antevox.fr

auteur de

Premiers pas en CSS et HTML, éd. Eyrolles

1. Introduction aux feuilles de style

● **HTML** : balises donnant un **sens**

CSS : définition de la **mise en forme**

- **Titre** : dans le code HTML : `<h1>Titre...</h1>`
et dans la feuille de style : `h1 {...mise en forme}`
- **Gras** : `` plutôt que `` ; **Italique** : `` plutôt que `<i>`
- **Menu** : `` et `` pour chaque ligne, mieux que `<p>`

● **Intérêt des feuilles de style**

- Code HTML **plus clair**
- Feuille de style **externes** : - unité graphique pour tout le site
- modifications \Rightarrow pour toutes les pages
- Mise en page **sans tableaux** : blocs positionnés dans la page,
ou les uns par rapport aux autres
- Pages accessibles en **mode texte**

2. Structure du (X)HTML

● Organisation des balises

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

*suite de l'en-tête avec **Feuille de style***

```
</head>
```

```
<body>
```

contenu de la page Web

```
</body>
```

```
</html>
```

*version
HTML
utilisée*

*type de codage
des caractères*

2. Structure du (X)HTML

● Différents types d'éléments

- Éléments de type **bloc**

ils se placent les uns sous les autres, ou à un endroit fixé
<div>, <h1>, <h2>, ..., , , , ...

- Éléments **en ligne** : ils se placent à la suite du texte

- Éléments en ligne "**remplacés**"

leur dimension peut être définie : images

- Éléments en ligne "**non remplacés**"

leur taille est fonction de leur contenu

, , ancre <a...>, , ...

2. Structure du (X)HTML

● Hiérarchie des éléments

```
<body id="ferme">
```

```
<div id="basse-cour">A l'ombre du noisetier...
```

```
<p id="poule">Cot ! Cot ! Cot!</p>
```

```
<p id="canard">Coin ! Coin ! Coin !</p>
```

```
<p id="chien">Ouah ! Ouah !
```

```
<em id="puce">une puce pique le  
chien</em></p>
```

```
</div>
```

```
<div id="enclos">Dans une prairie verte...
```

```
<p id="vache">Meuh ! Meuh !</p>
```

```
<p id="cochon">Groin ! Groin !</p>
```

```
</div>
```

```
</body>
```

A l'ombre du noisetier...

Cot ! Cot ! Cot!

Coin ! Coin ! Coin !

Ouah ! Ouah ! *une puce pique le chien*

Dans une prairie verte...

Meuh ! Meuh !

Groin ! Groin !

Hiérarchie : parents - fils - frères...

Héritage ou non suivant les propriétés
(police : héritée ; bordure : non ; ...)

3. Ecriture des feuilles de style

● Définition d'une **règle de style**

```
h3 { color: blue; font-family: Arial, sans-serif; }
```

sélecteur

1^{ère} déclaration

2^{ème} déclaration

ou

```
h3 {  
    color: blue;  
    font-family: Arial, sans-serif;  
}
```

Commentaires : entre /* et */

● Emplacement des styles

- dans le **code HTML** ⇒ en attribut des balises (à éviter)
- dans l'**en-tête** de la page ⇒ entre <head> et </head>
- dans un **fichier externe** ⇒ fichier à appeler dans l'en-tête

3. Ecriture des feuilles de style

● Feuille de style **interne**

```
<head>
  ...
  <style type="text/css">
  <!--
      h3 {
        color: blue;
        font-family: Arial, sans-serif;
      }
      ...règles de style ici...
  -->
  </style>
  ...
</head>
```

Les commentaires HTML `<!--` et `-->` sont destinés aux anciens navigateurs, qui ne connaissent pas les styles : ils ignoreront ces lignes.

3. Ecriture des feuilles de style

● Feuille de style **externe**

- Fichier texte d'extension **.CSS** (par exemple, *mes_styles.css*)
- Sans les balises `<style>` ni `<!--...-->`
- Appelé dans l'en-tête par :

```
<link rel="stylesheet" type="text/css" href="mes_styles.css" />
```

- ou par :

```
<style type="text/css">  
    @import url(mes_styles.css);  
</style>
```

La commande `@import` permet aussi d'importer une feuille de style à l'intérieur d'une autre feuille de style externe.

3. Ecriture des feuilles de style

● Style **en ligne**

- Style défini dans la balise, en tant qu'attribut :

```
<h2 style="text-align: center; color: red;">...Texte...</h2>
```

- Les styles en lignes sont **à éviter** :
 - Mélange du contenu et de la mise en forme
 - Code HTML moins clair
 - Maintenance du site à nouveau plus difficile

L'utilisation de styles en ligne fait réapparaître les défauts de la mise en forme par des attributs HTML.

3. Ecriture des feuilles de style

● Sélecteurs de style

- Sélecteur **simple**

```
p { font-family: Arial, sans-serif; }
```

- **Classe**

```
p.intro { font-style: italic; }
```

avec dans le code HTML : <p class="intro">

```
.menu { color: blue; }
```

avec dans le code HTML :

```
<p class="menu">, <ul class="menu">, ...
```

- **Identifiant**

```
p#auteur { color: gray; }
```

```
#auteur { color: gray; }
```

avec dans le code HTML : <p id="auteur">

3. Ecriture des feuilles de style

● Sélecteurs de style

- Pseudo-classes

`:active, :link, :hover, :visited,`

```
a:hover { color: red; }
```

⇒ *au passage de la souris sur une balise* `<a>`

```
a.menu:hover { color: red; }
```

⇒ *au passage de la souris sur une balise* ``

- Plusieurs sélecteurs

```
h1, h2, h3.sommaire { text-align: center; }
```

⇒ *pour* `<h1>`, `<h2>` *et* `<h3 class="sommaire">`

3. Ecriture des feuilles de style

● Sélecteurs de style

- **Hiérarchie** des sélecteurs

```
p#sommaire a { color: gray; }
```

⇒ *pour les balises <a...> incluses dans l'élément <p id="sommaire">*
`<p id="sommaire">... <a...></p>`

```
p a { font-weight: bold; }
```

⇒ *pour les balises <a ...> incluses dans un paragraphe <p>*
`<p>... <a...></p>`

4. Propriétés de mise en forme

● Mise en forme des caractères

- Police : **font-family**

```
p { font-family: Arial , Verdana, sans-serif; }
```

```
p { font-family: "Times New Roman", Garamond, serif; }
```

- Taille de police : **font-size**

```
h1 { font-size: 150%; }
```

Taille relative (conseillée) : en *em*, *ex*, % ou *px*

⇒ *Attention aux balises imbriquées : la taille est relative à celle du bloc parent*

Taille fixe (à éviter) : en *pt*, *pc*, *cm*, *mm* ou *in*

Mot-clé (moins précis) : *xx-small*, *x-small*, *small*, *medium* (taille standard), *large*, *x-large*, *xx-large*

4. Propriétés de mise en forme

● Mise en forme des caractères

- Couleur : **color** (*mais pas font-color*)

```
h1.menu { color : #6e05c3; }
```

```
.utile { color: rgb(255,0,0); }
```

```
.remarque { color: rgb(0,0,100%); }
```

```
em { color: green; }
```

- Gras (épaisseur d'écriture) : **font-weight**

```
.principal { font-weight: bold; }
```

Valeurs possibles : *normal, bold, lighter, bolder*

ou nombre *100, 200, ..., 900* (400 = normal ; 700 = bold)

4. Propriétés de mise en forme

● Mise en forme des caractères

- Italique : **font-style**

```
.remarque { font-style: italic; }
```

Valeurs possibles : *normal*, *italic*, *oblique* (*oblique est peu utilisé*)

- Soulignement - décorations : **text-decoration**

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

```
h1 { text-decoration: underline overline; }
```

Valeurs possibles : *none*, *underline*, *overline*, *line-through*, *blink*

4. Propriétés de mise en forme

● Mise en forme des caractères

- Raccourci de propriétés de caractères : **font**

```
.nom_propre h3 { font: bold 1.2em Verdana, sans-serif; }
```

```
.note { font: italic 80% Garamond, serif; }
```

```
#slogan { font: bold italic small-caps 2em Times, serif; }
```

La propriété font permet de définir les valeurs de :

font-style, font-variant, font-weight, font-size, line-height et *font-family*

Ces propriétés sont facultatives, sauf *font-family* qui est obligatoire.

4. Propriétés de mise en forme

● Mise en forme des blocs et paragraphes

- Alignement horizontal du texte : **text-align**

```
p { text-align: justify; }  
.auteur { text-align: right; }
```

Valeurs possibles : *left, right, center, justify*

- Retrait de première ligne : **text-indent**

```
p { text-indent: 5em; }
```

Valeur positive ou négative, qui peut être un pourcentage.

4. Propriétés de mise en forme

● Mise en forme des blocs et paragraphes

- Interligne minimum : **line-height**

```
a.menu { line-height: 2em; }
```

Valeurs possibles : *normal* ou *valeur positive* (% possible)

- Largeur / hauteur minimum : **min-width, min-height**

```
h1 { min-width: 50%; }
```

```
div.remarque { min-height: 50em; }
```

- Largeur / hauteur maxi. : **max-width, max-height**

```
p { max-width: 80%; }
```

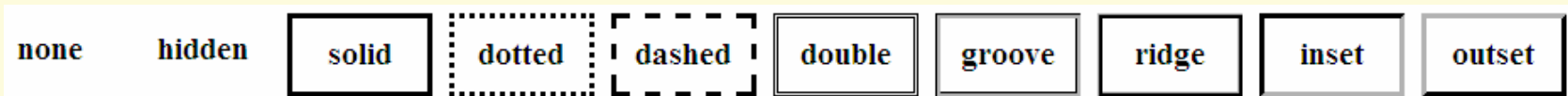
```
#extraits { max-height: 200px; }
```

4. Propriétés de mise en forme

• Bordures

- Style de bordure : **border-style**

```
p.note { border-style: double; }
```



- Détail des bordures pour chaque côté :
border-top-style (en haut), **border-right-style** (à droite),
border-bottom-style (en haut), **border-left-style** (à droite)
- Détail du style de bordure avec **border-style** :

```
p { border-style: solid double; }
```

⇒ ① haut et bas / ② droite et gauche

```
p { border-style: dashed solid dotted; }
```

⇒ ① haut / ② droite et gauche / ③ bas

```
p { border-style: dashed dotted solid double; }
```

⇒ ① haut / ② droite / ③ bas / ④ gauche

4. Propriétés de mise en forme

• Bordures

- Epaisseur de bordure : **border-width**

```
p.note { border-width: 2px; }
```

Valeurs possibles : *thin, medium, thick* ou *valeur numérique* (pas %)

- Détail de l'épaisseur pour chaque côté :

border-top-width (en haut), **border-right-width** (à droite),

border-bottom-width (en haut), **border-left-width** (à droite)

- Détail de l'épaisseur de bordure avec **border-width** :

```
p { border-width: 1em 2em; }
```

⇒ ① haut et bas / ② droite et gauche

```
p { border-width: thin medium thick; }
```

⇒ ① haut / ② droite et gauche / ③ bas

```
p { border-width: 1px 3px 3px 1px; }
```

⇒ ① haut / ② droite / ③ bas / ④ gauche

4. Propriétés de mise en forme

● Bordures

- Couleur de bordure : **border-color**

```
div.remarque { border-color: gray; }
```

Valeurs possibles : *transparent, nom ou code de couleur*

- Détail de la couleur pour chaque côté :

border-top-color (en haut), **border-right-color** (à droite),

border-bottom-color (en bas), **border-left-color** (à gauche)

- Détail de la couleur de bordure avec **border-color** :

```
p { border-color: blue red; }
```

⇒ ① haut et bas / ② droite et gauche

```
p {border-color : blue gray green; }
```

⇒ ① haut / ② droite et gauche / ③ bas

```
p {border-color : blue gray gray blue; }
```

⇒ ① haut / ② droite / ③ bas / ④ gauche

4. Propriétés de mise en forme

● Bordures

- Raccourci pour toutes les propriétés de bordure : **border**

```
h2.chapitre { border: 5px gray double; }
```

Valeurs possibles : toutes les valeurs de *border-width*,
border-style et *border-color*.

Ces valeurs sont facultatives, sauf *border-style* (sinon, pas de bordure)

- Raccourci des propriétés de bordure de chaque côté :
border-top (en haut), **border-right** (à droite),
border-bottom (en bas), **border-left** (à gauche)

4. Propriétés de mise en forme

● Couleurs et images d'arrière-plan

- Couleur d'arrière-plan : **background-color**

```
p.relief { background-color: yellow; }
```

Valeurs possibles : *transparent, nom ou code de couleur*

- Image d'arrière-plan : **background-image**

```
body { background-image: url(images/maison.png); }  
.pub { background-image: url(http://www.sncf.com/logo.gif); }
```

Valeur : *url(nom image avec chemin relatif ou absolu) ou none*

- Raccourci pour les arrière-plans : **background**

```
h1 { background: blue url(logo.png) 50% repeat-x fixed; }
```

Valeurs de *background-color, background-image, background-repeat, background-attachment, background-position* dans un ordre quelconque

4. Propriétés de mise en forme

● Listes à puces ou numérotées

- Type de puce ou de numérotation : **list-style-type**

```
ul { list-style-type: square; }  
ol { list-style-type: upper-roman; }
```

Valeurs possibles :

disc, circle, square ;
decimal, decimal-leading-zero,
lower-roman, upper-roman, lower-alpha, upper-alpha, lower-greek, ...

- Utilisation d'une image comme puce : **list-style-image**

```
ul { list-style-image: url(image/puce.gif); }  
li { list-style-image: url(http://www.top.org/logo.gif); }
```

Valeur : url(nom image avec chemin relatif ou absolu) ou none

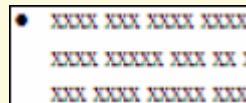
4. Propriétés de mise en forme

● Listes à puces ou numérotées

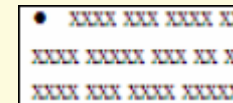
- Position de la puce : **list-style-position**

```
ul { list-style-position: inside; }
```

Valeurs possibles : outside



inside



- Raccourci pour toutes les propriétés de liste : **list-style**

```
ul { list-style: circle inside; }
```

Valeurs de `list-style-type`, `list-style-image` et `list-style-position`.

Ces propriétés sont facultatives (si propriété absente : valeur par défaut)

4. Propriétés de mise en forme

Tableaux

- Largeur fixe ou variable des colonnes : **table-layout**

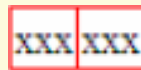
```
table { table-layout: fixed; }
```

Valeurs possibles : *auto* ou *fixed*

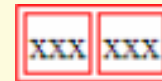
- Recouvrement des bordures : **border-collapse**

```
table, td { border: solid 1px red; border-collapse: collapse; }
```

Valeurs possibles : collapse



separate



- Espacement entre les bordures de cellule : **border-spacing**

(uniquement si "border-collapse: separate;")

```
table { border-collapse: separate; border-spacing: 5px; }
```

```
table { border-collapse: separate; border-spacing: 2px 5px; }
```

Une ou deux valeurs positives (une valeur = pour toutes les bordures ;
deux valeurs = espacement horizontal et vertical)

4. Propriétés de mise en forme

● Tableaux

- Alignement sur la virgule : **text-align**

```
td.nombre { text-align: ","; }  
td.montant { text-align: "&euro;"; }
```

Valeurs possibles : *left, right, center, justify* ou une *chaîne de caractères*
(par exemple ",", pour un alignement sur la virgule)

- Alignement vertical dans les cellules : **vertical-align**

```
.commentaire { vertical-align: top; }  
.titre { vertical-align: middle; }
```

Valeurs possibles : *baseline, top, middle, bottom*

5. Positionnement des éléments

● Marges et dimensions d'un bloc

- Marges extérieures autour d'un bloc :

margin-left, margin-right, margin-top, margin-bottom

```
p { margin-left: 4em; margin-right: 3em; }
```

```
p { margin-top: 5px; margin-bottom: 6px; }
```

- Raccourci pour les marges extérieures : **margin**

```
p { margin: 0; }
```

⇒ même marge de chaque côté

```
p { margin: 3em 5em; }
```

⇒ ① haut et bas / ② droite et gauche

```
p { margin: 5% 10% 8%; }
```

⇒ ① haut / ② droite et gauche / ③ bas

```
p { margin: 15px 10px 20px 15px; }
```

⇒ ① haut / ② droite / ③ bas / ④ gauche

5. Positionnement des éléments

● Marges et dimensions d'un bloc

- Marges intérieures d'un bloc :

padding-left, padding-right, padding-top, padding-bottom

```
p { padding-left: 20px; padding-right: 15px; }
```

```
p { padding-top: 5%; padding-bottom: 10%; }
```

- Raccourci pour les marges intérieures : **padding**

```
p { padding: 0; }
```

⇒ même marge de chaque côté

```
p { padding: 10px 20px; }
```

⇒ ① haut et bas / ② droite et gauche

```
p { padding: 2em 1em 3em; }
```

⇒ ① haut / ② droite et gauche / ③ bas

```
p { padding: 5% 8% 6% 10%; }
```

⇒ ① haut / ② droite / ③ bas / ④ gauche

5. Positionnement des éléments

● Marges et dimensions d'un bloc

- Largeur d'un élément : **width**

```
p div { width: 300px; }  
.menu { width: 20%; }
```

- Hauteur d'un élément : **height**

```
div { height: 50%; }  
img#logo { height: 10em; }
```

Pourcentages pour width et height : % de la largeur / de la hauteur du bloc conteneur (premier bloc parent)

- Niveau d'empilement des blocs : **z-index**

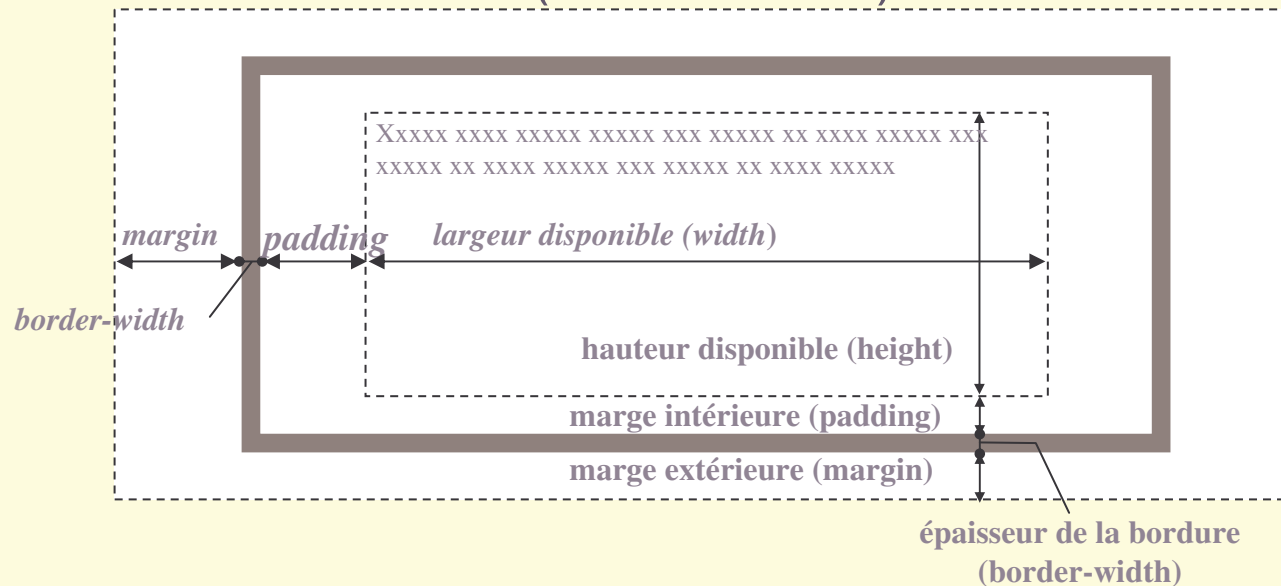
```
.menu { z-index: 10; }  
p#exemple { z-index: -5; }
```

Valeurs possibles : *auto* ou nombre entier (nul, positif ou négatif)

5. Positionnement des éléments

● Marges et dimensions d'un bloc

- Dimensions des boîtes (= des blocs)



Dans le cas simplifié où les *marges horizontales* sont *égales* :

$$\text{Largeur totale} = 2 \times (\text{margin} + \text{border-width} + \text{padding}) + \text{width}$$

Dans le cas simplifié où les *marges verticales* sont *égales* :

$$\text{Hauteur totale} = 2 \times (\text{margin} + \text{border-width} + \text{padding}) + \text{height}$$

5. Positionnement des éléments

● Position des éléments

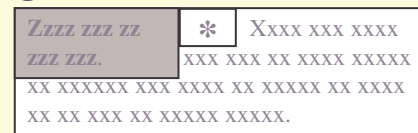
● Flux normal des éléments

- Le **flux normal** est la façon dont les éléments se placent, à l'intérieur de leur conteneur (bloc dans lequel ils sont inclus).
- Les **éléments en ligne** (texte, images, ...) sont placés **les uns à la suite des autres** sur la ligne (retour automatique en fin de ligne).
- Les **éléments de type bloc** sont placés **les uns sous les autres**.
- **Dimensions** par défaut d'un bloc dans le flux normal :
 - *Largeur* = celle de son *bloc conteneur*
 - *Hauteur* = celle de son *contenu* (zéro s'il est vide)
- Les *blocs successifs* sont séparés entre eux par leurs **marges extérieures** (ces marges sont **fusionnées** entre deux blocs).

5. Positionnement des éléments

● Position des éléments (blocs ou éléments "remplacés")

- Position normale, relative, absolue ou fixe
 - Dans le **flux** (pas de positionnement) : le plus souvent possible
 - Position **relative** : décalage par rapport à la position normale du bloc
 - Position **absolue** : fixée à l'intérieur du bloc conteneur
 - Position **fixe** : par rapport à l'écran
 - Éléments **flottants** : juxtaposés à gauche / à droite (côte à côte)



Deux boîtes flottantes à gauche,
habillées par le texte qui suit.

● Attention

- Positionnement d'un bloc imbriqué : son conteneur doit être positionné, sinon, positionnement par rapport au premier bloc parent positionné
- Ordre de superposition : *z-index* (si bloc transparent, ceux du dessous sont visibles)

5. Positionnement des éléments

● Position des éléments

- Type de positionnement d'un bloc : **position**

```
p.note { position: relative; ...positionnement ... }  
div.menu { position: absolute; ...positionnement ... }
```

Valeurs possibles : *static, relative, absolute, fixed*



- Positionnement d'un bloc (décalage) : **top, bottom, left, right**

```
p.note { position: relative; top: 5px; left: 10px; }  
div.menu { position: absolute; top: 30%; right: 20%; }
```

Valeurs possibles : *auto* ou valeur de décalage en *em, ex, %, px ...* (>0 ou <0)

5. Positionnement des éléments

● Position des éléments

- Transformation d'un bloc en bloc flottant : **float**

```
img.vignettes { float: left; }  
div.infos { float: right; }
```

Valeurs possibles : *none, left, right*

- Interdiction de boîtes flottantes sur le côté : **clear**

```
div.menu h1 { clear: both; }  
.remarque { clear: left; }
```

Valeurs possibles : *none, left, right, both*

5. Positionnement des éléments

● Position des éléments

- Type d'affichage d'un élément : **display**

```
p.secret { display: none; }
```

```
span.bloc { display: block; }
```

Valeurs possibles : *none, inline, block, list-item, inline-block, run-in, table, inline-table, table-cell, table-row, table-column, table-caption, ...*

● Forme du curseur

- Modification de la forme du curseur au passage de la souris : **cursor**

```
img { cursor: url(logo.gif); }
```

```
.aide { cursor: help; }
```

5. Positionnement des éléments

● Délimitation des blocs

- La position et les dimensions des blocs qui constituent une page Web ne sont pas toujours évidents.
- **Astuce** pour visualiser un ou plusieurs blocs
⇒ appliquer temporairement le style suivant :

```
border: 1px solid red;
```

- Cette manipulation permet :
 - de corriger facilement une feuille de style
 - de déceler les bogues d'un navigateur

6. Autres medias

- **Média paginé**

⇒ pour l'impression

- **Média sonore**

⇒ fonctions audio

7. Editeurs (X)HTML - CSS

● Gratuits

- PsPad
- Aptana
- Kompozer, Nvu
- Notepad++
- ...

● Payants

- WebExpert
- Dreamweaver
- ...

8. Sites Web

- *World Wide Web Consortium* (normes Web) : <http://www.w3c.org>
- *Spécifications CSS2 en français* :
<http://www.yoyodesign.org/doc/w3c/css2/indexlist.html>
- *OpenWeb* : <http://openweb.eu.org/css/>
- *Pompage* : Web design puisé à la source <http://pompage.net/>
- *SelfHTML* : <http://fr.selfhtml.org/>
- *Alsacrérations* : <http://www.alsacreations.com>
- *Aidenet* : <http://www.aidenet.com>
- *Framasoft* (logiciels et documentation) : <http://www.framasoft.net>
- *Zen Garden* : <http://csszengarden.com/tr/francais>