

HTML 5

**Une référence
pour le développeur web**

Rodolphe Rimelé

Préface de Raphaël Goetter

© Groupe Eyrolles, 2011, ISBN : 978-2-212-12982-3

EYROLLES



Feuilles de style CSS

B

Tout n'est que cascade, style et volupté. Ainsi, du moins, devrait se présenter tout document HTML.



Figure 2-1 background:red ; color:white

Dans des temps reculés, lors de l'apparition du HTML et des premiers navigateurs, l'esthétisme des pages était très limité. L'essentiel de l'information n'était pas destiné à un large public et on pouvait se contenter d'une mise en pages sommaire. Peu à peu, des balises ont été ajoutées, et certains éléments ont été détournés de leur usage initial par les webdesigners/intégrateurs pour obtenir un rendu plus complexe à l'écran.

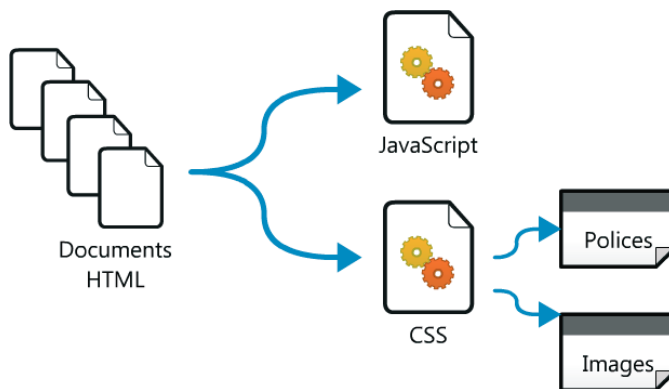
Ainsi, les tables ont été mises à profit pour disposer du contenu en colonnes puis en grilles, on a vu naître des artifices tels que les images de séparation (nommées *spacer.gif*) ou encore l'imbrication de multiples balises de présentation (par exemple ``) aujourd'hui obsolètes.

Ce qui palliait un manque intrinsèque du HTML en capacités de contrôle graphique est devenu au fur et à mesure un cauchemar en termes de maintenance du code, de lisibilité et d'accessibilité du contenu. La sémantique n'était que peu considérée, et la « soupe de tags » au menu de nombreux sites.

C'est pourquoi les feuilles de style en cascade (*Cascading Style Sheets*) ont été adoptées par le W3C pour régir l'apparence des éléments HTML d'une page, concernant la disposition, les dimensions, les couleurs, afin de séparer le contenu et la forme.

Une feuille de style est applicable à une infinité de documents HTML, ce qui en facilite la maintenance et réduit les temps de chargement.

Figure 2-2
Une feuille CSS pour
de multiples pages HTML



L'objet de ce chapitre n'est pas de dresser un inventaire exhaustif de toutes les techniques afférentes à CSS, mais de proposer une introduction et un petit aide-mémoire vis-à-vis de HTML 5 auquel les feuilles de style sont intimement liées.

RESSOURCES

- Goetter Raphaël, *CSS 3, Pratique du design web*, Eyrolles 2011
- Goetter Raphaël, *CSS avancées*, Eyrolles 2011

Les spécifications sont nombreuses et pour certaines encore en mouvement.

RESSOURCE Spécifications CSS

Cascading Style Sheets Level 2 Revision 1 (CSS 2.1)

▶ <http://www.w3.org/TR/CSS21/>

Tableau récapitulatif des spécifications

▶ <http://www.w3.org/Style/CSS/current-work>

Cascading Style Sheets (CSS)

▶ <http://www.w3.org/TR/CSS/>

Principe général

L'application d'une feuille de style CSS s'effectue lorsque celle-ci est liée au document par la balise `<link>`, présente dans la section `<head>`.

```
<link rel="stylesheet" href="styles.css">
```

Plusieurs feuilles de style peuvent être chargées de la sorte pour un seul document. Il est aussi envisageable de déclarer directement des instructions CSS entre les balises `<style>` et `</style>`, situées également dans `<head>` ou dans une portion du document, mais cette façon de faire ne facilite pas la maintenance ni la mise en cache.

Dans le fichier de la feuille de style, on retrouve une ou plusieurs déclarations CSS. Elles comprennent un sélecteur dont le rôle est de cibler les éléments concernés par chaque déclaration, suivi d'un bloc entre accolades regroupant les propriétés à appliquer.

```
p {
  text-align: center;
  font-weight: bold;
  color: orange;
}
header img {
  border: 3px solid gray;
  margin-bottom: 3em;
}
```

Dans cet exemple, trois propriétés sont appliquées aux éléments `<p>` (les paragraphes) et deux aux éléments `` situés dans `<header>`.

Sélecteurs

L'art d'écrire de bons sélecteurs est somme toute un grand jeu d'assemblage parmi les quelques briques de base existantes, pour s'adapter à la structure du document HTML qui doit être stylé. Si HTML et CSS sont issus du même auteur, alors il lui appartient de nommer les éléments en conséquence grâce à des classes, des identifiants ou des attributs pour les lier aux déclarations CSS, et ce en toute simplicité pour faciliter la lecture du code source ainsi que son analyse par le navigateur.

Tableau 2-1 Quelques sélecteurs

Sélecteur	Exemples	Éléments concernés
Sélecteur d'élément	<code>nav { }</code> <code>p { }</code> <code>ul { }</code>	<code><nav></code> <code><p></code> <code></code>
Sélecteur de classe	<code>.remarque { }</code> <code>div.remarque { }</code>	Tout élément portant l'attribut <code>class="remarque"</code> <code><div class="remarque"></code> .
Sélecteur d'id	<code>#intro { }</code> <code>header#intro { }</code>	Tout élément portant l'attribut <code>id="intro"</code> <code><header id="intro"></code> .
Sélecteur d'attribut	<code>[alt] { }</code> <code>input[type=submit] { }</code> <code>[rel=nofollow] { }</code>	Tout élément possédant un attribut <code>alt</code> <code><input type="submit"></code> . Tout élément portant l'attribut <code>rel="nofollow"</code> .
Sélecteur d'enfant direct	<code>ul>li { }</code>	Tout élément <code></code> enfant direct d'un <code></code> .
Sélecteur d'élément adjacent	<code>h1+p { }</code>	Tout élément <code><p></code> immédiatement précédé par un élément <code><h1></code> .
Sélecteur d'éléments frères	<code>h2~p { }</code>	Un ou plusieurs éléments <code><p></code> précédés par un élément <code><h2></code> .

Les sélecteurs peuvent s'enchaîner, séparés par des espaces, pour combiner leurs effets.

```
nav ul>li { ... }
```

Ce sélecteur s'adresse aux éléments `` enfants directs de ``, lui-même situé dans un quelconque `<nav>`.

```
#intro p.remarque { ... }
```

Ce sélecteur s'adresse aux éléments `<p>` de classe `remarque`, situés dans l'élément possédant l'identifiant `intro`.

Séparés par des virgules, les sélecteurs s'appliquent à une énumération de plusieurs familles d'éléments.

```
nav ul>li, #intro p.remarque { ... }
```

Cette déclaration s'adresse aux éléments `` enfants directs de ``, lui-même situé dans un quelconque `<nav>`... et indépendamment aux éléments `<p>` de classe `remarque`, situés dans l'élément possédant l'identifiant `intro`.

Propriétés

Les propriétés sont des paires de clés et valeurs. La clé est le nom de la propriété, suivie des deux-points, terminée par la valeur qui peut revêtir différentes formes : mots-clés, valeur numérique simple, valeur numérique avec unité, chaîne de texte, etc.

Les combinaisons offertes par les propriétés CSS, leurs valeurs et les techniques associées sortent du cadre de ce modeste chapitre qui pourrait à lui seul dépasser tous les autres en longueur. Pour en savoir plus, consultez les références proposées dans l'introduction.

Tableau 2-2 Quelques unités courantes

Unité	Description	Type d'unité
<code>px</code>	Pixels	Relative
<code>%</code>	Pourcentage (en général par rapport à l'élément parent)	Relative
<code>em</code>	Cadratin (relatif à la taille de la police)	Relative
<code>ex</code>	Hauteur de la lettre x	Relative
<code>in</code>	Pouces (= 2,54 cm)	Absolue
<code>cm</code>	Centimètres (= 10 mm)	Absolue
<code>mm</code>	Millimètre	Absolue
<code>pt</code>	Points (= 1/72 in)	Absolue
<code>pc</code>	Picas (= 12 pt)	Absolue

Les unités les plus pertinentes pour un affichage à l'écran restent `px`, `%`, et `em`.

Tableau 2-3 Texte

Propriété	Rôle
<code>color</code>	Couleur du texte
<code>font</code>	(Déclaration groupée)
<code>font-weight</code>	Graisse
<code>font-size</code>	Taille de police
<code>font-family</code>	Famille de police
<code>font-variant</code>	Variante
<code>font-stretch</code>	Étirement du texte
<code>text-decoration</code>	Décoration de texte

Tableau 2-3 Texte (suite)

Propriété	Rôle
<code>text-transform</code>	Transformation de texte
<code>text-align</code>	Alignement
<code>text-justify</code>	Justification
<code>text-indent</code>	Indentation
<code>line-height</code>	Hauteur de ligne
<code>word-spacing</code>	Espacement de mot
<code>word-wrap</code>	Retour à la ligne

Exemple

```
p {
  font-weight: bold;
  color: #005A9C;
  text-align: left;
  line-height: 150%;
}
```

Tableau 2-4 Fonds

Propriété	Rôle
<code>background</code>	(Déclaration groupée)
<code>background-color</code>	Couleur de fond
<code>background-image</code>	Image de fond
<code>background-repeat</code>	Répétition de l'image de fond
<code>background-position</code>	Position de l'image de fond
<code>background-attachment</code>	Attache de l'image de fond par rapport à la page
<code>background-origin</code>	Position du fond par rapport à la boîte de l'élément
<code>background-size</code>	Taille de l'image de fond par rapport aux dimensions de l'élément

Exemple

```
div {
  background-image: url(image.jpg);
  background-repeat: no-repeat;
  background-size: 100% 100%;
  background-origin: content-box;
}
```

RESSOURCE Tutoriels CSS sur Alsacreations

Arrière-plans avec CSS 3 Backgrounds

► <http://www.alsacreations.com/tuto/lire/808-arriere-plans-css3-background.html>

Tableau 2-5 Ombrages et transparence

Propriété	Rôle
<code>text-shadow</code>	Ombrage du texte
<code>text-outline</code>	Contour du texte
<code>box-shadow</code>	Ombrage d'un élément boîte
<code>opacity</code>	Opacité

Exemple

```

footer {
  opacity: 0.5;
  text-shadow: 0px 0px 9px #777;
  box-shadow: rgba(0,0,0,0.4) 10px 10px 0 10px;
}

```

RESSOURCE Tutoriels CSS sur Alsacreations

Les ombrages en CSS 3

▶ <http://www.alsacreations.com/tuto/lire/910-creer-des-ombrages-ombres-css-box-shadow-text-shadow.html>

Tableau 2-6 Bordures

Propriété	Rôle
<code>border</code>	(Déclaration groupée)
<code>border-color</code>	Couleur du bord
<code>border-spacing</code>	Espacement du bord
<code>border-collapse</code>	Fusion du bord
<code>border-style</code>	Style du bord
<code>border-width</code>	Largeur de la bordure
<code>border-radius</code>	Rayon du bord arrondi
<code>border-image</code>	Style de bord avec image
<code>outline</code>	Contour
<code>outline-style</code>	Style du contour
<code>outline-color</code>	Couleur du contour
<code>outline-width</code>	Largeur du contour

Exemple

```

.arrondi {
  border: 5px solid yellow;
  border-radius: 20px;
}

```



```

:focus {
  outline: thick solid #fc0;
}

```

Tableau 2-7 Positionnement et dimensionnement

Propriété	Rôle
<code>display</code>	Mode d'affichage
<code>float, clear</code>	Mode flottant
<code>position</code>	Positionnement
<code>z-index</code>	Ordre de recouvrement « vertical »
<code>overflow</code>	Mode de dépassement de bloc
<code>overflow-y</code>	Mode de dépassement de bloc vertical
<code>overflow-x</code>	Mode de dépassement de bloc horizontal
<code>width</code>	Largeur
<code>height</code>	Hauteur
<code>top</code>	Position par rapport au haut
<code>left</code>	Position par rapport à la gauche
<code>right</code>	Position par rapport à la droite
<code>bottom</code>	Position par rapport au bas
<code>padding</code>	Marge interne
<code>margin</code>	Marge externe
<code>min-height</code>	Hauteur minimale
<code>max-height</code>	Hauteur maximale
<code>min-width</code>	Largeur minimale
<code>max-width</code>	Largeur maximale
<code>vertical-align</code>	Alignement vertical
<code>visibility</code>	Visibilité
<code>rotation</code>	Rotation
<code>rotation-point</code>	Point de rotation

Exemple

```

img.illustration {
  float: right;
  margin-left: 2em;
  max-width: 50%;
  rotation: 10deg;
  rotation-point: bottom left;
}

```

Tableau 2-8 Listes

Propriété	Rôle
<code>list-style</code>	Style de liste
<code>list-style-type</code>	Type de puce pour les éléments de la liste
<code>list-style-image</code>	Image de puce pour les éléments de la liste
<code>list-style-position</code>	Position de la puce

Exemple

```
ul li {
  list-style: disc;
}
```

Tableau 2-9 Autres

Propriété	Rôle
<code>cursor</code>	Apparence du curseur
<code>direction</code>	Direction d'écriture

Exemple

```
input[type=submit]
  cursor: pointer;
}
```

Tableau 2-10 Animations

Propriété	Rôle
<code>animation</code>	(Déclaration groupée)
<code>animation-delay</code>	Délai d'animation
<code>animation-direction</code>	Sens d'animation
<code>animation-duration</code>	Durée d'animation
<code>animation-iteration-count</code>	Nombre d'itérations
<code>animation-name</code>	Nom d'animation
<code>animation-timing-function</code>	Fonction d'accélération

Exemple

```
div {
  animation-name: 'diagonale';
  animation-duration: 3s;
  animation-iteration-count: 5;
}
```

```
@keyframes 'diagonale' {
  from {
    left: 0;
    top: 0;
  }
  to {
    left: 150px;
    top: 200px;
  }
}
```

Tableau 2-11 Transformations

Propriété	Rôle
<code>transform</code>	(Déclaration groupée)
<code>transform-origin</code>	Point d'origine de la transformation
<code>transform-style</code>	Style de transformation
<code>perspective</code>	Effet de perspective
<code>perspective-origin</code>	Point d'origine de la perspective
<code>backface-visibility</code>	Visibilité de l'arrière de l'élément transformé

Exemple

```
h1 {
  transform: rotate(8deg);
}
```

Tableau 2-12 Transitions

Propriété	Rôle
<code>transition</code>	(Déclaration groupée)
<code>transition-delay</code>	Délai de transition
<code>transition-duration</code>	Durée de transition
<code>transition-property</code>	Propriété à laquelle appliquer la transition
<code>transition-timing-function</code>	Fonction d'accélération pour la transition

Exemple

```
img.transition.couleur {
  transition: transform .5s ease-in;
}
img.transition.couleur:hover {
  transform: rotate(10deg);
}
```

RESSOURCE Tutoriels CSS sur Alsacreations

Transitions CSS 3

▶ <http://www.alsacreations.com/tuto/lire/873-transitions-css3-animations.html>

Une petite quantité de nouvelles propriétés CSS 3 ont été implémentées au fur et à mesure dans les moteurs de rendu, officiellement de manière expérimentale, avec des préfixes spécifiques. C'est par exemple le cas de `border-radius` qui existait initialement sous le nom de `-moz-border-radius` pour les navigateurs Gecko (Mozilla), `-webkit-border-radius` pour les navigateurs WebKit (Chrome, Safari). Opera a quant à lui choisi de l'intégrer directement, mais utilise le préfixe `-o-` pour ses propres expérimentations. Une fois tous les tests concluants, la propriété adopte son nom définitif, sans préfixe.

RESSOURCE Tutoriels CSS sur Alsacreations

Les préfixes vendeurs en CSS

▶ <http://www.alsacreations.com/article/lire/1159-prefixes-vendeurs-css-proprietaires.html>

Pseudo-classes et pseudo-éléments

Les pseudo-classes et pseudo-éléments affinent les capacités des sélecteurs. Ils s'écrivent à la suite du sélecteur initial, concaténés par un signe deux-points « : ».

Tableau 2-13 Quelques pseudo-classes et pseudo-éléments

Pseudo-*	Rôle
<code>:link</code>	Lien
<code>:visited</code>	Lien visité
<code>:hover</code>	Élément survolé
<code>:active</code>	Élément actif
<code>:focus</code>	Élément ayant le focus
<code>:first-child</code>	Premier enfant
<code>:last-child</code>	Dernier enfant
<code>:nth-child(n)</code>	n-ième enfant
<code>:nth-last-of-child(n)</code>	n-ième dernier enfant
<code>:nth-of-type(n)</code>	n-ième type d'élément
<code>:nth-last-of-type(n)</code>	n-ième dernier type d'élément
<code>:first-of-type</code>	Premier type d'élément

Tableau 2-13 Quelques pseudo-classes et pseudo-éléments (suite)

Pseudo-*	Rôle
<code>:last-of-type</code>	Dernier type d'élément
<code>:only-child</code>	Enfant unique
<code>:only-of-type</code>	Élément de type unique
<code>:checked</code>	État coché
<code>:enabled</code>	État activé
<code>:indeterminate</code>	État indéterminé
<code>:not(expr)</code>	Négation de l'expression
<code>::first-letter</code>	Première lettre
<code>::first-line</code>	Première ligne

Exemple

```
a:hover {
  text-decoration:underline;
}
p::first-letter {
  font-size: 2em;
}
```

Règles @

Les « règles @ » sont des instructions plus évoluées pouvant se retrouver dans les feuilles de style pour moduler leur comportement ou ajouter des informations qui ne pourraient trouver leur place dans des déclarations classiques.

Tableau 2-14 Quelques règles @

Propriété	Rôle
<code>@import</code>	Importer une autre feuille de style CSS.
<code>@charset</code>	Déclaration de l'encodage des caractères.
<code>@page</code>	Définir des règles générales pour les médias paginés.
<code>@font-face</code>	Importer un fichier de police (fonte) externe.
<code>@media</code>	Définir des requêtes de média.

Exemple

```

@import "autres_styles.css";
@import url("impression.css") print;

@page {
  size: 15cm 20cm;
  margin: 2cm;
  marks: cross;
}

@font-face {
  font-family: maPolice;
  src: url(ToutSaufComicSans.otf);
  font-weight: bold;
}
p {
  font-family: maPolice;
}

@media print {
  body {
    font-size: 2em;
    background: white;
  }
  nav {
    display: none;
  }
}

```

Media queries

Les *media queries* (ou requêtes de média) sont une fonctionnalité bien utile de CSS pour adapter le design et la présentation générale d'une page web selon le périphérique de consultation. Depuis la navigation sur mobiles jusqu'aux plages braille, la présentation est ajustée selon les capacités de rendu, par exemple en modifiant la taille du texte ou son agencement.

La syntaxe d'une *media query* est une suite de conditions à satisfaire. Au besoin, le navigateur évaluera l'expression et chargera les styles y correspondant ou les rechargera dynamiquement s'il y a lieu.

Exemple de media query

```
@media screen and (min-width: 200px) and (max-width: 640px) {  
  section {  
    display: block;  
    clear: both;  
    margin: 0;  
  }  
}
```

Ici, l'on s'adresse à un écran (*screen*) dont la résolution en largeur est comprise entre 200 et 640 pixels. Si le navigateur se retrouve dans cette condition, les éléments de type `<section>` passeront tous en mode bloc, sans aucune marge, pour une meilleure disposition verticale sur un petit écran.

Pour découvrir la magie des *media queries*, consultez l'article en ligne rédigé par votre auteur dévoué.

RESSOURCE Tutoriels CSS sur Alsacreations

Les Media Queries CSS 3

▶ <http://www.alsacreations.com/article/lire/930-css3-media-queries.html>