

# CSS

## Objectifs des feuilles de style

Les feuilles de style (cascading style sheet) complètent HTML, en offrant un langage permettant de définir des propriétés de style et de présentation du contenu de certaines balises. On peut par exemple affecter à certaines balises, une couleur d'arrière-plan, une image d'arrière-plan, des bordures, des polices de caractères spécifiques, etc.... On peut aussi positionner au pixel près dans la fenêtre d'affichage du navigateur certains éléments, et par exemple modifier l'aspect du curseur.

Les styles peuvent s'appliquer à une ou plusieurs balises. Ceci permet de définir un style une seule fois, et de l'appliquer autant de fois que nécessaire, sans avoir à le redéfinir. De plus les styles peuvent être définis à l'intérieur d'une page HTML, mais aussi être définis dans un fichier externe qui sera appelé par les pages HTML qui en ont besoin. Ceci permettra de regrouper dans un seul fichier toute la « charte graphique » et les aspects présentation d'une application web, ce qui bien sûr en simplifiera la maintenance et les évolutions.

Les feuilles CSS sont constituées de styles que l'on définit pour certains éléments HTML. Pour choisir cet ensemble d'éléments HTML auquel doit s'appliquer un style, on utilise dans les CSS ce qu'on appelle des sélecteurs. Pour chaque sélecteur (qui permet de sélectionner les balises HTML concernées par le style), on va définir un style de présentation en utilisant un certain nombre de propriétés auxquelles on va affecter des valeurs. Par exemple, on peut définir un style pour les balises <h1> pour lesquelles on va fixer les propriétés telles la taille de police, la couleur de police, etc...

Les feuilles de style offrent de multiples possibilités. Ce qui est décrit ici concerne uniquement les utilisations les plus courantes.

## Utilisation des feuilles de style

Plusieurs méthodes sont utilisables pour intégrer des feuilles de style en HTML :

### 1. Définir globalement des styles pour un fichier HTML

Les balises `<style...>` permettent de définir des formats CSS. Cette définition se positionne dans l'entête (head) du document.

Exemple :

```
<html>
<head>
  <style type="text/css">
    /* ... ici sont définis les formats ... */
  </style>
</head>
<body>...</body>
</html>
```

### 2. Définir un style global dans un fichier CSS séparé

On peut définir les formats dans un fichier texte (.css) séparé qu'il suffira d'incorporer dans les fichiers HTML que l'on veut. Toute mise à jour du fichier .css ira se répercuter sur toutes les pages HTML utilisant cette feuille de style. C'est la meilleure solution lorsque l'on développe une application web professionnelle (séparation des données et de la présentation).

L'appel du fichier .css s'effectue à l'aide de la balise :

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

Exemple :

```
<html>
<head>
<title>Titre du fichier</title>
<link rel="stylesheet" type="text/css" href="formats.css" />
</head>
<body>
...
</body>
</html>
```

Il est possible de combiner l'appel à un fichier .css externe et la définition d'un style local `<style>`. Dans ce cas, en cas de conflit, les formats directement définis dans `<style>` sont prioritaires.

### 3. Définir des styles directement dans un élément HTML

L'application d'un style peut se faire directement dans une balise HTML.

Exemple :

```
<html>
<head>
<title>Titre du fichier</title>
</head>
<body>
<h1 style="color:blue">...</h1>
</body>
</html>
```

## Définitions des styles

Considérons la déclaration de style suivante :

```
body {
    background-color:#FFFFCC;
    margin-left:100px;
}

h1 {
    font-size:48pt;
    color:#FF0000;
    font-style:italic;
    border-bottom:solid thin black;
}

p,li {
    font-size:12pt;
    line-height:14pt;
    font-family: Arial;
    letter-spacing:0.2mm;
    word-spacing:0.8mm;
    color:blue;
}
```

On déclare des sélecteurs (ici body, h1, p, li). Par "sélecteur" on entend ce qui se trouve devant les accolades. Le sélecteur désigne quelle balise sera affectée par les propriétés qui suivent. Les sélecteurs peuvent être une (ou plusieurs) balises HTML.

C'est le cas de l'exemple ci-dessus. h1 {...} : concerne toutes les balises <h1>.

Il est aussi possible d'avoir des sélecteurs beaucoup plus élaborés. Exemple :

balise1 balise2 {...} : concerne tous les éléments <balise2> contenus dans un élément <balise1>

balise1 > balise2 {...} : concerne tous les éléments <balise2> fils direct d'un élément <balise1>

On peut aussi définir des sélecteurs qui nous sont propres (et que l'on appellera par l'attribut class). Ces sélecteurs ne concerne pas forcément une ou un ensemble de balise, mais uniquement les éléments qui seront 'marqués' comme devant appliquer le style défini. Le marquage se fait par l'attribut **class**.

Par exemple :

```
<html>
<head>
<style type="text/css">
.rouge {
    color: red;
}
</style>
```

```
</head>
<body>
<h3 class="rouge">Titre rouge</h3>
<h3>Titre normal </h3>
<p class="rouge">Texte écrit en rouge</p>
</body>
<html>
```

Dans cet exemple, on définit un format nommé rouge, qui va s'appliquer à toutes les balises (si on avait mis P.rouge, il ne se serait appliqué qu'à la balise P, par exemple).

Toutes les balises dans lesquelles on retrouve l'attribut class="rouge" appliqueront le style défini.

## LES BALISES <div> ET <span>

Les deux balises <div> et <span> permettent d'identifier des parties de document.

<div> est utilisé pour des blocs de texte entiers (et génère un saut de ligne), et <span> est utilisé pour des portions de texte (mots ou lettres). Ces balises, lorsqu'elles sont définies, désactivent les styles précédemment définis. Ils peuvent avoir leur propre style (différent des styles des balises qui les englobent), en utilisant l'attribut class.

Définition des propriétés

Pour chaque sélecteur défini, on va lui attribuer différentes valeurs à certaines propriétés.

Voici la liste des principales propriétés et les valeurs possibles les plus usitées :

### Couleurs

color	Couleur d'avant-plan
background-color	La couleur d'arrière-plan
background-image	Une image à utiliser comme fond

Pour utiliser des dégradés de couleurs, vous pouvez utiliser la syntaxe suivante :

```
background: linear-gradient(white, black);
```

Ou

```
background: linear-gradient(#ff00ff, #66ffff);
```

### Polices

font-family	Une liste de familles de la police à utiliser
p { font-family: Baskerville, "Heisi Mincho W3", Symbol }	
font-style	normal, italic, oblique
font-weight	normal, bold, bolder, lighter
font-size	La taille de la police exprimée soit par une taille absolue déterminée par le visualiseur (xx-small, x-small, small, medium, large, x-large, xx-large), soit par une mesure relative (larger, smaller) soit par une mesure de longueur soit par un pourcentage par rapport à l'élément englobant
font	Raccourci pour toutes les propriétés ci-dessus
p { font: bold italic large Palatino, serif }	

### Texte

text-indent	Indentation de la première ligne d'un bloc
text-align	Alignement horizontal d'un bloc de texte : left, right, center, justify. Peut être également utilisé pour les cellules de tableau.
text-decoration	Attributs additionnels de décoration : underline,

	overline, line-through, blink
text-transform	Modification de la capitalisation d'un bloc de texte : capitalize, uppercase, lowercase

## Espacement

margin-top margin-right margin-bottom margin-left	Largeur de marge. La valeur est soit une longueur, soit un pourcentage de la boîte englobante. margin est un raccourci pour ces quatre spécifications individuelles.
padding-top padding-right padding-bottom padding-left	Largeur de remplissage (cette zone de remplissage entoure la zone d'affichage et est elle-même entourée par la zone de marge). padding est un raccourci pour ces quatre spécifications individuelles.

## Bordures

border-top-width border-right-width border-bottom-width border-left-width	Largeur de bordure. La valeur est soit une longueur soit thin, medium ou thick. border-width est un raccourci pour ces quatre spécifications individuelles.
border-style	Le style de cadre : dotted (pointillé), dashed (pointillé long), solid, double, groove (creusé), ridge (relief)
border-top-color border-right-color border-bottom-color border-left-color	Couleur de bordure. border-color est un raccourci pour ces quatre spécifications individuelles.
border	Raccourci pour les trois propriétés précédentes

## Listes

list-style-type	Type de numérotation de la liste : disc, circle, square, decimal, lower-roman, upper-roman, lower-latin, upper-latin, ...
list-style-image	Une image à utiliser comme marqueur d'élément de liste (surcharge list-style-type).
ul { list-style-image: url(http://png.com/ellipse.png) }	

## Tableaux

caption-side	Position de la légende du tableau : top, bottom, left ou right.
vertical-align	La hauteur d'une rangée dépend de la taille et du placement des cellules qu'elle contient. Une cellule est donc souvent moins haute que la hauteur de la rangée à laquelle elle appartient. Cette propriété détermine l'alignement vertical d'une cellule par rapport à sa rangée : baseline



	(aligne la ligne de base texte de la cellule avec la ligne de base de la rangée), bottom (le bas de la cellule coïncide avec le bas de la rangée), top, middle
border-spacing	Espacement des bordures de cellules. La spécification comprend un ou deux longueurs (une seule longueur s'applique à l'espacement horizontal et vertical), deux longueurs correspondent respectivement à l'espacement horizontal et vertical.
table { border-spacing: 15pt }	
empty-cells	Attributs de bordure pour les cellules vides : border ou noborder.

## LE POSITIONNEMENT ET L’AFFICHAGE D’ELEMENTS

À partir de la version CSS 2.0 il existe différentes mentions de feuilles de style pour positionner exactement des éléments d'une page HTML dans la fenêtre du navigateur, et pour contrôler exactement la position des éléments les uns par rapport aux autres.

On peut positionner les éléments de façon relative ou absolue, élargir des éléments ou superposer des éléments. Le positionnement absolu et relatif de documents vous donne la possibilité de mieux gérer en fonction de l'écran l'apparence de vos pages web.

Propriété	Utilité	Valeur(s)
position	Mode de positionnement	absolute = Positionnement absolu, mesuré à partir du bord de l'élément parent; peut défiler. fixed = Positionnement absolu, mesuré à partir du bord de l'élément parent; reste fixe lors du défilement. relative = Positionnement relatif mesuré à partir de la position de départ de l'élément proprement dit. static = Pas de positionnement spécial, flux normal de l'élément (réglage par défaut).
top	Position à partir du haut	Valeur en cm ou pixel
left	Position à partir de la gauche	Valeur en cm ou pixel
bottom	Position à partir du bas	Valeur en cm ou pixel
right	Position à partir de la droite	Valeur en cm ou pixel
width	Largeur	Valeur en cm ou pixel
min-width	Largeur minimale	Valeur en cm ou pixel
max-width	Largeur maximale	Valeur en cm ou pixel
height	Hauteur	Valeur en cm ou pixel
min-height	Hauteur minimale	Valeur en cm ou pixel
max-height	Hauteur maximale Valeur en cm ou pixel	
overflow	Passage d'éléments au contenu trop important	visible = L'élément sera agrandi de manière à ce que son contenu soit complètement visible dans tous les cas. hidden = L'élément sera coupé s'il dépasse les limites. scroll = L'élément sera coupé s'il dépasse les limites. Le navigateur propose des barres de défilement. auto = Le navigateur Web doit décider en cas de conflit, comment l'élément sera affiché.
direction	Sens de lecture des éléments	ltr = de gauche à droite rtl = de droite à gauche
z-index	Position de la couche en cas de superposition	Valeur numérique. Les éléments dont le numéro est le plus élevé couvrent les éléments avec le numéro le moins élevé
display	Mode d’affichage ou	block = impose un bloc - l'élément crée une

	plutôt non affichage sans perdre de place	nouvelle ligne. inline = impose l'affichage en texte - l'élément est affiché dans le cours du texte. list-item = comme block, mais précédé d'une puce comme dans une liste énumérative. none = L'élément n'est pas affiché et aucune place ne lui sera réservée
visibility	Affichage ou non affichage avec réservation de place	hidden = Le contenu de l'élément est d'abord caché (non affiché). visible = Le contenu de l'élément est d'abord affiché (réglage normal).
clip	Seul un extrait d'un élément est affiché	Coordonnées du rectangle d'affichage rect(x,y,z,w) {clip :rect(0px 130px 130px 0px)}

## VI LES PSEUDOS FORMATS

Les pseudo-formats s'appliquent à l'élément a en HTML, ils permettent de préciser des informations sur les liens. Les pseudos-formats sont les suivants :

a:link = pour les liens aux pages qui n'ont pas encore été visitées

a:visited = pour les liens aux pages qui ont déjà été visitées

**a:hover** = pour les liens sur lesquels l'utilisateur passe avec la souris

a:active = pour les liens en train d'être cliqués

a:focus = pour les liens qui deviennent actifs (focus)

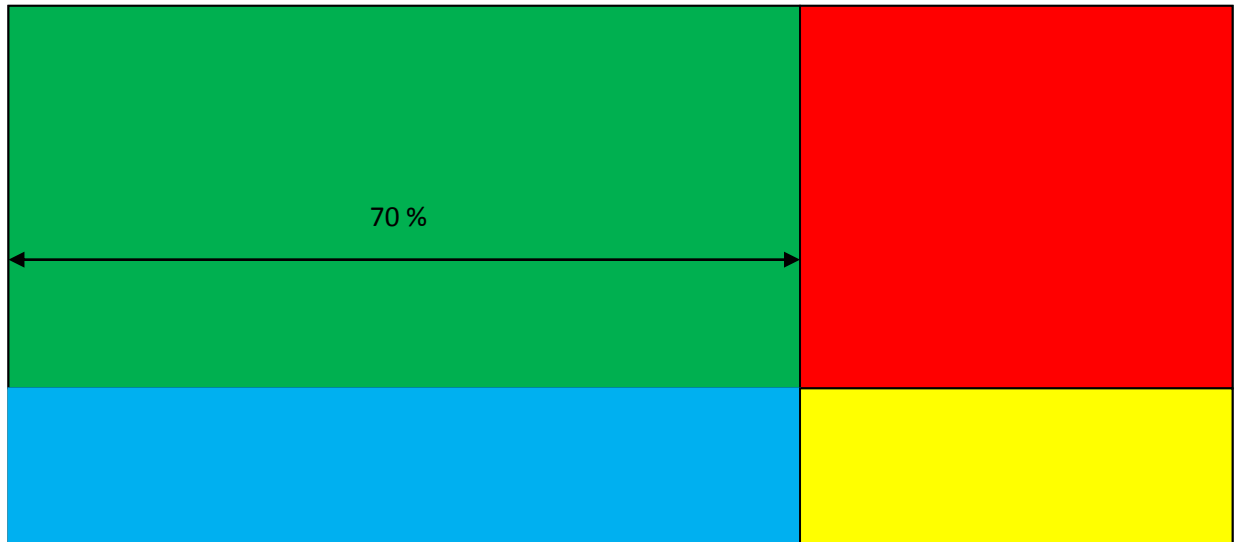
Exemple :

```
<style type="text/css">
a:hover { font-weight:bold; color:#E00000; text-decoration:none }
</style>
```

On peut modifier la forme du curseur, par l'intermédiaire de l'attribut cursor. Les principales différentes valeurs possibles sont : auto, crosshair, hand, move, resize, text, wait, help.....

## Exercices

1) Réalisez une page HTML permettant de représenter le motif suivant



La forme verte doit occuper 70% de la largeur et de la hauteur de la fenêtre.

2) En utilisant HTML et CSS, reproduisez le formulaire suivant

Please complete the form below. Mandatory fields marked \*

Delivery Details	
Name *	<input type="text"/>
Address *	<input type="text"/>
Town/City	<input type="text"/>
County *	<input type="text"/>
Postcode *	<input type="text"/>
Is this address also your invoice address? *	
<input type="radio"/> Yes	
<input type="radio"/> No	

3) Réalisez une page HTML/CSS permettant de reproduire la capture ci-dessous

