

CSS





Introduction

INTRODUCTION

Le langage CSS (Cascading Style Sheets ou feuilles de style en cascade) sert à décrire la présentation des documents HTML et XML.

Dans le domaine du web, le langage CSS (Cascading Style Sheets) est utilisé pour définir l'aspect futur de votre site, comme par exemple la couleur du fond de la page, le type de police ou encore le placement des différentes zones et du texte.

Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C).

Les feuilles de style ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation.

En effet, le HTML offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte, toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent.

Grâce aux feuilles de style, lorsque la charte graphique d'un site composé de plusieurs centaines de pages web doit être changée, il suffit de modifier la définition des feuilles de style en un seul endroit pour changer l'apparence du site tout entier.

LES ENJEUX ET LES AVANTAGES

L'un des objectifs majeurs de CSS est de permettre la mise en forme hors des documents.

Il est par exemple possible et vivement conseillé de ne décrire que la structure d'un document en HTML, et de décrire toute la présentation dans une feuille de style CSS séparée.

Voici plusieurs avantages liés à l'utilisation des feuilles de styles CSS :

- Obtenir une présentation homogène sur tout un site en faisant appel sur toutes les pages à une même définition de style ; aisément modifiable.
- Permettre le changement de l'aspect d'un site complet entier par la seule modification de quelques lignes.
- Possibilité de doter une page de présentations alternatives au choix de l'utilisateur en appliquant différents styles à un même contenu.
- Une plus grande lisibilité du HTML, car les styles sont définis à part. Le code HTML est réduit en taille et en complexité. La structure et la présentation sont donc gérées séparément.
- Allègement des coûts de développement et de maintenance.
- Des chargements de page plus rapides (allègement du code source des pages Web, et donc économie de bande passante).
- Un positionnement plus rigoureux des éléments.
- Possibilité de doter une page de feuilles de style spécifiques selon les medias (navigateurs graphiques, lecteurs d'écran...).
- Création de mise en page flexible s'adaptant aux différentes résolutions d'écran ;
- Meilleure accessibilité des pages Web par la disparition des tableaux détournés à des fins de présentation.

ORIGINE

Les origines des CSS sont liées à trois questions majeures :

- HTML est-il un format de structuration ou un format mixte de mise en forme et de structuration ?
- Les choix de présentation de l'auteur doivent-ils avoir la primauté sur ceux de l'utilisateur ?
- La réponse aux besoins est-elle un format de description de la mise en forme, ou un langage de transformation ?

Groupe de travail et réflexion

Le groupe de travail au sein du W3C, présidé par Chris Lilley comporte notamment des représentants d'Apple, Google, IBM, Microsoft, Adobe, de la Fondation Mozilla et d'Opera.

A large teal circle is partially visible on the left side of the slide, with a thin white line curving around its edge.

DEVELOPPEMENTS

Les spécifications CSS ne sont pas développées par versions successives, qui permettraient à un navigateur de se référer à une version en particulier

CSS est développé par « niveaux », ce qui contraint chaque nouveau niveau à intégrer le précédent, et chaque implémentation à être compatible avec la précédente : CSS1 est donc développé pour être un sous-ensemble de CSS2, qui est lui-même développé pour être un sous-ensemble de CSS3.

CSS 1.0

La spécification CSS1 finale est publiée le 17 décembre 1996, et définit une cinquantaine de propriétés.

CSS1 se définit comme un « mécanisme de feuilles de styles simples, permettant aux auteurs et aux lecteurs d'attacher des styles au document HTML ».

CSS 2.0

Publié comme une recommandation en mai 1998, le second niveau de CSS étend considérablement les possibilités théoriques des feuilles de styles en cascade, avec en particulier environ 70 propriétés supplémentaires.

Depuis le 12 mai 1998, la norme CSS 2.0 est donc le standard en vigueur.

La norme CSS2 apporte un grand nombre de nouvelles fonctionnalités par rapport à la précédente norme (1.0), parmi lesquelles :

- La prise en compte de la diversité de médias.
- L'ajout de propriétés auditives pour les aveugles et malvoyants.
- La manipulation de curseurs.
- La gestion des débordements et de la visibilité des éléments.

CSS 2.1

Les retours d'implémentation de CSS2 conduisent le groupe de travail CSS du W3C à rédiger à partir de 2001 une version révisée CSS 2.1 (« CSS Level 2, Revision 1 »), sur la base de ce qui était effectivement adopté par les différents navigateurs. CSS 2.1 corrige CSS2 sur de nombreux points de détail.

CSS 3

Le développement du troisième niveau des feuilles de styles en cascade commence dès 1999, parallèlement à celui de CSS 2.1.

CSS3 devient « modulaire », afin de faciliter ses mises à jour, mais aussi son implémentation par des agents utilisateurs aux capacités et aux besoins de plus en plus variés : navigateurs graphiques, navigateurs pour mobiles, navigateurs vocaux.



Mise en œuvre



LIER LE LANGAGE CSS AVEC LE LANGAGE HTML

Afin de lier du code CSS avec du code HTML ; Plusieurs méthodes s'offrent à vous.

Les voici dans l'ordre de la moins bonne à la meilleure :

Méthode 1 – CSS dans un attribut HTML

Directement dans le code d'une balise HTML en passant par l'attribut « *style* » :

```
<p style = 'code CSS ici'> un texte </p>
```

Conclusion

Non seulement le CSS sera peu lisible à l'intérieur des balises, mais en plus cela ne nous permet pas de profiter de tous les avantages du CSS, comme par exemple la possibilité de changer la couleur de tous les titres d'un site en un seul coup.

Méthode 2 – CSS dans une balise <style>

Dans la partie « head » de la page entre les balises « style » :

```
<!DOCTYPE html>
<html>
  <head>
    <title> Titre </title>
    <style type = "text/css">
      Code CSS ...
    </style>
  </head>
  <body>
  </body>
</html>
```

Conclusion

Cette méthode n'est pas plus recommandée que la première étant donné que le code CSS inscrit entre les balises « style » sera local à la page et ne pourra pas servir dans les autres pages du site.

Méthode 3 – CSS avec @import

Dans un fichier externe, nommée « style.css ».

Ce ou ces fichiers seront appelés dans l'en-tête du fichier HTML (toujours dans la partie HEAD) sous cette forme :

```
<!DOCTYPE html>
<html>

  <head>

    <title> Titre </title>

    <style type = "text/css">
      <!--
      @import url(../../styles/styles.css)
      @import url(page.css);
      -->
    </style>

  </head>

  <body>

  </body>

</html>
```

Cette méthode est intéressante car elle permet de décentraliser la mise en forme de la structure de la page.

Elle appelle effectivement une feuille de style (CSS) externe au fichier HTML.

Méthode 4 – CSS avec la balise LINK

En ajoutant une ligne entre les balises «head » qui va permettre d'indiquer au fichier HTML quel fichier CSS il se doit de charger :

```
<!DOCTYPE html>
<html>

  <head>

    <title> Titre </title>

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

  </head>

  <body>

  </body>

</html>
```

Comme la précédente, cette méthode est intéressante car elle permet également de décentraliser la mise en forme de la structure de la page.

Elle appelle effectivement une feuille de style (CSS) externe au fichier HTML.

C'est LA méthode recommandée et utilisée sur la plupart des sites internet.

ECRIRE DU CODE CSS

Vous l'aurez compris, nous écrivons le code dans le fichier ayant l'extension « .css ».

Dans l'exemple ci-dessous, les fichiers sont placés au même endroit, cependant il arrive parfois que le fichier CSS ne soit pas dans le même dossier que les fichiers HTML.

Dans ce cas, lors de l'appel du fichier CSS dans la balise link du fichier HTML, il faut indiquer le bon chemin («../ »).





Les sélecteurs

FONCTIONNEMENT

La syntaxe des CSS est la suivante:

```
selecteur{ propriete : valeur;}
```

- ✓ **Les sélecteurs** : pourront être de différents types et nous permettront de sélectionner l'élément dont nous voulons modifier l'apparence.
Par exemple, si nous voulons modifier l'apparence des titres de page sur notre site <h2>, nous devons écrire « h2 ».
- ✓ **Les propriétés** : sont les "effets de style" que nous souhaitons appliquer à un ou plusieurs élément(s).
Il y a par exemple la propriété « color » qui permet d'indiquer la couleur du texte.
- ✓ **Les valeurs** : pour chaque propriété nous devons indiquer une valeur.
Par exemple, pour la couleur, il faudra indiquer le nom de la couleur (en anglais) ou le code couleur (en hexadécimale).

DIFFERENTS TYPES DE SELECTEURS

Les sélecteurs vont nous permettre d'appliquer un style à différents éléments.

- Le sélecteur universel est noté : `*`
- Une balise HTML : `p { color:red; }`
- Une classe indépendante : `.texte-rouge { color:red; }`
- Un identificateur : `#texte-rouge { color:red; }`
- Une sous-classe : `p.texte-rouge { color:red; }`
- Des éléments imbriqués (hiérarchie) : `div p { color:red; }`
- Un attribut : `p[class=exemple] { color:red; }`
- Un pseudo-élément : `a:hover { color:red; }`

SELECTEUR DE BALISES

Imaginons que nous souhaitons mettre en rouge tous les titres de page et en bleu tous les textes sur notre site Internet, nous allons devoir sélectionner les balises auxquelles vont s'appliquer ce style.

Exemple CSS

```
h1{ color : red;}  
p{ color : blue}
```

Cela aura pour but de mettre en couleur rouge les titres contenus entre les balises «<h1></h1>» et en bleu tous les textes contenus entre les balises «<p></p>».

Exemple HTML

```
<h1> ce titre est en rouge </h1>  
<p> ce texte est en bleu </p>
```

SELECTEUR DE CLASSE

Une classe permet de cibler un élément HTML en lui donnant un nom, sur l'exemple ci-dessous, nous avons choisi la classe « texte-rouge ».

Exemple HTML

```
<p> Paragraphe n°1 </p>
<p class = "texte-rouge"> Paragraphe n°2 </p>
<p> Paragraphe n°3 </p>
```

Exemple CSS

```
.texte-rouge
{
    color : red;
}
```

La classe définie dans le HTML peut être appelée via la feuille de style CSS afin de cibler un seul des trois paragraphes.

Sans cette classe, nous aurions été obligés d'appeler la balise « p » et cela aurait eu pour effet de styliser les trois paragraphes en couleur rouge, ce qui n'était pas notre objectif.

Une même classe peut être utilisée à plusieurs endroits dans le fichier HTML.

SELECTEUR D'IDENTIFICATEUR

Un identificateur nous permet également de cibler et d'isoler un élément HTML en lui donnant un nom.

La différence entre un id et une classe réside dans le fait que ce nom doit être unique et ne pas être utilisé 2 fois sur la même page (*règle et convention w3c*), il doit également être privilégié pour des zones importantes.

Exemple HTML

```
<div id = "conteneur" > </div>
```

Exemple CSS

```
#texte-rouge  
{  
    border : 1px solid;  
}
```

SELECTEUR ET REGROUPEMENT

Nous pouvons regrouper plusieurs sélecteurs dans une liste, séparés par des virgules, quand ceux-ci partagent la même mise en forme.

Exemple CSS

```
h1{ font-family: verdana;}  
h2{ font-family: verdana;}  
h3{ font-family: verdana;}
```

Ceci est équivalent à :

```
h1,h2,h3  
{  
    font-family: verdana;  
}
```

Dans cet exemple, nous condenseons trois règles qui ont les mêmes déclarations en une seule.

Cela attribuera la police « verdana » à nos titres : h1, h2, h3.

SELECTEUR D'ATTRIBUTS

Les sélecteurs d'attributs peuvent trouver une correspondance.

Exemple

```
p[class = exemple] { color : blue;}
```

Dans cet exemple, le sélecteur correspond avec tous les éléments « p » dont la valeur de l'attribut « class » est exactement « exemple » :

```
<p class = "exemple"> ce texte sera donc de couleur bleu </p>
```

SELECTEUR PSEUDO-ELEMENT

Nous pouvons sélectionner un élément HTML grâce au child.

Exemple HTML

```
<p> Paragraphe n°1 </p>
<p> Paragraphe n°2 </p>
<p> Paragraphe n°3 </p>
```

Exemple CSS

```
p:first-child
{
    color : red;
}
```

Seul le 1er paragraphe sera coloré en rouge grâce au pseudo-élément first-child (voir aussi nth-child, last-child, etc.).

Ceci permet d'éviter de donner une classe, néanmoins, son utilisation doit rester occasionnelle car si les éléments HTML changent d'ordre, cela continuera de styliser le 1^{er} élément, qui ne sera peut-être plus le même.



Les attributs

MISE EN FORME DU TEXTE ET DES TITRES – POLICE, TAILLE

Propriété	Valeur	Description
font-family @font-face	font-family: Verdana; (police)	Permet de modifier la police utilisée sur le site
font-size	font-size : 15px ; (pixels) font-size : 30% ; (pourcentage) font-size : 20em ; (taille relative)	Taille du texte
font-weight	font-weight : bold; (gras) font-weight : normal; (pas gras)	Epaisseur du texte
font-style	font-style : italic ; (italique) font-style : normal ; (normal)	Italique
text-decoration	text-decoration : underline ; (souligné) text-decoration : overline ; (ligne au-dessus) text-decoration : line-through ; (barré) text-decoration : none ; (normal)	Soulignement et diverses mises en forme
font-variant	font-variant : small-caps ; (petites capitales) font-variant : normal ; (normal)	Petites capitales

MISE EN FORME DU TEXTE ET DES TITRES – POLICE, TAILLE

Propriété	Valeur	Description
font-variant	font-variant : small-caps ; (petites capitales) font-variant : normal ; (normal)	Petites capitales
text-transform	text-transform : uppercase ; (tout mettre en majuscules) text-transform : lowercase ; (tout mettre en minuscules) text-transform : capitalize ; (début des mots en majuscules) text-transform : none ; (normal)	Minuscule, Majuscule.
word-spacing	word-spacing : 10px ; points (pt), inches (in), centimètres (cm), pixels (px) ou pourcentage (%)	définit l'espace entre les mots.
letter-spacing	Letter-spacing : 5px ;	définit l'espace entre les lettres.
font	font: bold 17px Arial;	Ceci est un raccourci pour les différentes propriétés de police. Cela mettra votre texte en gras, 17 pixels, Arial.
content	- une valeur, entre "", - url(url.gif), une image,	permet de spécifier un contenu à insérer avec les propriétés de feuille de style css

ALIGNEMENT

Propriété	Valeur	Description
text-align	text-align : left ; (à gauche) text-align : center ; (centré) text-align : right ; (à droite) text-align : justify ; (texte justifié)	Alignement horizontal du texte
vertical-align	vertical-align : top ; (en haut) vertical-align : middle ; (au milieu) vertical-align : bottom ; (en bas)	Alignement vertical. A utiliser uniquement dans des cellules de tableau, ou dans des éléments inline.
line-height	line-height : 30px ; valeur en pixels (px) ou en pourcentage (%)	Hauteur de ligne
text-indent	text-indent : 10px ; valeur en pixels (px) pour définir l'alinéa de vos paragraphes.	Alinéa
white-space	white-space : normal ; le passage à la ligne est automatique white-space : nowrap ; pas de passage à la ligne automatique white-space : pre ; le passage à la ligne se fait tel que le texte a été saisi dans le code source	Césure
text-shadow	text-shadow : 1px 1px 6px;	Ombre du texte : (horizontale, verticale, fondu, couleur)

DIMENSIONS

Propriété	Valeur	Description
width	width : 900px ;	Largeur (par défaut, la largeur dépendra du texte à l'intérieur) Valeur en px, %, ou encore "auto"
min-width	min-width : 900px ;	Largeur minimum - en pixels
max-width	max-width : 900px ;	Largeur maximum - en pixels
height	height : 900px ;	Hauteur (par défaut, la largeur dépendra du texte à l'intérieur) - Valeur en px, %, ou encore "auto"
min-height	min-height : 900px ;	Hauteur minimum - en pixels
max-height	max-height : 900px ;	Hauteur maximum - en pixels

LES TABLEAUX

Propriété	Valeur	Description
border-collapse	border-collapse : collapse ; (les bordures du tableau et des cellules sont mélangées.) border-collapse : separate ; (les bordures du tableau et des cellules sont séparées)	Type de bordure
empty-cells	empty-cells : show ; (les bordures des cellules vides sont affichées). empty-cells : collapse ; (les cellules vides sont masquées)	Cellules vides
caption-side	via la balise <caption> : caption-side : top ; (en haut du tableau) caption-side : bottom ; (en bas du tableau) caption-side : left ; (à gauche du tableau) caption-side : right ; (à droite du tableau)	Position du titre



Le positionnement

LA POSITION ABSOLUE

Le positionnement absolu « retire » totalement du flux le contenu concerné : sa position est déterminée par référence aux limites du conteneur. Celui-ci peut être :

- Une boîte elle-même positionnée (position relative ou absolue) ;
- Le bloc conteneur initial, à défaut de boîte positionnée, c'est-à-dire en pratique le plus souvent la fenêtre du navigateur.

Code HTML :

```
<div id = "zone">

    <p> Bonjour tout le monde </p>
    <p class = "orange"> Hello ! </p>

</div>
```

Code CSS :

```
p{ margin : 0;}

.orange{ color #ff9c00; position: absolute; top:50%; left: 0px; }

#zone{ border: 2px solid #336699;}
```


Résultat :

Bonjour tout le monde

Hello!

LA POSITION RELATIVE

Un élément en position relative permet de se placer par rapport aux limites de la zone dans laquelle il se trouve. Sa position est déterminée par référence aux limites du conteneur.

Code HTML :

```
<div id = "zone">

    <p> Bonjour tout le monde </p>
    <p class = "orange"> Hello ! </p>

</div>
```

Code CSS :

```
p{ margin : 0;}

.orange{ color #ff9c00; position: relative; left: 30px; }

#zone{ border: 2px solid #336699;}
```

Résultat :

Bonjour tout le monde

Hello!

LA POSITION FIXE

Comme dans un positionnement absolu, le contenu concerné est retiré totalement du flux. Mais il est cette fois positionné uniquement par rapport aux limites de la zone de visualisation, autrement dit la fenêtre du navigateur. Le défilement de la page n'a aucun effet sur un contenu en position fixe.

Code HTML :

```
<div id = "menu"> Menu </div>
```

```
<div class = "contenu">
```

Elle provoque la « fermeture du curseur d'analyse des résultats ». Cela signifie, en d'autres termes plus humains, que vous devez effectuer cet appel à `closeCursor()` chaque fois que vous avez fini de traiter le retour d'une requête, afin d'éviter d'avoir des problèmes à la requête suivante. Cela veut dire qu'on a terminé le travail sur la requête.

```
</div>
```

Code CSS :

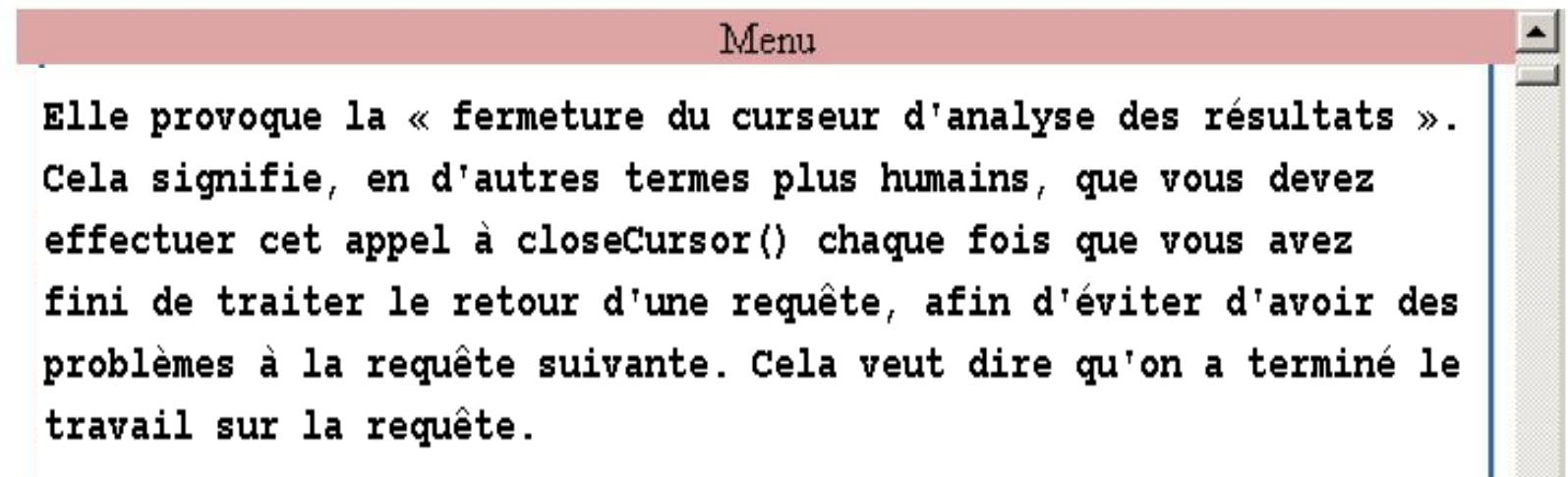
```
.menufixe
{
    background: #dea5a5;
    position: fixed;
    text-align: center;
    width: 100%;
    top: 0;
    left: 0;
}
```

```
.contenu
{
    border: 2px solid #336699;
    margin: 30px 0 0 0;
}
```

Résultat :

Pendant que nous descendons la page à l'aide de l'ascenseur, la zone du menu ne bouge pas.

Le positionnement CSS donne donc toute liberté à la mise en page.



LES FLOTTANTS

Une boîte flottante est retirée du flux normal, et placée le plus à droite (float: right) ou le plus à gauche (float: left) possible dans son conteneur.

Cela permet de faire plusieurs colonnes sur une page web.

Le contenu suivant cette boîte flottante se place le long de celle-ci, dans l'espace laissé libre.

Exemple n°1

Code HTML :

```
<p class = "orange"> Une boite orange </p>
<p class = "bleu"> Une boite bleue </p>
```

Code CSS :

```
.orange
{
    background-color: #ffc600;
    float: right;
    width: 100%;
    margin: 0;
}
.bleu
{
    background-color: #00b4ff;
    height: 50px;
}
```

Résultat :

Une boîte bleu

Une boîte
orange flottant