CSS : Cascade Style Sheet / Feuilles de style en cascade

Principes

Simplification des pages HTML

Dans une construction modulaire et évolutive des pages Web, on dissocie le contenu d'une page Web (texte, structure [tableaux, listes, etc]) de sa mise en forme (couleurs, fond, souligné, italique, position dans la pages, etc) .

Cela permet :

- de disposer de pages plus compactes
- de factoriser la mise en page dans un fichier séparé, réutilisable pour l'ensemble des pages
- de faciliter l'évolution de la mise en forme
- de diminuer le transfert d'informations entre le serveur et l'utilisateur (la mise en forme n'est transmise qu'une fois)
- de pouvoir créer des mises en pages différentes selon le support de destination (ordinateur, tablette, téléphone).

Bonnes pratiques

Les normes du W3C recommandent de ne jamais faire apparaître de mise en forme dans le code HTML. Ainsi, les attributs *align*, *bgcolor*, et autres *width* devraient systématiquement être renvoyées dans une feuille de style.

Balises et Styles personnalisés

Les feuilles de style peuvent redéfinir les balises standard en affectant la valeur que l'on souhaite pour chacune des options. On utilise ce principe lorsque l'on veut par exemple que tous les paragraphes aient la même mise en page (balise <*p*>), ou tous les formulaires (balise <*form*>).

On peut aussi créer des styles personnalisés en ajoutant de nouveaux intitulés auxquels on fera appel par un attribut *class* ou *id* à l'intérieur d'une balise standard. On utilise ce choix lorsqu'on souhaite que la mise en forme s'applique à différents objets d'une page web (tableau, paragraphe, zone de saisie, etc).

Fonctionnement

Application des styles

Les styles peuvent être définis à de multiples endroits, depuis l'intérieur d'une balise jusqu'à un fichier de style indépendant. Il est conseillé de tout centraliser dans une ou plusieurs feuilles de style, mais on peut à l'occasion faire un écart à ce principe.

Les éditeurs de page ou de site ont tendance à surcharger le code HTML d'options de style directement à l'intérieur des balises, perdant tout l'intérêt de factorisation des styles.

Style dans une balise

Dans toute balise HTML, on peut avoir recours à la notion de style. Ce n'est pas la solution la plus pertinente et cela ne respecte pas les recommandations du W3C.

Occasionnellement, pour une mise en forme complexe que les seules options de la balise (par exemple pour un tableau ou dans une division DIV) ne permettent pas, cette solution peut éviter d'avoir à créer un style pour une utilisation ponctuelle. Cela permet aussi de limiter l'utilisation de multiples balises (pour la police, l'alignement d'un paragraphe et la couleur d'un encadrement par exemple).

<HTML><table style="align: center; color: white; font-family:arial ; border-color:green ;"></HTML>

Style dans une page HTML

On peut définir les styles à utiliser dans une page au sein même du code HTML de cette page.

Si l'on veut appliquer une mise en forme particulière à une page au sein d'un site, cela peut être une solution adaptée. Pour cela, on utilisera la balise STYLE :

<HTML><STYLE>liste_des_styles_définis_ou_redéfinis</STYLE>.</HTML>

Feuille de style

Une feuille de style est un fichier ayant l'extension CSS et ne contenant que des descriptions de style (il n'y a pas de HTML).

Une page HTML fera appel à une feuille de style par la balise suivante dans l'entête (<HEAD>)

<HTML><LINK REL="stylesheet" HREF="feuille_de_style" TYPE="text/css"></HTML>

Hiérarchie

Dans la mesure où les valeurs des options de mise en forme peuvent être paramétrées à plusieurs endroits (dans une feuille de style, dans les styles d'une pages HTML, dans les divisions de texte (balise DIV) et dans les balises directement), le navigateur appliquera la première valeur rencontrée dans l'ordre suivant :

- valeur de la propriété dans la balise
- valeur de la propriété dans la balise DIV qui encadre une autre balise
- valeur dans le style de la page HTML elle-même
- valeur dans le style de la feuille de style appelée par le fichier HTML
- valeur dans le style de la feuille de style associée au navigateur.

Syntaxe

Paramétrage des feuilles de style

Les styles peuvent être définis dans un fichier portant par convention l'extension .css ou entre les balises <STYLE> </STYLE>. Un style s'écrit :

```
   ObjetDePresentation {
        optionMiseenForme1: valeur1 ;
        [optionMiseenForme2: valeur2 ;…]
        }
```

Les sélecteurs

Une feuille de style décrit les comportements d'affichage pour les objets de présentation (ou sélecteurs) suivants :

- Les balises standard : on indique le nom de la balise

```
body {                          /* redéfinition de la balise body */
      color: lightgreen ;               /* couleur du texte vert clair */
      background: rgb(130,124,237)  ;   /* couleur du fond mauve */
      }
```

- Une *classe de style* : c'est un sélecteur qui décrit un style universel, applicable à toute balise et plusieurs fois dans une page. Il est préfixé par un « . » et est appelé par l'option *class="nomClasse"* à l'intérieur de la balise qui l'utilise.

```
 .encadreRouge {                 /* création d'une classe encadreRouge */
     border:2px solid red ;      /* trait de 2 pixel, solide (continu) de
couleur rouge */
     }
     …
    //utilisation de la classe dans le HTML
    <div class="encadreRouge">Important</div>
```

- Un *identifiant de style* : c'est un sélecteur unique, applicable à toute balise. Il est préfixé par un « # » et est appelé par l'option *id="nomID"* à l'intérieur de la balise qui l'utilise.

Remarque : l'*id* servant à nommer les objets (de formulaire ou autre), il ne doit y avoir normalement qu'un seul appel à un *id* dans une page HTML. On ne devrait donc l'utiliser que pour décrire des éléments uniques utilisant telle mise en forme (un logo, un menu, etc)

```
#menu {                                 /* création d'une mise en forme pour
l'ID nommé menu */
       color:beige ;                    /* couleur du texte */
       border:2px solid green ;         /* trait de 2 pixel, solide (continu)
de couleur vert */
       background: rgb(100,100,137)  /* couleur du fond */
```

```
     }
     …
     /* utilisation de l'id dans le HTML (qui ne devrait être qu'une fois
dans la page)  */
     <table id="menu >tableau décrivant un menu…</table >
```

Combinaison et hiérarchisation

On peut simplifier l'écriture du code HTML en définissant le mode d'application des styles dans une page. En effet, les classes et ID s'appliquant à toute balise, la mise en forme est universelle. Cependant, on pourrait vouloir que certains éléments de mise en forme ne s'appliquent qu'à une balise particulière, ou que le même nom (de classe) réalise quelques nuances selon qu'il s'applique à telle ou telle balise.

```
div.menu {                          /* création d'une classe menu pour la
balise DIV */
     background: rgb(100,100,137)   /* couleur du fond */
     }
table.menu {                        /* création d'une classe menu pour la
balise table */
     background: rgb(100,100,137)   /* couleur du fond */
     border-color : rgb(200,100,100)
     }
```

En outre, on peut aussi définir comment se comportent les balises lorsqu'elles sont encadrées dans d'autres.

```
div.menu  a {        /* création d'un style pour un lien (balise a) à
l'intérieur d'un div */
                     /* appliquant la classe menu */
                color: lightgreen        /* couleur du texte */
     }
```

Principales options de style pour la mise en forme

Police de caractère

Type	Propriété	Valeur	Description
Police	font-family	Nom de police	Définit un nom de police ou une famille
	font-style	normal, italic, oblic	Définit le style d'écriture
	font-weight	normal, bold, light, bolder, lighter, valeur numérique	Définit l'épaisseur de la police
	font-size	Valeur numérique	Définit la taille de la police en points (pt), cm, % ou pixels (px)
	font-variant	small-caps ou normal	Définit le mode d'affichage (petites majuscules)
	color	#RRGGB ou nom de couleur ou rgb(valRouge, valVert, valBleu)	Définit la couleur de la police
	font	{font: bold italic}	Permet de valoriser toutes les options en une seule fois

Paragraphes et habillage du texte

Type	Propriété	Valeur	Description
Style	*text-align*	*left*, *center*, *right*, *justify*	Définit l'alignement horizontal du texte
	text-decoration	*blink* (clignotant), *underline* (souligné), *line-through* (barré), *overline* (surligné)	Définit une décoration pour la police
	text-transform	*uppercase* (majuscule), *lowercase* (minuscule) , *capitalize* (première lettre en majuscule)	Définit la casse du texte
	text-indent	Valeur numérique	Permet une tabulation de n caractères
	vertical-align	*baseline* (tout en bas), *middle* (milieu), *sub* (en-dessous), *super* (au-dessus) *text-top* (sous le texte) ou *text-bottom* (au-dessus du texte)	Définit l'alignement vertical d'un élément par rapport à son parent (image par rapport à du texte, texte dans un tableau, etc).

Arrière plan (pour le corps du texte, les tableaux)

Type	Propriété	Valeur	Description
Arrière-plans	*background-color*	"#RRGGBB" ou nom couleur ou rgb(valRouge, valVert, valBleu)	Définit la couleur du fond
	background-image	URL(nom de l'image)	Définit l'image d'arrière-plan
	background	"#RRGGBB" ou nom couleur ou rgb(valRouge, valVert, valBleu)	Permet d'affecter l'ensemble des propriétés en une seule option
	background-repeat	*repeat* (mosaïque), *repeat-x* (répétition horizontale), *repeat-y* (répétition verticale), no-repeat (pas de répétition)	Indique le mode de répétition de l'image d'arrière plan
	background-attachement	*fixed* ou *scroll*	Détermine si l'image d'arrière plan se déplace avec le texte ou non, permet de créer un filigrane

Marges

Type	Propriété	Valeur	Description
Marges	*margin-top*	En points (*pt*), pouces (*in*), en cm (*cm*), en pixels (*px*), ou en % (*%*)	Valeur de la marge au dessus du texte
	margin-right	idem	Valeur de la marge droite
	margin-left	idem	Valeur de la marge gauche
	margin-bottom	idem	Valeur de la marge en dessous du texte
	margin	idem	Définit les valeurs *top right left bottom* en une seule ligne
Espacements	padding	En points (*pt*), pouces (*in*), en cm (*cm*), en pixels (*px*), ou en % (*%*)	Espacement entre le texte et le conteneur (tableau, division ou autre)
	Top, left, bottom et right	En points (*pt*), pouces (*in*), en cm (*cm*), en pixels (*px*), ou en % (*%*)	Espacement entre le texte et le bord supérieur, ou gauche, ou inférieur ou droit

Bordures

Type	Propriété	Valeur	Description
Bordure	*border-width*	En points (*pt*), pouces (*in*), en cm (*cm*), en pixels (*px*), ou en % (*%*)	Définit les propriétés d'épaisseur de trait
	border-color	"#RRGGBB" ou *rgb(*valRouge, valVert, valBleu*)*	Couleur de la bordure
	border-style	*dotted* (pointillé), *dashed* (tiret), *solid* (trait plein), *double* (trait double), *groove* (effet de relief léger)	Définit le style de bordure
	border-top border-bottom border-left border-right)	épaisseur style couleur	Définit les caractéristique de bordure en une seule commande pour le dessus (ou bas ou gauche ou droit)

Style et liste (puce)

Type	Propriété	Valeur	Description
Listes à puces	*list-style-type*	*disc*, *circle* ou *square*	Type de puces et de numérotation
	list-style-image	URL(nom de l'image)	Permet de remplacer les puces par une image
	list-styleposition	*inside* ou *outside*	Spécifie si les puces sont à l'intérieur ou à l'extérieur du texte

Pseudo classes

Il est aussi possible de définir la mise en forme en fonction des actions menées (avec la souris par exemple), notamment lorsqu'un objet (zone, lien, image, etc) est survolé par la souris, ou qu'un lien a été visité, ou en fonction de la position d'une balise par rapport à une autre.

On parle d'une *pseudo classe*, qui est systématiquement préfixée par « *:* » et s'applique à un sélecteur (balise, classe ou id ou combinaison) sous la forme *selecteur:pseudo_classe*. Ces éléments

ne sont pas supportés par tous les navigateurs (voir http://fr.selfhtml.org/css/proprietes/pseudoformats.htm: page qui n'existe plus mais dont les sources en français sont accessibles ici http://fr.selfhtml.org/)

Type	Pseudo-classe	Description
Liens	:link	Définit les mises en forme sur un lien non visité. S'applique à la balise *<a>* // a:link{…}//
	:visited	Idem pour un lien non visité // a:visited{…}//
Souris	:hover	Définit les mises en forme lorsque la souris est amenée sur une zone. Doit venir après les pseudo classes de visite pour être actif sur un lien. S'applique à toute balise // div:hover{…}//
	:active	Idem lorsque la souris reste cliquée sur un élément (doit venir après hover).
	:focus	Idem lorsque le curseur de saisie ou une tabulation arrive sur un élément
Position	:first-line	Indique le comportement pour la première ligne à l'intérieur d'un bloc // div:first-line{…}//
	:first-letter	Idem pour le premier caractère, ce qui permet de faire des lettrines // p:first-letter{…}//
	:first-child	Indique le comportement de la première apparition d'une balise (ici p) à l'intérieur d'une autre (ici div) : // div p:first-child//
Encadrement	:before	Indique le texte à insérer avant un sélecteur : *.important:before {font-weight:bolder ; color :red ; content:"!";}*
	:after	Indique le texte à insérer après un sélecteur : *.prix:after {content:" €";}*

Organisation de l'espace avec les styles et les balises DIV

Evolution

Pour répartir l'information au sein d'une page Web, on a pendant une période utilisé la notion de cadres (*frames*) qui permettait, sur des connexions au débit faible, de limiter les chargements de parties fixes.

Avec l'avènement des liaisons *DSL*, le problème de la quantité d'informations à acheminer est devenu moins critique, et l'on a préféré revenir au principe d'origine : toutes les informations d'une page (l'entête, le menu de navigation, le contenu informatif) sont contenues directement dans cette page.

Il a alors fallu penser à des techniques d'organisation de l'espace :

- faire apparaître une zone fixe en haut,
- répartir les zones à droite, au milieu, à gauche,
- encadrer une image par du texte,
- etc.

La première idée qui vient (et qui reste très actuelle et précise) est le recours à des tableaux dont les bordures sont masquées.

Une seconde évolution, plus incertaine (les navigateurs ne réagissent pas tous exactement de la même façon) est le recours à des définissions de classes de style et à leur application par le balises *<DIV>* (division). Ces balises ont pour but d'organiser des divisions de texte en leur appliquant des mises en forme particulières, portant aussi bien sur les caractères que les paragraphes, les couleurs et autres dispositions.

Principe d'utilisation

Pour organiser l'espace, il faut penser les différentes zones d'un écran comme autant de cadres flottants ou de flux d'informations qui se coulent dans l'espace laissé disponible par les autres portions de texte.

Par exemple, sur copie d'écran ci-contre, du site de l'académie de Caen, on peut imaginer que les zones encadrées en rouge sont autant d'espaces indépendants emboîtés les uns dans les autres.

Le principe des divisions est donc de définir comment elles se comportent les unes par rapport aux autres en terme de superposition, d'alignement et de retour à la ligne, et comment elles se positionnent par rapport à l'écran en terme d'alignement (gauche ou droite).

Par défaut, une division occupe une portion de l'espace, et tout le reste du texte ou les autres divisions se collent contre ce cadre dans l'espace disponible de l'écran.

Propriétés //float// et //clear//

Le paramétrage de ces positionnements se fait principalement par les propriétés *float* et *clear*.

- *Float* permet d'indiquer la position d'une division par rapport à l'écran. Les valeurs possibles sont *left* et *right*, indiquant si la division est alignée à gauche ou à droite de l'écran (ou collée à la division de gauche ou de droite qui la précède).
- *Clear* définit comment une division se comporte par rapport à celles qui existent précédemment. Les valeurs possibles sont :
  - *left* (on ne se coule pas contre la division existant à gauche, et l'on retourne à la ligne),
  - *right* (on ne se coule pas par rapport à la division de droite)
  - *both* (on ne se coule contre aucune des divisions existante)

On trouvera une explication illustrée de ces éléments à l'adresse suivante : http://www.journaldunet.com/developpeur/tutoriel/css/061130-css-comprendre-float-clear.shtml

Une autre explication plus détaillée est lisible ici : http://openweb.eu.org/articles/initiation_flux/

Propriétés des divisions

Pour définir exactement l'espace occupé par une division, on utilisera les propriétés de marge (voir plus haut) ainsi que la hauteur (*height*), la largeur (*width*). On pourra aussi jouer sur la couleur d'arrière plan, sur les bordures, etc. Mise en œuvre

La mise en œuvre de ces divisions se fait en deux étapes. Dans un premier temps, on définit les propriétés d'un style correspondant à chacune des zones (dans la feuille de style), et dans un second temps on applique le style à une balise <*DIV*> grâce à l'option <*... class="nom_du_style"*>.

Illustration

On peut imaginer que la page du site de l'académie de Caen avait, dans sa feuille de style les éléments suivants correspondant au cadre de tout en haut, au cadre de tout à gauche et celui de tout à droite :

```
.zoneHaut {
       background-image: URL("imageAca.gif") ;
       float : left ;
       width : 80% ;
     }

.zoneGauche {
       background-color : green ;
       float : left ;    /*collé à ce qu'il y a à gauche*/
       clear : left ;    /*retour à la ligne par rapport à ce qu'il y a à
gauche*/
     }

.zoneDroite {
        background-color : green ;
        float : right ;        /* est aligné sur le bord de droite */
        width : 20% ;           /*occupe 20% de l'espace à droite*/
     }
```

La page HTML qui constitue l'accueil serait alors composée, notamment, des descriptions suivantes (en faisant appel à la feuille de style ci-dessus) :

<HTML>

```
   …
   <div class="zoneHaut"></div>
   <div class="zoneDroite>…texte_ministère…</div>
   <div class="zoneGauche">….texte_des_menus…</div>
```

</HTML>

Remarque : si on avait choisi *clear :both* ; pour le style *zoneGauche*, la partie de gauche correspondant au menu se serait alors retrouvée calée à gauche en dessous de la dernière ligne de la zone de droite (logo du ministère). Complexité

La complexité de l'utilisation de ces éléments grandit à mesure que les zones s'imbriquent et se superposent. Il est même possible d'aller jusqu'à des chevauchements.

En outre, une balise <*DIV*> ne peut appliquer qu'une seule *classe de style*. On peut alors être amené à imbriquer des <*DIV*> faisant appel à des classes de style complémentaires.

Il s’agit d’un véritable travail de designer Web, qui nécessite une grande rigueur, une parfaite connaissance de l’organisation des sites.

En outre, les balises et les styles ne s’appliquent pas toujours parfaitement selon les différents navigateurs.

DIV vs Tableaux

L’utilisation des tableaux reste encore une solution de secours très pratique, qui permet d’avoir une vision directe de ce que l’on souhaite obtenir dans l’interface d’un éditeur de page.

L’inconvénient de ces derniers est de s’adapter à la largeur des écrans 4/3 et 16/9. On jouera alors sur des pourcentages pour définir la largeur du tableau et de ses différentes colonnes.

Pour remarque, la page de l’académie de Caen est constituée d’un mélange de tableaux et de DIV.

From:
https://wiki.sio.bts/ - **WIKI SIO : DEPUIS 2017**

Permanent link:
https://wiki.sio.bts/doku.php?id=css

Last update: **2020/07/26 16:27**