

MES RECHERCHES SUR LE CSS

1- Utilités du CSS

Le **CSS** est un langage informatique utilisé pour mettre en forme les fichiers HTML sur un site web. Ainsi, les fichiers **CSS** permettent de gérer l'aspect graphique d'une page en HTML.

L'**utilisation de CSS structure votre site internet**. En effet, cela permet une plus grande cohérence de votre conception ainsi qu'une maintenance de ce dernier plus facile. Les feuilles de style CSS existent depuis de nombreuses années. Il est devenu indispensable de savoir séparer la structure HTML de votre site des feuilles de style CSS.

Les avantages de la mise en place de feuilles de style CSS

- **Plus facile d'entretenir votre site** internet et de le mettre à jour.
- Une plus **grande cohérence** lors de la conception de votre site.
- Le code sera **plus léger**.
- Le temps de téléchargement de **votre site sera plus rapide**.
- **Avantages lors de l'optimisation** pour les moteurs de recherches.

1- Définition et syntaxes

Les feuilles de styles en cascade permettent de mettre en forme des documents web. Le terme cascade signifie que la mise en forme d'une page peut faire appel à plusieurs feuilles de styles.

Une règle CSS est composée de 2 éléments : le sélecteur et la déclaration. Le sélecteur indique l'élément sur lequel vont s'appliquer les déclaration CSS.

La déclaration les propriétés et les valeurs à appliquer à cet élément. Exemple de syntaxe CSS :

```
#style { /*sélecteur*/
```

```
background-color: pink } /*propriétés : valeur*/
```

2- Intégration du CSS à un projet Web

Il existe plusieurs méthodes pour intégrer le CSS à un projet Web

- Une première méthode consiste à **intégrer le [code CSS](#) directement dans votre balise HTML** en utilisant l'attribut style. Cette technique est à éviter puisqu'elle ne permet pas de dissocier le contenu de la mise en forme. La mise à jour des pages devient alors difficile et cela supprime la clarté du code dans votre document.
Cette solution existe mais on la réserve à une utilisation ponctuelle, pour par exemple définir un style sur un seul élément n'apparaissant que sur une seule page de votre site.
- Il est préférable d'identifier les balises concernées à l'aide de noms afin de leur appliquer des règles distinctes. Il est possible d'écrire du code CSS dans l'entête d'un document HTML, dans la balise <Head>. Il suffit pour cela **de l'encadrer par une balise <style>** ayant pour attribut type="text/css".
Cette méthode permet de retirer la mise en forme du corps du document, mais elle ne s'appliquera qu'à la page concernée et non à l'ensemble du site.
- La dernière méthode consiste à rédiger le code CSS dans un document externe afin qu'il puisse être appelé par les différentes pages de notre site. La séparation du fond et de la forme permet de retrouver et de corriger plus rapidement le texte des pages, de facilement mettre à jour le design et

rendre vos document HTML plus léger.

Pour déclarer une feuille de styles externe, il suffit de l'appeler dans le <head> de votre document HTML.

3- Les sélecteurs CSS

- **Le sélecteur de balise** : il permet tout simplement d'appliquer du style à toutes les balises concernées de la page. Il suffit de déclarer la définition comme ceci :

baliseHTMLconcernée {propriété: valeur; }

- **Le sélecteur de classe** : Imaginons maintenant que nous voulions changer la taille de plusieurs éléments dans la page, mais sans les changer toutes. Nous utiliserons alors un sélecteur de type class qui sera attribué uniquement aux éléments concernés. Le sélecteur class est représenté par un . suivi de son nom.

Prenons l'exemple d'images de chien et chat se voyant affectées

des dimensions différentes : **En CSS** : **.chien { width:120px; height:80px; }**

.chat { width:120px; height:120px; }

Intégration HTML <body>

** ... l'image aura pour dimensions 120px x 80px**

** ... l'image aura pour dimensions 120px x 120px**

<div class="auto">... la balise aura pour dimensions 120px x 80px

- **Le sélecteur d'identifiant** : Si vous souhaitez sélectionner un unique élément de la page, il faut donner à la balise HTML un identifiant. Celui-ci se déclare en HTML de cette manière :

#identifiant { ... }

Prenons exemple avec un élément <h2> auquel vous souhaitez appliquer un style propre :

En CSS :

```
#texterouge { color: red ; }
```

Intégration HTML

```
... <h2>titre affiché normalement </h2>
```

```
<h2 id="texterouge">titre affiché en rouge </h2>
```

- **Les sélecteurs d'attributs** : Cette méthode permet de sélectionner une balise ayant un attribut donné. Plutôt que de différencier chaque balise par un id spécifique, vous pouvez les sélectionner grâce à leur attribut.

Modifions, pour l'exemple, le style des balises de type texte et de type bouton très utilisées dans les formulaires :

```
input [type="text"] {  
border: solid 2px #919191; }  
input [type="submit"] {  
border: solid 1px #919191;  
font: black 10px arial; }
```

- **Le sélecteur universel** : Ce sélecteur est représenté par une étoile et permet d'appliquer une règle de style à l'ensemble des balises du document.

Si, par exemple, vous souhaitez ôter toutes les marges par défaut de toutes les balises, vous saisissez la règle suivante :

```
* { margin: 0;  
padding:0; }
```

4- Les commentaires

il est possible d'insérer des commentaires dans un code CSS, comme vous le feriez pour votre document HTML, afin de vous y retrouver plus facilement lorsqu'il faudra y apporter des modifications.

Les commentaires se placent entre les signes /* et */, vous verrez alors le texte saisi en grisé et ne sera pas pris en compte dans la lecture des règles CSS.

5-Priorité et Héritage en CSS

L'héritage des CSS est fondé sur le modèle Parent-Enfant(s) : **chaque élément Enfant reçoit en héritage tous les styles de son élément Parent**. Par exemple, la balise <html> est parent de <body>, et <table> est parent de <tr> qui lui-même est parent de <td>. Cet héritage est très pratique et permet d'éviter beaucoup de répétitions inutiles : en effet, en attribuant une propriété à un parent (par exemple *font-size: 1.5em*), elle sera transmise à tous ses enfants, mais aussi aux enfants de ces enfants, etc...

L'élément enfant héritera de toutes les propriétés de l'élément parent uniquement si ces propriétés s'héritent, car l'héritage ne fonctionne pas non plus sur toutes les propriétés css (margin, padding et autres propriétés de bloc).

6-Les couleurs

La couleur fait partie intégrante des moyens d'expressions. Lorsqu'on écrit un site web, il est naturel d'y ajouter des couleurs dans la mise en forme. Avec [CSS](#), il existe de nombreuses façons d'ajouter de la couleur aux [éléments HTML](#) afin d'obtenir le résultat souhaité.

De façon générale, la propriété [color](#) permet de définir la couleur de premier plan pour le contenu d'un élément HTML et la propriété [background-color](#) permet de définir la couleur utilisée pour l'arrière-plan de l'élément. Ces propriétés peuvent être utilisées sur la quasi-totalité des éléments HTML.

7-La mise en forme

- **La taille**

Pour modifier la taille du texte, on utilise la propriété CSS `font-size` . Mais comment indiquer la taille du texte ? C'est là que les choses se corsent, car plusieurs techniques vous sont proposées :

- indiquer une **taille absolue** : en pixels, en centimètres ou millimètres. Cette méthode est très précise, mais il est conseillé de ne l'utiliser que si c'est absolument nécessaire, car on risque d'indiquer une taille trop petite pour certains lecteurs ;
- indiquer une **taille relative** : en pourcentage, « em » ou « ex », cette technique a l'avantage d'être plus souple. Elle s'adapte plus facilement aux préférences de taille des visiteurs.

- **La police**

Pour qu'une police s'affiche correctement, il faut que tous les internautes l'aient. Si un internaute n'a pas la même police que vous, son navigateur prendra une police par défaut (une police standard) qui n'aura peut-être rien à voir avec ce à quoi vous vous attendiez.

Il est possible de faire télécharger automatiquement une police au navigateur.

La propriété CSS qui permet d'indiquer la police à utiliser est `font-family` . Vous devez écrire le nom de la police comme ceci :

balise

{

font-family: police;

}

Seulement, pour éviter les problèmes si l'internaute n'a pas la même police que vous, on précise en général *plusieurs* noms de police, séparés par des virgules :

balise

{

```
font-family: police1, police2, police3, police4;  
}
```

Le navigateur essaiera d'abord d'utiliser la police1 . S'il ne l'a pas, il essaiera la police2 . S'il ne l'a pas, il passera à la police3 , et ainsi de suite.

En général, on indique en tout dernier serif , ce qui correspond à une police par défaut (qui ne s'applique que si aucune autre police n'a été trouvée).

Voici une liste de polices qui fonctionnent bien sur la plupart des navigateurs :

- Arial ;
- Arial Black ;
- Comic Sans MS ;
- Courier New ;
- Georgia ;
- Impact ;
- Times New Roman ;
- Trebuchet MS ;
- Verdana.

• **L'alignement**

Le langage CSS nous permet de faire tous les alignements connus : à gauche, centré, à droite et justifié.

C'est tout simple. On utilise la propriété text-align et on indique l'alignement désiré :

- left : le texte sera aligné à gauche (c'est le réglage par défaut) ;
- center : le texte sera centré ;
- right : le texte sera aligné à droite ;
- justify : le texte sera « justifié ». Justifier le texte permet de faire en sorte qu'il prenne toute la largeur possible sans laisser d'espace blanc à la fin des lignes. Les textes des journaux, par exemple, sont toujours justifiés.

- **Les bordures**

Pour border, on peut utiliser jusqu'à trois valeurs pour modifier l'apparence de la bordure :

- **La largeur** : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme 2px) ;
- **La couleur** : c'est la couleur de votre bordure. Utilisez, comme on l'a appris, soit un nom de couleur (black , red ...), soit une valeur hexadécimale (#FF0000), soit une valeur RGB (rgb(198, 212, 37)) ;
- **Le type de bordure** : là, vous avez le choix. Votre bordure peut être un simple trait, ou des pointillés, ou encore des tirets, etc. Voici les différentes valeurs disponibles :
 - none : pas de bordure (par défaut) ;
 - solid : un trait simple ;
 - dotted : pointillés ;
 - dashed : tirets ;
 - double : bordure double ;
 - groove : en relief ;
 - ridge : autre effet relief ;
 - inset : effet 3D global enfoncé ;
 - outset : effet 3D global surélevé.

- **Les Ombres**

Les ombres font partie des nouveautés récentes proposées par CSS3. Aujourd'hui, il suffit d'une seule ligne de CSS pour ajouter des ombres dans une page !

Nous allons ici découvrir deux types d'ombres :

- Les ombres des boîtes : box-shadow
- Les ombres du texte: text-shadow

8-Les images d'arrière plan

La propriété permettant d'indiquer une image de fond est `background-image` . Comme valeur, on doit renseigner `url("nom_de_l_image.png")` . Par exemple :
body

```
{  
  background-image: url("neige.png");  
}
```

9- Les types d’affichages

Par défaut, les éléments HTML sont affichés selon l'un des modes suivants :

1- Bloc / block



Occupe toute la largeur disponible. Lorsque 2 éléments blocs se suivent dans une page, ils sont positionnés (par défaut) l'un **sous** l'autre.

Exemple typique d'élément bloc : l'élément <p> (le paragraphe).

2- En-ligne / inline



N'occupe que la largeur indispensable à l'affichage du contenu et ne provoque pas de retour à la ligne. Lorsque 2 éléments en-ligne se suivent dans une page, ils sont positionnés l'un **à côté** l'autre (si la largeur de page le permet).

Exemple typique d'élément en-ligne : l'élément (image).

3- Invisible / not displayed

✂ Certains éléments ne servent qu'à apporter des informations invisibles pour l'internaute.

Exemple typique d'élément non affiché : l'élément <meta>.

10- Le positionnement

- **Le float** : La position float retire une boîte du flux normal pour la placer le plus à droite ou le plus gauche possible dans son conteneur.
- **Le flex-box** : Flexbox permet d’appréhender les feuilles de style (CSS) plus aisément et notamment le positionnement des blocs (alignement vertical – horizontal – ordre – espacement – remplissage – etc). Afin de mieux comprendre ce qu’est Flexbox, rien de plus simple que la pratique :).

Dans ce billet, nous allons procéder à une “battle” : Alexandre Démoulin VS Georges Damien : le premier fera du flexbox et le second du CSS “classique”. Imaginons donc une maquette cible pour notre futur site internet décrite de la sorte :

- Un header
- Un footer
- Un menu à gauche
- Un contenu à droite qui se redimensionne automatiquement.