**Les bases sur CSS**

**CSS :**

* Cascading Style Sheets, c’est un langage qui vient compléter le html.
* Un langage dédié à la mise en forme du contenu d’une page Web.
* Permet de mettre en forme la page (couleurs, tailles, police, bordure…).
* Éditeur de texte : Sublime Text, Notepad++.
* Pour créer des styles CSS, je peux utiliser un éditeur de texte quelconque.
* Avec les CSS, je peux modifier la mise en forme des balises HTML et les styles définis par le concepteur.
* Dernière version : CSS3.
* Entre /\* et \*/ = commentaires.

**OU L'ÉCRIT-ON :**

Soit dans :

* Un fichier **.css** (recommande)(meilleures façons pour ne pas se perdre).
* L’en-tête **<head>** du fichier **.html**.
* Dans les balises via l’attribut **style** (moins recommandé).

**DANS UN FICHIER .CSS :**

* Dans un fichier **.html** c’est **<link rel=’stylesheet’ href=’style.css’ />** qui indique que le fichier HTML est associé à un fichier nommé **style.css** qui est chargé de la mise en forme.
* Il faut avoir un fichier .html et bien sur un fichier .css placer tous deux dans un même dossier. Le fichier .html fait automatiquement appel au fichier .css (attention à la nomination du dossier, pour le retrouver).

**Exemple :** p

{

Color: blue;

}

* Ce code ci-dessus signifie “je veux que tous mes paragraphes soient écrits en bleu.” Le site web affichera donc les paragraphes en bleu.

**Autre Exemple :** h1

{

color: blue;

}

* Seul le titre de la page web s’affiche en bleu.

**DANS L'EN-TÊTE <HEAD> :**

* Pour utiliser du CSS il faut l’utiliser dans l’en-tête **<head>** avec la balise **<style></style>**, il n’y a besoin que d’un seul fichier **.html**.

**Exemple :** <head>

<meta charset =’utf-8’ />

<**style**>

p

{

color: blue;

}

</**style**>

**DANS LES BALISES :**

* On peut ajouter un attribut **style** à n’importe quelle balise.

**Exemple :** <p **style**=’color: blue;’>Bonjour et Bienvenue ici ! </p>

* Seul ce paragraphe qui contient le code CSS sera coloré en bleu.

**APPLIQUER UN STYLE :**

* **Nom des balises :** En CSS si je veux modifier l’apparence de tous les paragraphes **<p>**, je dois écrire **p**.
* **Les propriétés CSS :** Les effets de style de la page sont rangés dans des propriétés.

**Exemple : color** qui permet d’indiquer la couleur du texte, **font-size** qui permet d’indiquer la taille du texte, etc.

* **Les valeurs :** Pour chaque propriétés une valeur est nécessaire,

**Exemple : color**, il faut indiquer le nom d’une couleur, **font-size**, il faut indiquer la taille que l’on veut.

**Exemple feuille de style CSS :** balise1

{

propriete1: valeur1;

propriete2: valeur2;

propriete3: valeur3;

}

* On écrit le nom de la balise (par exemple **h1**), on ouvre ensuite des accolades pour y mettre les propriétés et les valeurs que l’on souhaite (on peut en mettre autant que l’on veut). Chaque propriété est suivie de **( : )** puis de la valeur qui correspond et se termine ensuite par **( ; )**.

**APPLIQUER UN STYLE A PLUSIEURS BALISES :**

* En CSS, si les deux balises doivent avoir la même présentation (dans notre cas la même couleur), il suffit de combiner la déclaration en séparant les balises par une virgule.
* On peut en indiquer autant que l’on veut à la suite.

**Exemple :** h1, em

{

color: blue;

}

* On peut également l’écrire comme ceci : (moins pratique).

**Exemple :** h1

{

color: blue;

}

em

{

color: blue;

}

* Cela signifie “je veux que le texte de mes **h1** et **em** soit en bleu.

**LES COMMENTAIRES DANS DU CSS :**

* Ils servent à se retrouver dans son fichier CSS (surtout s’il est long), ils ne seront pas affichés sur la page web.
* On peut d’ailleur créer plusieurs fichiers **.css** si celui-ci est trop long, on peut par exemple les faires par sections de notre site
* Pour faire un commentaire il faut taper **/\*** suivi du commentaire et puis **\*/** pour finir le commentaire, les commentaires peuvent être écrits sur une ou plusieurs lignes.

**Exemple :** p

{

color: blue;  **/\* les paragraphes seront en bleu \*/**

}

**APPLIQUER UN STYLE : CLASS ET ID**

* Il y a un problème avec ce qu’on a fait avant, tous les paragraphes vont être affichés en bleu (ils possèdent tous la même présentation)
* Pour résoudre ce problème on peut utiliser les attributs **class** et **id**. C’est attributs sont quasiments identiques

1. L'attribut **class**, on peut le mettre sur n'importe quelle balise, aussi bien titre, images etc.

**Exemple :** <body>

<h1 **class**= “ “> </h1>

<p **class**=” “> </p>

<img **class**=” “> </>

* On doit écrire entre les **“** et **“** : la valeur, donc un nom qui commence par une lettre.

**Exemple :** <body>

<h1>mon site de merde </h1>

<p class= “**introduction**”>Bonjour et bienvenue sur le site!</p>

* Le paragraphe est identifié avec le nom **introduction,** on peut réutiliser le nom dans le fichier CSS pour dire que “je veux que seules les balises qui on le nom **introduction** soient en bleu”.

**Exemple sur le fichier CSS : .introduction**

{

color: blue;

}

(Seul le paragraphe appelé **introduction** va s'afficher en bleu.)

1. L’attribut **id**, ne peut être utilisé qu’une fois dans le code contrairement à l’attribut **class**. On l’utilise souvent sur des éléments uniques comme “le logo”.

**Exemple :** <img src= “images/logo.png” alt= “Logo du site” id= “logo”>

**Exemple sur fichier CSS :** #logo

{ /\* Indiquez les propriétés CSS ici \*/ }

**BALISES UNIVERSELLES :**

* Si on a besoin d'appliquer un attribut **class** ou **id** à un mot qui n’est pas entouré de balises
* Il y a les balises qui ne servent à rien, ce sont deux balises qui n’ont aucune signification particulière.

Il y a la balise :

* **<span> </span>** : balise de type **inline** (balise placée au sein d’un paragraphe de texte pour sélectionner que certains mots), comme <strong> </strong> ou <em> </em>.
* **<div> </div>** : balise de type **block** (qui entoure un bloc de texte), comme <p> </p> et <h1> </h1>. Elles créent un nouveau bloc dans la page et font un retour à la ligne.

**Exemple sur fichier HTML et fichier CSS:**

**HTML :**

<p>Bonjour et <span class=”salutations”>bienvenue</span>sur mon site !</p>

**CSS :**

.salutations

{

color: blue;

}

**LES SÉLECTEURS AVANCÉS :**

* Pour sélectionner les éléments de la page que l’on veut modifier on utilise des sélecteurs. Il y en a à savoir par coeur comme :

**p** - Je veux toucher tous les paragraphes, reste juste à dire ce que l’on veut en faire (bleu,rouge…).

**{**

**}**



**h1, em** - Tous les titres et tous les textes importants (deux balise d’un coup).

**{**

**}**



**.class**  - On sélectionne des balises précises à qui nous avons donné un nom, grâce aux attributs class et id.

**{**

**}**

**#id**

**{**

**}**

* Ce sont les plus connues mais il en existe des dizaines d’autres. Comme…

**SÉLECTEUR UNIVERSEL**

**\*** - Sélectionne toutes les balises sans exception (sélecteur universel).

**{**

**}**

**UNE BALISE DANS UNE AUTRE**

h3 em - Sélectionne toutes les balises <em> situées à l’intérieur d’une balise

h3 (pas virgules entre les deux).

{

}

* **Exemple sur fichier HTML :**

<h3>Titre avec <em>texte important</em></h3>

**UNE BALISE QUI EN SUIT UNE AUTRE**

h3 + p - Sélectionne la première balise <p> située après un titre <h3>.

{

}

* **Exemple sur fichier HTML :**

<h3>Titre</h3>

<p>Paragraphe</p>

**UNE BALISE QUI POSSÈDE UN ATTRIBUT**

a[title] - Sélectionne tous les liens <a> qui possèdent un attribut title.

{

}

* **Exemple sur fichier HTML :**

<a href="http://site.com" title="Infobulle">

**UNE BALISE, UN ATTRIBUT ET UNE VALEUR EXACTE**

a[title=”cliquez ici”] - Même chose qu’avant mais l’attribut doit avoir exactement pour valeur “cliquez ici”.

{

}

* **Exemple sur le fichier HTML :**

<a href="http://site.com" title="Cliquez ici">

**UNE BALISE, UN ATTRIBUT ET UNE VALEUR**

a[title\*=”ici”] - Même chose mais l’attribut doit contenir dans sa valeur le mot “ici”(peu importe sa position).

{

}

**Exemple sur le fichier HTML :**

<a href="http://site.com" title="Quelque part par ici">

* pour une liste complète, il faut se rendre sur le site <https://www.w3.org/Style/css3-selectors-updates/WD-css3-selectors-20010126.fr.html#selectors> pour se renseigner.

**FORMATER DU TEXTE**

* On peut modifier l’apparence du texte (le mettre en forme), on peut modifier la taille du texte, changer la police et aligner le texte en CSS.

**LA TAILLE**

* Pour modifier la taille du texte on utilise la propriété **font-size**. Pour ce faire on peut utiliser soit :

- Une **taille absolue** : en pixels ou cm ou millimètres.

- Une **taille relative** : en pourcentage, “em” ou “ex”. Plus recommandé car elle s’adapte aux préférences de taille des visiteurs.

**Exemple de la taille absolue sur fichier CSS :**

p

{

**font-size**: 14**px**; /\* Paragraphes de 14 pixels \*/

}

h1

{

**font-size**: 40**px**; /\* Paragraphes de 40 pixels \*/

}

* On peut également définir des tailles en remplaçant px par “cm” ou “mm” (mais cependant moins adaptées aux écrans).

**Exemple de la taille relative (recommandé) sur fichier CSS :**

* Il y a plusieurs moyens d’indiquer une valeur relative comme par exemple avec des mots anglais :
* **xx-small** : minuscule ;
* **x-small** : très petit ;
* **small** : petit ;
* **medium** : moyen ;
* **large** : grand ;
* **x-large** : très grand ;
* **xx-large** : gigantesque.

p

{

**font-size**: small;

}

h1

{

**font-size**: large;

}

**Exemple de la taille relative avec une autre solution :**

* 1**em** : taille normale, grossir le texte avec une valeur supérieure à 1 (1.3) et le rétrécir avec une valeur inférieure à 1 (0.8).
* Attention ! : mettre un point (.) et non pas une virgule (,).

p

{

**font-size**: 0.8**em**;

}

h1

{

**font-size**: 1.5**em**;

}

* On peut aussi utiliser “ex” à la place de “em” mais qui est plus petit de base et les pourcentages (80 %, 130 %).

**LA POLICE**

* Pour modifier la police on utilise la propriété **font-family**.

balise

{

**font-family**: police;

}

* Pour éviter les problèmes si l’internaute n’a pas la même police, on précise en général plusieurs nom de police.

balise

{

**font-family**: police1, police2, police3, police4, police5;

}

* Le navigateur essaiera d’appliquer la police1 puis s’il n’y arrive pas la police2 puis la police 3 ainsi de suite.

* En général on ajoute la police **serif** (police par défaut) à la fin ou **sans-serif**.
* police1, police2 sont bien sûr remplacé par le nom des polices comme :

Arial

Arial Black

Georgia

Times New Roman

…

* On écrit donc :

p

{

**font-family**: Arial, “Arial Black”, Georgia, “Times New Roman”, sans-serif;

}

* Arial Black écrit : “Arial Black” à cause des espaces tout comme “Times New Romans”.
* On peut utiliser une police personnaliser avec **@font-face** toutefois le nav des visiteurs du site devront télécharger automatiquement la police (avec un poids pouvant aller ou dépasser 1Mo).

Certains sites proposent des polices libres de droits comme <https://www.fontsquirrel.com/> qui propose des packs prêts pour CSS 3.

Les formats de fichiers de polices ne fonctionnent pas tous sur tous les nav !

* Les différents formats de fichiers qui existent et qu’il faut connaître sont les suivants :

**.ttf** - TrueType Font qui fonctionne sur IE9 et tous les autres navigateurs ;

**.eot** - Embedded OpenType qui fonctionne sur IE uniquement, toutes versions, ce format est propriétaire et produit par Microsoft ;

**.otf** - OpenType Font qui ne fonctionne pas sur IE ;

**.svg** - SVG Font, le seul format reconnu sur les iPhone et iPad pour le moment ;

**.woff** - Web Open Font Format qui est un nouveau format conçu pour le Web, qui fonctionne sur IE9 et tous les autres navigateurs.

- Pour définir une nouvelle police on doit procéder comme ceci :

**@font-face**

{

**font-family**: ‘MaSuperPolice’;

**src**: url(‘MaSuperPolice.eot’);

}

Le fichier contenant MaSuperPolice.eot doit être situé dans le même dossier que le fichier CSS ou alors dans un sous dossier si on utilise un chemin relatif.

- On peut également proposer plusieurs formats de police (c’est mieux) pour que le nav télécharge celui qu’il sait lire, exemple :

**@font-face**

{

**font-family**: ‘MaSuperPolice’;

**src**: url(‘MaSuperPolice.eot’) format (‘eot’),

url(‘MaSuperPolice.woff’) format (‘woff’),

url(‘MaSuperPolice.ttf’) format (‘truetype’),

url(‘MaSuperPolice.svg’) format (‘svg’);

}

* Sur FontSquirrel, il faut cliquer sur “Webfont Kit” qui permettra d’obtenir un kit prêt à l’emploi avec tous les formats pour la police.
* Le fichier CSS ressemble donc à ceci avec utilisation de la police “CAC Champagne” de Font Squirrel :

**@font-face /\*** Définition d’une nouvelle police nommée CAC Champagne \*/

{

**font-family**: ‘cac\_champagneregular’;

**src**: url(‘cac\_champagne-webfont.eot’);

**src**: url(‘cac\_champagne-webfont.eot?#iefix’) format (‘embedded-opentype’),

url(‘cac\_champagne-webfont.woff’) format (‘woff’),

url(‘cac\_champagne-webfont.ttf’) format (‘truetype’),

url(‘cac\_champagne-webfont.svg#cac\_champagneregular’) format (‘svg’);

}

h1 /\* utilisation de la police que l’on vient de définir sur les titres \*/

{

**font-family**: ‘cac\_champagneregular’, Arial, serif;

}

* **Font-face** permet de définir un nouveau nom de police qui sera utilisé dans le fichier CSS, ensuite avec la propriété font-family, nous l’utilisons pour modifier l’apparence des titres avec h1.
* La présence de #iefix est dû au site Font Squirrel pour pallier certains bugs de IE.

**ITALIQUE, GRAS ET SOULIGNÉ…**

**ITALIQUE**

* En CSS pour mettre en italique on utilise **font-style** qui peut prendre la valeur :

**italic** - Le texte sera mis en italique ;

**oblique** - Le texte sera passé en oblique (les lettres sont penchées, le résultat est légèrement différent de l'italique proprement dit) ;

**normal** - Le texte sera normal (par défaut). Cela vous permet d'annuler une mise en italique. Par exemple, si vous voulez que les textes entre <em> ne soient plus en italique, vous devrez écrire :

em

{

**font-style**: normal;

}

* Quand on se sert de **font-style** pour mettre en italique tous les titres cela donne ceci :

h2

{

font-style: italic;

}

**GRAS**

* La propriété CSS pour mettre en gras est **font-weight** et prend les valeurs suivantes :

**bold** - Le texte sera en gras ;

**normal** - Le texte est écrit normalement (par défaut).

Exemple titre en gras :

h1

{

**font-weight**: bold;

}

**SOULIGNEMENT ET AUTRES DÉCORATIONS**

* La propriété CSS qui permet ceci est **text-decoration** et elle peut prendre les valeurs suivantes :

**underline** - Souligné ,

**line-through** - Barré ;

**overline** - Ligne au-dessus ;

**blink** - Clignotant, ne fonctionne pas sur tous les navigateurs (IE et Google Chrome);

**none** - Normal (par défaut).

Exemple :

h1

{

**text-decoration**: blink;

}

.souligne

{

**text-decoration**: underline;

}

.barre

{

**text-decoration**: line-through;

}

.ligne\_dessus

{

**text-decoration**: overline;

}

**L'ALIGNEMENT**

* 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.

Exemple :

h1

{

**text-align**: center;

}

p

{

**text-align**: justify;

}

.signature

{

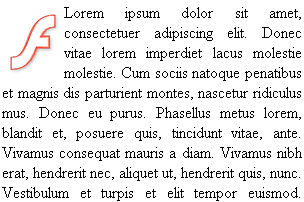
**text-align**: right;

}

* On peut modifier l’alignement des balises block (<p>, <div>, <h1>, et non pas les balises inline (<span>, <a>, <em>, <strong>), c’est donc en général le paragraphe entier qu’il faut aligner.

**LES FLOTTANTS**

* Le CSS permet de faire flotter un élément autour du texte, on dit qu’on fait un “habillage”, comme la figure ci-contre.



* Pour ce faire on utilise la propriété **float** (flottant en anglais) qui peut prendre deux valeurs simples :

**left** - L’élément flottera à gauche ;

**right** - L’élément flottera à droite.

* Pour l’utiliser, on applique un **float** à une balise puis on continue à écrire du texte à la suite.

On peut utiliser cette propriété (float) sur des balises block et aussi inline.

* Faire flotter une image, Exemple :

- Dans un premier temps, placer l’élément flottant dans le code HTML :

<p><img src="flash.gif" class="imageflottante" alt="Image flottante" /> Ceci est un texte normal de paragraphe, écrit à la suite de l'image et qui l'habillera car l'image est flottante.</p>

- Puis dans le code CSS :

.imageflottante

{

**float**: left;

}

Pour faire flotter l’image à droite, il suffit de mettre **right** au lieu de **left**.

**STOPPER UN FLOTTANT**

* En CSS il existe une propriété qui permet de dire stop, “ce texte doit être en dessous du flottant et non plus à côté”, il s’agit de la propriété **clear** qui peut prendre trois valeurs :

**left** - Le texte se poursuit après un **float**: left;

**right** - Le texte se poursuit après un **float**: right;

**both** - Le texte se poursuit en dessous, que ce soit après un **float**: left; ou un **float**: right; .

* Exemple avec **clear**: **both**;
* Sur le fichier HTML :

<p><img src="flash.gif" class="imageflottante" alt="Image flottante" /></p>

<p>Ce texte est écrit à côté de l'image flottante.</p>

<p class="dessous">Ce texte est écrit sous l'image flottante.</p>

* Puis sur fichier CSS :

.imageflottante

{

**float**: left;

}

.dessous

{

**clear**: both;

}

* On applique **clear**: **both**; au paragraphe que l’on souhaite voir continuer sous l’image flottante, et le tour est joué.