



Les feuilles de style CSS

Formation

Développeur Web et Web Mobile

Module : 03

Développer la partie front-end d'une application web

Séquence : 01


Réaliser une interface utilisateur web statique et adaptable

Séance : 01

Développer des pages web statiques en utilisant Html et Css

Libellé réduit :	CSS 3
Type de document :	Support de cours
Version :	1
Auteur :	Didier Bonneau
Centre afpa :	Créteil
Date de création :	11/01/2016
Date de mise à jour :	07/09/2022



	Auteur	Nom région	Formation	Date Mise à jour	Page 2/ 21
	Didier Bonneau	GRN164	DWWM	07/09/2022	122-Support-De-Cours_CSS.docx

Sommaire

Contenu

I.	Introduction	5
A.	Historique	5
B.	Principe des feuilles de style CSS	5
II.	Les feuilles de style	6
A.	Structure de base	6
1.	Les règles @	6
2.	Les règles de sélection.....	7
B.	Notion de cascade	7
C.	Enregistrement d'une feuille de style	8
1.	Feuille de style externe.....	8
2.	Feuille de style interne.....	9
3.	Style CSS dans une balise Html.....	9
III.	Les différents sélecteurs	9
A.	Le sélecteur id	9
B.	Le sélecteur class	9
C.	Le sélecteur de type	10
D.	Le sélecteur descendant.....	10
E.	Le sélecteur universel (*).....	10
F.	Le sélecteur d'enfants (>)	10
G.	Le sélecteur de frères adjacents (+)	10
H.	Le sélecteur d'adjacence (~).....	11
I.	Le sélecteur d'attributs	11
J.	Les sélecteurs multiples.....	11
K.	Les pSeudo-sélecteurs	12
IV.	Mettre en forme du texte	13
A.	Notions importantes.....	13
B.	Choix de la police	13
C.	Unité de taille des caractères	13
D.	Choix de couleurs	14
E.	Autres attributs	15
F.	Propriété @font-face (css3).....	15
V.	Bordure en CSS	16
A.	Style de bordures.....	16
VI.	Les marges	17
A.	Déclaration des tailles.....	17

VII.	Image de fond en CSS.....	17
VIII.	Effets rollover à l'aide de : HOVER	18
A.	Transformation d'apparence du texte	18
B.	Cumuler image de fond et caractère.....	19
IX.	Le positionnement.....	20
A.	La propriété « position »	20
B.	La propriété « float »	20
C.	La propriété « display »	21

I. INTRODUCTION

A. HISTORIQUE

La CSS (Cascading Style Sheets : Feuilles de Style en Cascade) a été développée par le W3C à partir de 1996.

Elle a pour but de séparer totalement la présentation d'une page Web de son contenu (c'est à dire du langage (HTML), et de faciliter ainsi la maintenance et l'accessibilité d'un site.

Sur le site du W3C vous pouvez consulter en ligne les recommandations CSS. Il existe d'autres recommandations (HTML, HTTP, XML, ...)

- CSS 1 → 1996
- CSS 2 → 1998
- CSS 2.1 → 2001 (1^{er} workingdraft)
 - En 2007 aucun navigateur n'a implanté 100% de la recommandation
- CSS 3 → 2005 (1^{er} workingdraft)
 - En 2008, aucun module n'est considéré comme terminé par le W3C
- CSS 4 → démarré en 2010

B. PRINCIPE DES FEUILLES DE STYLE CSS

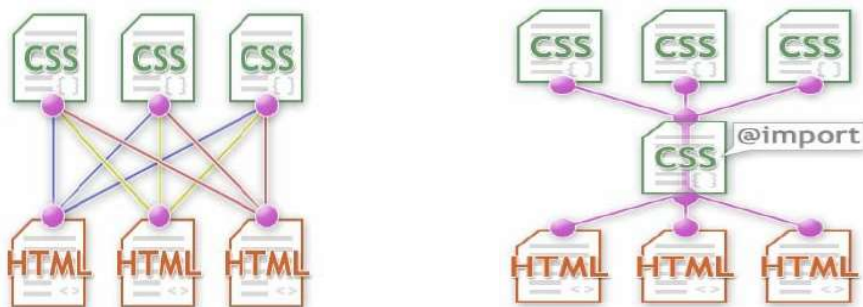
Auparavant la mise en page et l'ergonomie d'un site web se faisait directement dans le contenu des pages html. Ceci était un handicap majeur pour la maintenance et l'évolution du site.

Ce que les feuilles de style CSS apportent, c'est la séparation entre le contenu du site (contenu des balises html) et la mise en forme du site (couleur, position,...).

Une feuille de style est commune à l'ensemble des pages html du site.



Si plusieurs feuilles de styles sont définies, il est possible d'utiliser la directive « **@import** » mais attention à la compatibilité des navigateurs.



II. LES FEUILLES DE STYLE

A. STRUCTURE DE BASE

Quelle que soit la version de CSS, une feuille de style CSS consiste en une liste de déclarations. Elles sont de deux sortes :

- les règles at (@)
- les jeux de règles de sélection

1. Les règles @

Les règles « at » commencent par un mot-clé « at », composé du caractère « @ » immédiatement suivi par un identifiant (ex. @import, @page, @media) et se termine au premier point-virgule.

Les plus utiles sont :

- @charset : définit le jeu de caractères utilisé par la feuille de style.
- @import : indique au moteur de rendu CSS d'inclure une feuille de style externe.
- @font-face : permet de spécifier une police de caractère externe.

Il existe des règles @ dites conditionnelles de groupe :

- @page : utilisée pour modifier certaines propriétés CSS lors de l'impression d'un document. Vous pouvez seulement modifier les marges, les veuves et orphelins ainsi que les sauts de page du document.
- Des pseudos classes existent afin de traiter la première page, les pages de gauche ou les pages de droite : « :first », « :left » ou « :right ».


Exemple :

```
@page {
    size:landscape;
    margin-top:1.7cm;
    margin-bottom:1.4cm;
    margin-left:5cm;
    margin-right:5cm;
}
@page { size:21.0cm 14.85cm;
        margin-top:1.5cm; margin-bottom:2cm }
@page :left { margin-left:1.5cm; margin-right:2cm }
@page :right { margin-left:2cm; margin-right:1.5cm }
```

Pour les sauts de page, utiliser le style « page-break-before:always »

- @media : permet de définir des règles css en fonction du média. Les médias peuvent être : all, print, screen, embossed (braille)

Exemple :

	Auteur	Nom région	Formation	Date Mise à jour	Page 6/ 21
	Didier Bonneau	GRN164	DWWM	07/09/2022	122-Support-De-Cours_CSS.docx

```

@media print {
    body { font-size: 10pt }
}
@media screen {
    body { font-size: 13px }
}
@media screen, print {
    body { line-height: 1.2 }
}

```

- @document : permet de définir des règles qui ne s'appliqueront qu'à certaines pages html en fonction de leurs urls.

2. Les règles de sélection

La déclaration d'une règle css est toujours la même :

sélecteur {
liste de couples « propriété:valeur » séparés par un « ; »
}

- sélecteur : c'est la balise html (**body**, **h1**, **p**, etc.), l'identifiant (**id**) ou la classe (**class**);
- propriété : c'est l'attribut qu'on veut appliquer (**font** ; **background** ; **margin** ; etc.)
- valeur : précise les caractéristiques de la propriété.

B. NOTION DE CASCADE

Cascading: les caractéristiques de présentations se propagent "en cascade" d'un élément à ses fils.

Si les définitions sont contradictoires, c'est la définition la plus spécifique qui l'emporte.

```

<html>
  <head>
    <style>
      p {color:#FF0000;}
    </style>
  </head>
  <body>
    <p style="color: green">texte1</p>
    <p>texte2</p>
    <p><font color="blue">texte3</font></p>
  </body>
</html>

```

- le texte 1 sera vert
- le texte 2 sera rouge
- le texte 3 sera bleu

On peut déclarer les styles à différents endroits, et selon ces endroits ils seront plus ou moins prioritaires. On obtient donc une cascade de styles.

1. Déclaration des styles dans une feuille de style externe : **c'est de loin la meilleure chose à faire et la plus pratique à maintenir**, mais c'est celle qui a le moins de poids.
2. Déclaration des styles en interne, dans l'en-tête de la page : à ne faire qu'avec des styles particuliers à une page. Les styles déclarés auront plus de poids que ceux de la feuille de style externe et donc l'emporteront en cas de conflits.

3. Déclaration des styles en attributs des éléments html : à faire pour des mises en forme ponctuelles mais à éviter au maximum. Ces styles l'emporteront sur tous les autres.

C. ENREGISTREMENT D'UNE FEUILLE DE STYLE

1. Feuille de style externe

La bonne pratique est de positionner les styles dans un fichier externe portant l'extension « .css » qui sera lié au code html par l'insertion dans le paragraphe « head » d'une balise « link ».

Exemple de fichier *exemple.css* et du code *html*

exemple.css

```
html, body {
    margin: 0;
    padding: 0;
}
body {
    background-color: white;
    font-family: Verdana, sans-serif;
    font-size: 100%;
}
h1 {
    font-size: 200%;
    color: navy;
    text-align: center;
}
p,ul,li,td {
    color: black;
}
```

Ceci est
conseillé pour la
compatibilité des
navigateurs

Fichier html utilisant cette feuille de style :

```
<html>
<head>
    <link href="exemple.css" rel="stylesheet" media="all" type="text/css">
    <title>CSS externe</title>
</head>
<body>
    <h1>Exemple de feuille de style externe</h1>
    <p>ceci est un essai de feuille de style externe</p>
</body>
</html>
```

Il est précisé « **media="all"** » dans les attributs de la déclaration de la feuille de style. Cela indique que cette feuille sera lisible par tous les types de medias (écrans, TV, imprimante...). Pour une feuille de style applicable aux écrans, on spécifiera « **media="screen"** », pour une feuille de style réservée à l'impression, ce sera « **media="print"** ».

2. Feuille de style interne

Si l'on doit appliquer des styles à une page unique, on peut alors déclarer ces styles directement dans la page html entre les balises suivantes :

```
<style type="text/css" media="screen">
...les différentes règles de style...
</style>
```

Tout ceci est à placer, comme précédemment, entre les balises **<head></head>**.

3. Style CSS dans une balise Html

De façon encore plus ponctuelle, si l'on veut attribuer un style à une seule balise, on peut déclarer le style à l'intérieur même de cette balise html à l'aide de l'attribut **style**.

Par exemple :

```
<p style="text-align: center; color:red">Paragraphe rouge centré</p>
```

III. LES DIFFERENTS SELECTEURS

A. LE SELECTEUR ID

Le sélecteur « **id** » s'applique aux balises html ayant un attribut « id ». La valeur de l'attribut id d'une balise doit être unique, ce qui veut dire que le sélecteur id ne s'appliquera qu'à une et une seule balise (contrairement au sélecteur **class** ci-après). C'est pour cela qu'il est plutôt utilisé à la mise en page qu'à la mise en forme de caractères.

Il est déclaré par un dièse suivi du nom de l'id.

Il est déclaré par un dièse suivi du nom de l'id.

Dans la feuille de style :

```
#menu {
background-color: silver;
width: 100px;
float: left;
}
```

Dans le code html :

```
<div id="menu">
<ul>
<li>item 1</li>
<li>item 2</li>
</ul>
</div>
```

B. LE SELECTEUR CLASS

Le sélecteur « **class** » définit un ensemble de propriétés de style applicable à toute balise de la page html dont l'attribut class est égal à ce sélecteur.

Il est déclaré par un point suivi d'un nom de classe.

Toute balise portant l'attribut class="nomClasse" appliquera les styles.

Dans la feuille de style :

```
.haut {
text-align: right;
}
```

Dans le code html:

```
<p class="haut">
<a href="#hautPage">Haut de page</a>
</p>
```

C. LE SELECTEUR DE TYPE

Le sélecteur de type permet d'appliquer un ensemble de propriétés de style à toutes les balises html de ce type.

Dans la feuille de style :

```
p {
    font-weight: bold;
    text-indent: 1em;
}
```

Dans le code html :

```
<p>Le sélecteur de type</p>
```

D. LE SELECTEUR DESCENDANT

Un sélecteur descendant est constitué de deux sélecteurs simples séparés par un espace. Le sélecteur qui suit est le « fils » du premier sélecteur et ainsi de suite.

Dans la feuille de style :

```
div p {
    font-weight: bold;
}
div#section ul li.informations {
    color: red;
}
```

Dans le code html :

```
<div id="section">
    <p>Le sélecteur descendant</p>
    <ul>
        <li>Element noir</li>
        <li class="informations">Element
                                rouge</li>
    </ul>
</div>
```

E. LE SELECTEUR UNIVERSEL (*)

Le sélecteur universel « * » désigne tous les éléments possibles.

Dans la feuille de style :

```
div * span {
    color: red;
}
```

Dans le code html :

```
<div>
    <p>janvier<br/><span> Element rouge
                                </span></p>
    <span> Element noir </span>
</div>
```

F. LE SELECTEUR D'ENFANTS (>)

Permet de sélectionner un élément à l'intérieur d'un autre, uniquement s'il n'y a pas d'éléments imbriqués intermédiaires.

Dans la feuille de style :

```
div>span {
    color: red;
}
```

Dans le code html :

```
<div>
    <p>janvier</p>
    <p>février<span> Element noir</span></p>
    <span> Element rouge</span>
</div>
```

G. LE SELECTEUR DE FRERES ADJACENTS (+)

Permet de sélectionner un élément lorsque celui-ci succède directement à un autre.

Dans la feuille de style :

```
div+span {
    color: red
}
```

Dans le code html :

```
<div>
    <p> Element noir <span> Element noir
                                </span></p>
    <span> Element noir </span>
</div>
<span> Element rouge </span>
```

H. LE SELECTEUR D'ADJACENCE (~)

Permet de sélectionner un élément lorsque celui-ci succède directement un autre. Au contraire du sélecteur +, des balises peuvent être présentes entre la balise de fermeture du premier élément et celle d'ouverture du second (spécification du CSS3).

Dans la feuille de style :

```
div~span { color: red; }
```

dans le code html :

```
<div>Les mois</div>
<p>janvier <span>février</span></p>
<span>mars (element rouge )</span>
<span>avril (element rouge )</span>
```

I. LE SELECTEUR D'ATTRIBUTS

Le sélecteur d'attributs permet de sélectionner les balises via leurs attributs. On a la possibilité de sélectionner des attributs qui contiennent des éléments qui commencent ou terminent par une chaîne déterminée (CSS 3)

Dans la feuille de style :

```
a { color:blue ; }
a[lang='en'] { color: red ; }
a[lang='es'] { color: green ; }
p[attribut^="pre"] {color: red;}
```

le <p> dont l'attribut "attribut" commence par "pre"

```
p[attribut$="xieme"] {color: yellow;}
```

le <p> dont l'attribut "attribut" se termine par "xieme"

```
p[attribut*="sie"] {color: green;}
```

le <p> dont l'attribut "attribut" contient par "sie"

Dans le code html :

```
<a href="#" lang="fr">un site en français</a>
<a href="#" lang="en">un site en anglais</a>
<a href="#" lang="es">un site en espagnol</a>
<p attribut="premier">le premier paragraphe commence par pre</p>
<p attribut="deuxieme">le deuxième paragraphe se termine par xieme</p>
<p attribut="troisieme">le troisième paragraphe contient sie</p>
```

J. LES SELECTEURS MULTIPLES

Pour spécifier une même règle pour un ensemble de balises, il faut séparer les sélecteurs par une virgule.

```
p, h1, h2, h3 {font-family: verdana;}
h1, h2, h3 {font-weight: bold;}
h1 { text-align: center; color: #A0E0FF; }
p { text-align: justify; }
```

K. LES PSEUDO-SELECTEURS

Ils sont utilisés pour ajouter des effets à certains sélecteurs.

Ils se définissent en décorant le sélecteur de type par un suffixe séparé par « : ».

```
selecteur:pseudo-class {propriété:valeur;}
selecteur.class:pseudo-class {propriété:valeur;}
```

Attention toutefois à la compatibilité des navigateurs.

■ CSS 1

:link	Permet de définir l'apparence d'un lien lorsque celui-ci n'a pas encore été visité
:visited	Permet de définir l'apparence d'un lien lorsque celui-ci a déjà été visité
:active	Permet de définir l'apparence d'un élément voulu interactif lorsque celui-ci est actif, en train d'être cliqué
:first-letter	Permet de créer une lettrine, de définir l'apparence uniquement de la première lettre d'un paragraphe
:first-line	Permet de définir l'apparence uniquement de la première ligne d'un paragraphe

■ CSS 2

:hover	Permet de définir l'apparence d'un élément lorsque celui-ci est survolé
:after	Permet d'insérer un élément de contenu automatiquement après l'élément affecté
:before	Permet d'insérer un élément de contenu automatiquement avant l'élément affecté
:first-child	Permet de cibler le premier élément enfant d'un autre élément.
:focus	Permet de modifier l'apparence d'un élément interactif lorsque celui-ci prend le focus.
:lang	Permet de cibler un élément qui possède l'attribut lang avec une valeur précise

■ CSS 3

:nth-of-type(n)	Permet de définir l'apparence du n ^{ème} élément enfant d'un parent de type donné en partant du début
:nth-last-of-type(n)	Permet de définir l'apparence du n ^{ème} élément enfant d'un parent de type donné en partant de la fin
:first-of-type	Permet de définir l'apparence du 1 ^{er} élément enfant d'un type donné
:last-of-type	Permet de définir l'apparence du dernier élément enfant d'un type donné
:nth-child(n)	Permet de définir l'apparence du n ^{ème} élément enfant d'un parent quelconque en partant du début
:nth-last-child(n)	Permet de définir l'apparence du n ^{ème} élément enfant d'un parent quelconque en partant de la fin

IV. METTRE EN FORME DU TEXTE

A. NOTIONS IMPORTANTES

Avant de choisir sa, ou plutôt ses, polices de caractères pour sa page web, il faut bien connaître certaines contraintes :

- Toutes les polices de caractères n'existent pas sur tous les ordinateurs.
Conséquence : si l'ordinateur de votre visiteur ne possède pas la police que vous avez choisie, une autre s'affichera ruinant probablement le design attendu.
- Mac et PC ne possèdent pas toujours les mêmes noms de polices, ni même la même résolution.
Conséquence : bien vérifier les équivalences entre Mac et PC
- Tous les navigateurs ne gèrent pas le rendu des polices de la même façon.
Conséquence : là encore, n'espérez pas avoir le même rendu partout.

B. CHOIX DE LA POLICE

Polices génériques :

Il existe 5 familles de polices de caractères dites "génériques" :

- Serif
 - Times new roman (Pc) : police par défaut de tous les navigateurs
 - Times (Mac)
 - Georgia (Mac/Pc)
 - Palatino Linotype (Pc)
- Sans-serif
 - Verdana (Mac/Pc)
 - Arial (Mac/Pc) : police par défaut de tous les navigateurs
 - Helvetica (Mac)
- Monospace
 - Courier New (Pc) : police par défaut de tous les navigateurs
- Cursive
- Fantasy

Exemple dans le fichier css :


```
body {
    font-family: trebuchet, helvetica, sans-serif;
}
```

Le fait de mettre plusieurs polices oblige le navigateur à les essayer dans l'ordre de leurs énumérations.

C. UNITE DE TAILLE DES CARACTERES

Ce qu'il ne faut pas faire :

1. Utilisez des tailles de caractères fixes comme les points (**pt**), les picas (**pc**) ou les centimètres (**cm**). Celles-ci ne sont pas redimensionnables à l'écran, on doit toujours laisser la possibilité au visiteur de zoomer la page.

	Auteur	Nom région	Formation	Date Mise à jour	Page 13/ 21
	Didier Bonneau	GRN164	DWWM	07/09/2022	122-Support-De-Cours_CSS.docx

2. Evitez aussi d'utiliser les pixels (**px**) car Internet Explorer les considère comme une unité de taille de caractère fixe, on se retrouve donc avec le même problème que précédemment.

Ce qu'il faut faire :

Il faut donc utiliser des unités relatives, telles que les **em** ou les **%**. Ces unités sont proportionnelles à la taille en pixels déclarée dans le navigateur. Par défaut, ceux-ci sont en général réglés à 16px. C'est donc une taille qui peut être modifiée par l'utilisateur... On n'a aucun pouvoir là-dessus.

La taille des caractères se déclare par l'attribut « **font-size** ».

Attention à l'héritage :

Considérons l'exemple suivant :

```
body {
    font-family: arial, sans-serif;
    font-size: 90%;
}
p {
    font-size: 90%;
}
```

Les textes inclus dans les paragraphes n'auront pas une taille de 90% des 16 pixels déclarés par défaut (soit 14,4 pixels), mais 90% des 90% des 16 pixels (soit à peu près 13 pixels...). En effet, le paragraphe hérite des propriétés de son/ses parent(s), donc ici des valeurs déclarées dans le **body**.

Il faut donc bien faire attention aux valeurs déclarées. Il est donc en général plus sage, et surtout plus simple, de déclarer la taille à 100%, et de ne la réduire (ou l'augmenter) que ponctuellement.


Exemple :

```
body {
    font-family: arial, sans-serif;
    font-size: 100%;
}
h1 {
    font-size: 200%;
}
#footer p {
    font-size: 90%;
}
```

Ici, seuls les paragraphes **<p>** de la balise ayant l'**id="footer"** auront une taille plus petite que les autres.

D. CHOIX DE COULEURS

Les couleurs se déclarent grâce à l'attribut « **color**, » et à l'aide de codes hexadécimaux ou rvg (on évite les noms).

	Auteur	Nom région	Formation	Date Mise à jour	Page 14/ 21
	Didier Bonneau	GRN164	DWWM	07/09/2022	122-Support-De-Cours_CSS.docx

```
body {
  font-family: arial, sans-serif;
  font-size: 100%;
  color: #000000; /*code hexadécimal du noir*/
}
```

E. AUTRES ATTRIBUTS

Les autres attributs possibles pour modifier les caractères sont :

- **font-style: normal | italic** : normale | italique
- **font-variant: normal | small-caps** : normale | petites capitales
- **font-weight: normal | bold** : normal ou gras
- **text-align: left | right | center | justify** : aligne à gauche | aligne à droite | centre | texte justifié
- **text-decoration: none | underline | overline | line-through | blink** : rien | souligne | surligne | raye | clignotant
- **text-transform: none | capitalize | uppercase | lowercase** : met en majuscule la 1^{ère} lettre d'un mot | met en majuscules | met en minuscules
- **line-height: normal | nombre | %**

F. PROPRIÉTÉ @font-face (CSS3)

La limitation du choix des polices de caractère, peut être résolue par la propriété @font-face. Elle permet en effet de faire visualiser à l'écran une police de caractère définie, même si elle n'est pas installée sur l'ordinateur du visiteur.

Soit le code html suivant avec le CSS :

```
<html>
  <head>
    <title>Essai de css @font-face</title>
    <style type="text/css">
      @font-face {
        font-family: maPoliceCeltique;
        src: url(celticmd.ttf) format("truetype");
      }
      .celtique {
        font-family: maPoliceCeltique, sans-serif;
        font-size: 4em;
        color: #000000;
      }
    </style>
  </head>
  <body>
    <h4 class="celtique">
      Ma belle police de caractères celtiques
    </h4>
  </body>
</html>
```

V. BORDURE EN CSS

Délimiter une zone, mettre en relief une information ou entourer les cellules d'un tableau, tout ceci est très répandu sur les pages web.

La propriété permettant d'obtenir une bordure en CSS est tout simplement « **border** » qui peut être divisé en « border-bottom », « border-top », « border-left », « border-right ».
A cet attribut, on pourra associer une épaisseur de bordure (**border-width**), un style (**border-style**) et une couleur (**border-color**).

Chacun de ces attributs peut de la même façon être décliné en quatre autres en ajoutant bottom, top, left ou right au milieu (ex : border-bottom-width).

Si toutes les bordures doivent être identiques ces trois informations peuvent être regroupées en une seule : « border: width style color ».

A. STYLE DE BORDURES

Dix styles sont possibles en CSS 2.1

- **none** (valeur par défaut)
- **dotted** (pointillé)
- **dashed** (tirets)
- **solid** (solide)
- **double** (double)
- **hidden** (caché)
- **groove** (rainurée)
- **ridge** (relief)
- **inset** (relief intérieur)
- **outset** (relief extérieur)

Exemple :

Dans la feuille de style :

```
p {
  border-width: 1px;
  border-style: dotted;
  border-color: #000000;
}
```

Equivalence :

```
p {
  border: 1px dotted black;
}
```

On peut aussi arrondir les coins grâce à la propriété CSS3 « **border-radius** ».

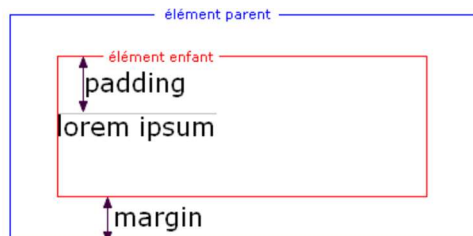
Attention, elle n'est pas encore implémentée dans tous les navigateurs, et il faut utiliser des préfixes spécifiques :

- moz-** pour les navigateurs firefox, seamonkey
- webkit-** pour les navigateurs safari, chrome.

```
div {
  border: 3px double black;
  border-radius: 10px;
  -moz-border-radius: 10px;
  -webkit-border-radius: 10px;
}
```


VI. LES MARGES

Pour affecter des marges aux différents éléments d'une page web, les propriétés CSS à utiliser sont **margin**, pour les marges extérieures, et **padding** pour les marges intérieures. Pour chaque élément html on peut donc définir l'espacement qui le séparera des autres éléments (**margin**) et les espacements intérieurs dont il peut bénéficier (**padding**).



A. DECLARATION DES TAILLES

Les tailles de ces marges peuvent se déclarer en pixels (**px**), en **em**, en **%**, etc. Tout dépend si l'on veut qu'elles soient fixes ou proportionnelles.

On peut détailler les tailles des marges à l'aide des suffixes **-top** (haut), **-right** (droite), **-bottom** (bas), **-left** (gauche), ou les synthétiser avec deux ou quatre valeurs d'un seul coup.

Exemple :

```
margin: 2px 5px 2em 0;
```

```
padding: 2px 5px;
```

revient à :

```
margin-top: 2px;
margin-right: 5px;
margin-bottom: 2em;
margin-left: 0;
```

revient à :

```
padding-top: 2px;
padding-bottom: 2px;
padding-right: 5px;
padding-left: 5px;
```

Attention : d'après le "box model", lorsqu'on attribue une taille à un élément de type **block** (à l'aide d'un **width** ou d'un **height**), les marges viennent s'ajouter à cette taille.

Notez aussi le très pratique **margin: auto** qui permet, dans le cas d'un bloc muni d'une largeur, de centrer horizontalement un élément.

Exemple :


```
p {
    margin: auto;
    width: 200px;
    padding: 40px;
}
```

Le paragraphe aura une largeur totale de $200 + 2 \times 40 = 280$ pixels et sera centré dans son élément parent.

VII. IMAGE DE FOND EN CSS

Mettre une image de fond dans une page html est un exercice très répandu. Avec les feuilles de style on peut faire beaucoup de chose.

Les attributs à utiliser sont :

	Auteur	Nom région	Formation	Date Mise à jour	Page 17/ 21
	Didier Bonneau	GRN164	DWWM	07/09/2022	122-Support-De-Cours_CSS.docx

- background-image : définition de l'URL de l'image
- background-repeat
- background-position

L'image de fond se déclare en général pour la balise **body** (corps de page), mais on peut aussi l'appliquer à un bloc (`<div></div>`), ou un titre (`<h1></h1>`), une citation (`<blockquote></blockquote>`), etc.

```
body {
    background-color: white;
    background-image: url(images/image_de_fond.png);
}
```

Sans plus de renseignements, l'image de fond apparait en mosaïque (elle est répétée pour couvrir l'ensemble du fond) et défile en même temps que le texte.

Si on veut bloquer la répétition ou le défilement, utiliser les attributs :

- `background-repeat:no-repeat;`
- `background-attachment:fixed;`

background-repeat accepte 4 attributs : **no-repeat** ; **repeat-x** pour une répétition de l'image uniquement horizontale ; **repeat-y** pour une répétition uniquement verticale ; et **repeat** pour la mosaïque complète.

Pour positionner l'image à un endroit bien précis :

- `background-position:right top;` (à droite en haut)
- `background-position: 100% 0%;`
- `background-position:center center;` (centrée)
- `background-position: 50% 50%;`
- `background-position:right bottom;` (à droite en bas)

Vous aurez noté, bien sûr, que les pourcentages dans le **background-position** offrent plus de latitude que les attributs **[right|center|left]** **[top|center|bottom]**.

VIII. EFFETS ROLLOVER A L'AIDE DE : HOVER

Pour tout effet de rollover (c'est à dire de changement d'aspect d'un élément au survol de la souris), la pseudo-classe CSS à utiliser est : « **hover** ».

A. TRANSFORMATION D'APPARENCE DU TEXTE


Le B.A-BA du langage CSS : le lien qui change d'apparence au survol de la souris.

Dans le code html :

```
<a href="toto.html">Le Site de Toto</a>
```

Dans la feuille de style :

```
a:hover {
```

	Auteur	Nom région	Formation	Date Mise à jour	Page 18/ 21
	Didier Bonneau	GRN164	DWWM	07/09/2022	122-Support-De-Cours_CSS.docx

```

    color: red;
    text-decoration: none;
}

```

text-decoration peut prendre d'autres valeurs :

none : rien
underline : souligne
overline : surligne
line-through : barre
blink : clignotant | (ce dernier étant à éviter).

a:hover est à placer après le **a:link**. Ceci est important pour que ça fonctionne ! Il faut en effet respecter un ordre pour les pseudo-classes.

En plus de **:link** et **:hover**, il existe aussi **:active** et **:visited** (lien actif, lien visite). L'ordre complet à respecter est :

1. **:link**
2. **:visited**
3. **:hover**
4. **:active**

B. CUMULER IMAGE DE FOND ET CARACTERE

Pour pouvoir combiner une image de fond (ici gris_anim.gif) et un caractère (ici un guillemet), on peut utiliser l'attribut **content** et le pseudo-élément **before**.

```

#menu a:hover {
    background-image:url(images/gris_anim.gif);
    background-repeat:no-repeat;
}
#menu a:hover:before {
    content:" » ";
}

```

Application pour un menu déroulant :

Dans le code html :

```

<ul>
  <li><a href="">Item 1</a></li>
  <li><a href="">Item 2</a>
    <ul>
      <li><a href="">Item
2.1</a></li>
      <li><a href="">Item
2.2</a></li>
    </ul>
  </li>
  <li><a href="">Item 3</a></li>
</ul>

```

Dans la feuille de style :

```

li ul {
    display: none;
}
li:hover ul {
    display: block;
    position: relative;
    top: 0;
    left: -25px;
}

```

La liste imbriquée dans l'item 2 (li ul) est "inexistante" grâce au **display: none**. Dès qu'on passe sur l'élément de la liste contenant la sous liste, l'attribut **display** change de statut pour passer du "rien" (**none**) au "bloc" (**block**), et ainsi redevenir visible. Le sous menu apparaît. Les valeurs de positionnements ne sont là que pour faciliter "l'accrochage" de la sous liste avec la souris.

IX. LE POSITIONNEMENT

Le positionnement est un élément capital en css afin d'obtenir une structure correcte de la page html et des éléments qui y sont affichés.

On a vu qu'une page peut être découpée soit par des <div> soit par les balises sémantiques (nav, section,...) encore faut-il bien organiser ce découpage.

A. LA PROPRIETE « POSITION »

Pour cela on va affecter un identificateur « id » à nos div et les positionnées via la feuille css en modifiant la propriété « position » qui peut donc prendre plusieurs valeurs :

- **fixed** : l'élément est positionné par les propriétés css « top » et « left » qui font référence à la fenêtre. Il ne bougera plus même en cas de scrolling. Ce type de positionnement est très peu utilisé. L'élément est sorti de flux.
- **static** : c'est le positionnement par défaut. Les éléments sont dans le flux et se positionnent suivant leur nature (type inline ou block).
- **absolute** : l'élément est positionné par les propriétés css « top », « left », « bottom » et « right » mais sont relatives à son premier ancêtre positionné (par défaut la fenêtre) et non pas son parent direct. L'élément est sorti de flux.
- **relatif** : ce positionnement permet de décaler un élément par rapport à sa position de référence dans le flux. Les autres éléments (précédents ou suivants) ne sont pas affectés puisque qu'ils le considèrent toujours dans le flux.
Une autre possibilité offerte par ce positionnement est de servir de référent pour ses enfants qui pourront se positionner en absolu par rapport à lui.


B. LA PROPRIETE « FLOAT »

Une autre solution pour positionner des blocs est d'utiliser la propriété « float » qui prend les valeurs « left » ou « right ».

Le bloc est partiellement sorti du flux et est calé respectivement à gauche ou à droite de son conteneur. Il prend la largeur de son contenu et le reste du flux est décalé d'autant.

Elle a été créée pour l'effet typographique d'enroulage (le texte s'enroule autour d'une image).


La propriété « clear » qui prend les valeurs « left », « right » ou « both » permet à l'élément qui suit un élément flottant de ne pas subir ses effets.

	Auteur	Nom région	Formation	Date Mise à jour	Page 20/ 21
	Didier Bonneau	GRN164	DWWM	07/09/2022	122-Support-De-Cours_CSS.docx

C. LA PROPRIÉTÉ « DISPLAY »

Il est aussi possible de jouer sur le rendu visuel des éléments html. On a vu que chaque balise possède un rendu par défaut (inline, block, inline-block) que l'on peut modifier par la propriété « display » avec les valeurs :

- inline : les éléments se placent les uns à la suite des autres sans pouvoir être redimensionnés ni bougés.
- block : les éléments se placent les uns en dessous des autres en prenant toute la largeur possible mais cette fois peuvent être redimensionnés par leur propriétés « width » et « height ». Il est aussi possible de les positionner en fixant la propriété « margin ».
- inline-block : ils se placent les uns après les autres tout en gardant la possibilité de redimensionnement.
- none : l'élément n'est pas affiché et est sorti du flux. Il ne conserve pas sa place contrairement au fait de positionner sa propriété « visible » à « hidden ».
- table : comportement de table de type bloc pour un élément
- table-cell : permet de centrer verticalement le contenu d'un block par la propriété « vertical-align:middle ».

	Auteur	Nom région	Formation	Date Mise à jour	Page 21/ 21
	Didier Bonneau	GRN164	DWWM	07/09/2022	122-Support-De-Cours_CSS.docx