



OFPPT

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle
et de la Promotion du Travail

Institut Spécialisé de Technologie Appliquée de Gestion Bab Tizimi - Meknès

Les Feuilles de style en CSS3

Année de Formation : 2020/2021

Table des matières

| | |
|--|----|
| 1) QU'EST-CE QUE C'EST CSS ? | 4 |
| 2) Déclaration d'un style CSS3 : | 4 |
| • Style en ligne, | 4 |
| • Style du document: | 4 |
| • Style externe..... | 6 |
| • Style importé | 7 |
| 3) Les sélecteurs CSS3 ? | 8 |
| Définition : | 8 |
| Les sélecteurs simples. | 8 |
| Le sélecteur universel..... | 8 |
| Les sélecteurs de type | 9 |
| <i>Les sélecteurs d'identifiant</i> | 10 |
| Les sélecteurs de classe..... | 11 |
| Les sélecteurs combinatoires. | 14 |
| Les sélecteurs de voisins directs (+). | 14 |
| Les sélecteurs de voisins (~). | 14 |
| Les sélecteurs des éléments fils(>). | 15 |
| Les sélecteurs des éléments descendants (espace). | 15 |
| Les sélecteurs d'attributs : | 16 |
| Les pseudo-classes : | 18 |
| Les pseudo-éléments..... | 19 |
| 4) Manipulation des feuilles de style..... | 21 |
| 1) Les styles d'un document HTML <body>:..... | 21 |
| Remplissage du document HTML : | 21 |
| Les marges d'un document HTML : | 22 |
| Les bordures d'un document HTML : | 23 |
| Les marges intérieures ou les padding d'un document HTML : | 24 |
| Les bordures d'un document HTML : | 26 |
| L'ombre d'un document HTML : | 26 |
| 2) Les styles ou formatage d'un texte | 28 |
| 3) Mise en forme d'un paragraphe en CSS3 : | 30 |
| 5) Les Modèle de boîtes en CSS3 | 31 |
| Définition : | 31 |

| | |
|---|----|
| Dépassement de contenu ou overFlow : | 33 |
| Les feuilles de style des images en CSS3. | 37 |
| Les feuilles de style d'une liste non ordonnée : | 42 |
| Les feuilles de style d'un tableau. | 45 |
| 6) La mise en page d'un document HTML..... | 48 |
| Disposition d'un élément HTML en CSS3. | 48 |
| 7) Les transformations, les transitions et les animations en CSS3 | 57 |
| Les transformations en css3 | 57 |
| Les transitions en css3 | 60 |
| Les animations en css3 : | 61 |

Les feuilles de style CSS3

(cascading style sheets)

1) QU'EST-CE QUE C'EST CSS ?

Le CSS est un langage informatique (coté client¹) utilisé pour gérer la présentation du contenu d'un document HTML (la mise en forme de tous ces éléments HTML); il est recommandé par l'organisation W3C (World Wide Web Consortium²).

Le but du CSS est de séparer la structure d'un document HTML de sa présentation; le langage HTML sert à structurer le contenu du document HTML, et CSS sert à mettre en forme plus facilement ce contenu structuré.

2) Déclaration d'un style CSS3 :

Les styles peuvent être incorporés au sein du document HTML de trois manières différentes :

- **Style en ligne**, c'est-à-dire en tant qu'attribut des balises

Dans ce cas l'attribut style déclarée dans la balise ouvrante permet de mettre en forme l'élément HTML en question (dans lequel l'attribut style est déclaré) ; ce style ne concerne que cet élément HTML

Exemples :

```
<p style= "color :red "> j'ai une couleur rouge</p>(p1)
```

```
<p style= "background :blue" > j'ai un remplissage blue
</p>(p2)
```

- **Style du document**: déclaré dans l'en-tête, c'est-à-dire au sein des balises `<HEAD>` et `</HEAD>`

¹ Un langage coté client est un langage qui s'exécute au niveau du poste client ; à la demande de ce dernier le serveur envoie une page web sous la forme d'un document HTML qui s'affiche sur le poste client ; au contraire d'un langage coté serveur qui est un langage qui s'exécute au niveau du serveur et ce dernier envoie la réponse sous la forme d'un document HTML.

² C'est un organisme international qui développent les standards (les plateformes web) pour le web pour assurer une communication facile entre les utilisateurs du Web.

Leur site web validator.w3.org à pour but la validation des documents HTML pour les rendre conformes à ces normes (voir notre cours HTML5).

4 Dans la balise <head> on ajoute la balise <style></style> dans laquelle on déclare les styles des éléments HTML de la page.

Exemple :

```
<head>
  <title>CSS3</title>
  <style>
    p
    {
      color:red;
    }

  </style>
</head>

<body>

<p> j'ai une couleur rouge</p>
  <p> j'ai une couleur rouge</p>

</body>
```

Le code HTML et CSS :

```
<head>
  <title>CSS3</title>
  <style>
    p
    {
      color:red;
    }

  </style>
</head>

<body>

<p > j'ai une couleur bleu</p>
  <p> j'ai une couleur rouge</p>

</body>
```

Résultat :

j'ai une couleur rouge
j'ai une couleur rouge

- Le style déclaré dans la partie <head> concerne tous les éléments HTML en question.
- Le style en ligne a la priorité qu'un style déclaré dans la partie <head>.

Exemple :

j'ai une couleur rouge
j'ai une couleur rouge
j'ai une couleur blue

Le code HTML et CSS :

```
<head>
  <title>CSS3</title>
  <style>
    p
    {
      color:red;
    }
  </style>
</head>
<body>
  <p> j'ai une couleur rouge</p>
  <p> j'ai une couleur rouge</p>
  <p style="color:blue">j'ai une couleur blue</p>
</body>
```

- **Style externe**, déclaré dans un fichier à part dont l'extension est **.css**

Un fichier de style est un fichier d'extension *.css, il peut contenir des styles qui seront appliqués sur un document HTML ou plusieurs documents.

C'est la méthode la plus pratique et la plus souple.

Un style en ligne à la priorité par rapport au style du document et ceci à la priorité par rapport aux styles déclarés dans un fichier de style.

Exemple :

Soit le fichier styles.css qui contient les styles suivants :

```
p
{
    color:red;
}
body
{
    background: #17CCAB;
}
```

Pour relier ce fichier avec le document index.html, on ajoute le lien suivant :

<link rel="stylesheet" href="styles/styles.css"/>

Ou

@IMPORT URL("styles/styles.css");

La balise **<link />** permet au navigateur de retrouver l'emplacement du fichier de style en utilisant l'attribut **href** .

L'attribut **rel** : indique que le type du fichier est une feuille de style CSS.

Remarque :

Les styles d'un fichier de style peut être appliqué à plusieurs document HTML, en revanche les styles de document ou en ligne.

- **Style importé**, déclaré dans un fichier à part dont l'extension est **.css**

Les recommandations du W3C offrent une dernière façon d'inclure des feuilles de style dans un document: en important des feuilles de style. Il est en effet possible d'importer des feuilles de style externes au niveau de la déclaration du style de document, en insérant la commande *@IMPORT* immédiatement après la balise style :

```
<style>

@IMPORT URL(url de la feuille à importer);

</style>
```

Remarque :

Il est possible de définir plusieurs styles en utilisant les différents moyens qu'offrent les CSS. Ainsi, lorsque plusieurs feuilles de style externes sont appelées, on obtient ce que l'on appelle une cascade de styles, c'est-à-dire une combinaison de styles pour divers éléments HTML. Si plusieurs styles concernent le même élément, seul le dernier style sera conservé.

```
<LINK rel=stylesheet href="style1.css">
<LINK rel=stylesheet href="style2.css">
<LINK rel=stylesheet href="style3.css">
```

3) Les sélecteurs CSS3 ?

Définition :

Un **sélecteur** CSS est code Css de base qui permet de sélectionner un ou plusieurs éléments HTML sur lesquels s'applique un style Css.

```
sélecteur
{
  Propriété1: valeur1;
  Propriété2: valeur2;
  Propriété3: valeur3;
  Propriété4: valeur4;
}
```

Les sélecteurs simples.

Le sélecteur universel.

Ce sélecteur permet de cibler tous les éléments HTML du document HTML.


```
* {
  Margin :0px ;
}
```

Pour supprimer les marges de tous les éléments HTML

Les sélecteurs de type

Ce sont les sélecteurs qui permettent de cibler un ou plusieurs éléments HTML par le nom de la balise (tag).

Exemple :

```
h1 {
  font-size: 200%;
  color: navy;
  text-align: center;
}
h2 {
  font-size: 150%;
  color: red;
  padding-left: 15px;
}
```

Remarque :

Les styles Css3 s'exécutent ligne par ligne pour une balise, et le dernier qui sera appliqué si plusieurs styles concernent cette balise.

Exemple :

```
<head>
<title>Hello!</title>
<style>
p
{
  color: red;
  font-size :18px ;
}
p
{
  color:blue;
}

</style>
</head>
```

```

<body>
  <p>Je suis le premier paragraphe</p>
  <p>Je suis le deuxième paragraphe</p>
  <p>Je suis le troisième paragraphe</p>
  <h1>Je suis le titre de niveau 1</h1>
  <h2>Je suis le titre de niveau 2</h2>
  <h3>Je suis le titre de niveau 3</h3>

</body>

```

La couleur **bleu** qui sera appliquée sur les paragraphes.

NB : **N'oublier pas la remarque de priorité des styles**

Remarque : On peut regrouper des sélecteurs de même contenu pour minimiser le code.

```

p, h1, h2
{
  color:blue;
}

```

Exemple d'application :

Soit le code Html suivant :

```

<body>
  <div>
    <p>paragraphe 1<span> div1</span></p>
    <p>paragraphe 2<span> div2</span></p>
    <p>paragraphe 3<span> div3</span></p>
    <span> div4</span>
  </div>
  <p>paragraphe 4</p>
  <p>paragraphe 5</p>
  <p>paragraphe 6</p>
</body>

```

- 1) Changer le remplissage de votre page en rouge.
- 2) Modifier la couleur des paragraphes en bleu.
- 3) Changer le remplissage des div en vert.
- 4) Changer le remplissage des span en Jaune.
- 5) Modifier la couleur du paragraphe 4 en utilisant la couleur : #CC558F.
- 6) Modifier la couleur des paragraphe 5, 6 en utilisant la couleur: #1C22CC.
- 7) Que remarquer Vous ?

Les sélecteurs d'identifiant.

Ce sélecteur permet de cibler un élément HTML en fonction de la valeur de l'attribut ID de cet élément HTML, cette valeur est unique dans un document HTML ; cela veut dire qu'on ne peut pas trouver deux éléments HTML qui ont la même valeur de leurs attributs ID dans un même document HTML.

```

<div id="wikipedia">
  <h1> Nom du site </h1>

  <div id="contenu">
    <h1> Titre 1 </h1>
    <p>Paragraphe...</p>

    <h2> Titre 1.1 </h2>
    <p>Paragraphe...</p>
  </div>
  <div id="contenu1">
    <h1> Titre 11 </h1>
    <p>Paragraphe...</p>

    <h2> Titre 1.1.1 </h2>
    <p>Paragraphe...</p>
  </div>
</div>

```

```

#contenu1
{
  background:yellow;
}

```

8) Répondre aux questions 5,6 en utilisant les sélecteurs d'identifiants.

Remarque :

Lorsqu'on applique un style sur un élément HTML dont une propriété a plusieurs valeurs la priorité est de la gauche vers la droite selon la prise en charge de la valeur.

Exemple :

```

#contenu1
{
  background:yellow;
  font-family: Koufi serif Arial ;
}

```

Les sélecteurs de classe.

Ce sélecteur permet de cibler un ou plusieurs éléments HTML en fonction de la valeur de l'attribut **class** de ces éléments HTML, on peut trouver des éléments HTML qui ont la même valeur de leurs attributs **class** dans un même document HTML.

```

<div id="wikipedia">
  <h1 class="h1"> Nom du site </h1>
  <div id="contenu">
    <h1 class="h1"> Titre 1 </h1>
    <p>Paragraphe...</p>
    <h2> Titre 1.1 </h2>
    <p>Paragraphe...</p>
  </div>
  <div id="contenu1">
    <h1 class="h1"> Titre 11 </h1>
    <p>Paragraphe...</p>
    <h2 class="h1"> Titre 1.1.1 </h2>
    <p>Paragraphe...</p>
  </div>
</div>

```

```

.h1
{
  Color : #396DCC ;
}

```

Remarque :

- Dans un document HTML, le poids d'un sélecteur d'identifiant est plus que celle du sélecteur de classe.
- Un élément HTML peut avoir plusieurs valeurs de l'attribut class.

```

<head>
  <title>Hello!</title>
  <style>
    .h2
    {
      background: red;
    }
    #h1
    {
      background: yellow;
    }
  </style>
</head>

<body>
  <div id="wikipedia">
    <h1 class="h1"> Nom du site </h1>
    <div id="contenu1">
      <h1 class="h1 h2" id="h1"> Titre 11 </h1>
      <p>Paragraphe...</p>
      <h2 class="h1"> Titre 1.1.1 </h2>
      <p>Paragraphe...</p>
    </div>
  </div>
</body>

```

Le code HTML :

```

<head>
  <title>Hello!</title>
  <style>
    .h2
    {
      background: red;
    }
    #h1
    {
      background:yellow;
    }
  </style>
</head>

<body>
  <div id="wikipedia">
    <h1 class="h1"> Nom du site </h1>
    <div id="contenu1">
      <h1 class="h1 h2" id="h1"> Titre 11 </h1>
      <p>Paragraphe...</p>
      <h2 class="h1"> Titre 1.1.1 </h2>
      <p>Paragraphe...</p>
    </div>
  </div>

```

Exercice d'application :

```

<body>
  <div id="div1">
    <p class="p">paragraphe 1<span id="sp1"> div1</span></p>
    <p>paragraphe 2<span> div2</span></p>
    <p class="p" id="p3">paragraphe 3<span> div3</span></p>
    <span> div4</span>
  </div>
  <p>paragraphe 4</p>
  <p>paragraphe 5</p>
  <p>paragraphe 6</p>
</body>

```

- 1) Modifier le remplissage de l'élément (#div1), Sachant qu'il sera jaune.
- 2) Modifier la couleur des paragraphes de classe (.p) en bleue.
- 3) Modifier le texte (#sp1) en rouge, gras.
- 4) Mettre ce dernier texte en majuscule.
- 5) Souligner le paragraphe 4.
- 6) Mettre les paragraphes de (#div1) au centre de la page.
- 7) Modifier le code html de telle façon que les paragraphes (4,5,6) auront la même couleur que les paragraphes (2,3).

Les sélecteurs combinatoires.

Les sélecteurs de voisins directs (+).

Le combinatoire (+) permet de sélectionner les éléments HTML qui suivent directement un élément HTML.

```
<body>
  <div id="div1">
    <p class="p">paragraphe 1<span id="sp1"> div1</span></p>
    <p>paragraphe 2<span> div2</span></p>
    <p class="p" id="p3">paragraphe 3<span> div3</span></p>
    <span> div4</span>
  </div>
  <p>paragraphe 4</p>
  <p>paragraphe 5</p>
  <p>paragraphe 6</p>
</body>
```

Questions :

1) Quel est l'élément HTML qui suit directement la balise<div> ?.

<p>paragraphe 4</p>

2) Quel est l'élément HTML qui suit directement la balise<p> ?.

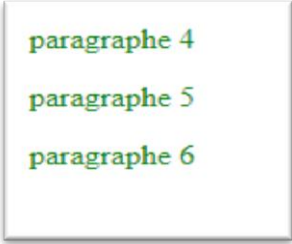
** div4**

Les sélecteurs de voisins (~).

Le combinatoire (+) permet de sélectionner les éléments HTML qui suivent un élément HTML et qui ont le même parent.

```
div~p
{
  color:green;
}
```

Le résultat est :



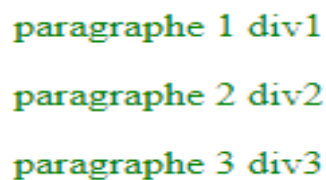
paragraphe 4
paragraphe 5
paragraphe 6

Les sélecteurs des éléments fils(>).

Le combinatoire (>) permet de sélectionner les éléments HTML qui sont des fils direct d'un élément HTML et qui ont le même parent.

```
div>p
{
  color:green;
}
```

Le résultat est :



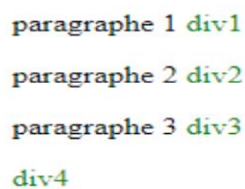
paragraphe 1 div1
paragraphe 2 div2
paragraphe 3 div3

Les sélecteurs des éléments descendants (espace).

Le combinatoire (espace) permet de sélectionner les éléments HTML qui sont des descendants d'un élément HTML et qui ont le même parent.

```
div span
{
  color:green;
}
```

Le résultat est :



paragraphe 1 div1
paragraphe 2 div2
paragraphe 3 div3
div4

Exercices d'application : TP1 CSS3

Les sélecteurs d'attributs :

C'est un sélecteur qui permet de modifier la mise en forme d'un élément HTML, selon leurs attributs ou les valeurs de ces attributs.

```
[attrib]
{
  Propriétés : valeurs ;
}
```

Ou

```
[attrib='valeur']
{
  Propriétés : valeurs ;
}
```

```
<div>
  <p class="p">paragraphe 1<span id="sp1"> div1</span></p>
  <p>paragraphe 2<span> div2</span></p>
  <p class="p" id="p3">paragraphe 3<span> div3</span></p>
  <span> div4</span>
</div>
<p>paragraphe 4</p>
<p>paragraphe 5</p>
<p>paragraphe 6</p>
```

Exemple1 :

```
[id]
{
  color:red;
}
```

Ce sélecteur cible les éléments HTML qui ont l'attribut **ID**.

```
paragraphe 1 div1
paragraphe 2 div2
paragraphe 3 div3
div4
paragraphe 4
paragraphe 5
paragraphe 6
```



```
[id='sp1']
```

```
{
  color:red;
}
```

paragraphe 1 div1

paragraphe 2 div2

paragraphe 3 div3

div4

paragraphe 4

paragraphe 5

paragraphe 6

Exemple 2 :

```
balise[attrib]
```

```
{
  Propriétés : valeurs ;
}
```

Ce sélecteur cible les éléments HTML qui ont le nom du tag « balise » et qui ont l'attribut **attrib**.

```
p[id]
```

```
{
  color:red;
}
```

Ce sélecteur cible les éléments HTML qui sont des paragraphes et qui ont l'attribut **ID**.

```
p[id='p3']
```

```
{
  color:red;
}
```

Ce sélecteur cible les éléments HTML qui sont des paragraphes et qui ont l'attribut **ID dont la valeur est p3**.

Exercice d'application TP1/ partie 2 & Partie3.

Les pseudo-classes :

C'est un type de sélecteur permet de définir le style des éléments HTML selon leur état.

Sélecteur:etat

```
{
  Propriétés : valeurs ;
}
```

Exemple 1 :

```
<div>
  <p class="p">paragraphe 1<span id="sp1"> div1</span></p>
  <p>paragraphe 2<span> div2</span></p>
  <p class="p" id="p3">paragraphe 3<span> div3</span></p>
  <span> div4</span>
</div>
<p>paragraphe 4</p>
<p>paragraphe 5</p>
<p>paragraphe 6</p>
<a href="http://www.ofppt.ma">lien vers la site de ofppt</a>
<a href="http://www.google.ma">lien vers la site de ofppt</a>
```

P:hover

```
{
  background: #CCCC18;
}
```

a:visited

```
{
  background: yellow;
}
```

La couleur du lien visité sera changée en jaune.

a:link

```
{
  background: green;
}
```

La couleur de l'élément HTML sera verte car c'est un lien.

Pour plus d'informations sur les propriétés de styles voir le site :

<https://yesyouweb.com/css3-guide-reference/>

Les pseudo-éléments

Ces sélecteurs représentent les éléments HTML qui ne sont pas décrite en HTML ; ils permettent de parcourir les éléments HTML par leur position dans ce document (par ligne, par caractères.....).

Sélecteur::position

```
{
  Propriétés : valeurs ;
}
```

Exemple1 :

```
<div>
  <p class="p">je suis la première ligne du paragraphe<br/>
  je suis la deuxième ligne du paragraphe</p>
</div>
```

```
p::first-line {
  font-size: 120%;
  font-weight: bolder;
}
```

```
p::first-letter {
  font-size: 120%;
  font-weight: bolder;
  color :red ;
}
```

Résultat :

je suis la première ligne du paragraphe
je suis la deuxième ligne du paragraphe

La propriété **content** permet d'insérer un texte au début ou après une élément HTML dans un document en utilisant le CSS,

```
.p::before {  
  content: "je serai inséré au début de vos paragraphes p."  
}
```

Le texte « je serai inséré au début de vos paragraphes p » sera inséré au début des paragraphes P.

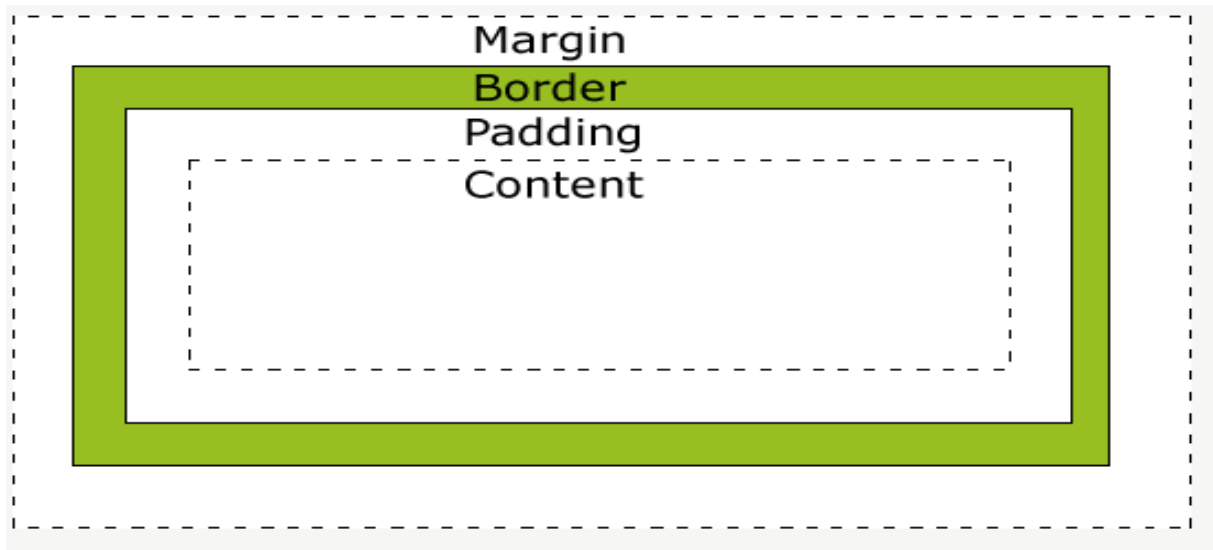
```
.p::after {  
  content: "je serai inséré au début de vos paragraphes p."  
}
```

Le texte « je serai inséré au début de vos paragraphes p » sera inséré au début des paragraphes P.

4) Manipulation des feuilles de style

1) Les styles d'un document HTML <body>:

Chaque élément HTML est considéré comme une boîte de dialogue qui caractérisée par les propriétés suivantes :



Remplissage du document HTML :

La propriété **Background** permet de définir le remplissage ou l'arrière-plan le document HTML ou un de ces élément Html.

Exemple 1 :

```
body
{
background: #18C9CC;
}
```

Exemple 2 :

```
body
{
background: linear-gradient(to bottom, #5CC3FF, #BF788D);
}
```

Exemple 3 ³:

```
body
{
background: radial-gradient (circle at 100%, #3f87a6, #ebf8e1, #f69d3c);
}
```

³ Utiliser le site : <https://cssgradient.io/> pour personnaliser le remplissage de votre document HTML.

```
}
```

On peut insérer une image en arrière-plan d'un document HTML.

```
body
{
background-image: url(images/image.jpg);
background-position: center;
background-repeat: no-repeat;
background-size: 800px 700px; /* width height */
background-attachment: fixed;
}
```

background-image: définit le chemin de l'image dans le serveur.

background-position: définit la position de l'image insérée dans le document HTML ; ses valeurs sont : **right, left, center, bottom, top** ou une combinaison de ces valeurs.

background-position: top right; (coins du document)

background-repeat : permet de répéter ou non l'image au long du document HTML, ses valeurs sont : **repeat repeat-x repeat-y no-repeat.**

background-size : définit la largeur et la hauteur de l'image ; ses valeurs sont : **width height** ou les valeurs **auto cover contain .**

background-attachment: définit l'état de l'image en cas de défilement du contenu du document HTML. Ses valeurs sont : **scroll fixed.**

Les marges d'un document HTML :

On peut regrouper les marges d'un document HTML comme suit :

Exemple 1 :

```
body
{
margin:5%;
}
```

Cette propriété applique la marge 5% de la largeur et de la hauteur du document HTML sur toutes les marges (marge à droite, marge à gauche, marge en haut – pour la marge en bas cela reste selon le contenu du document HTML).

On peut utiliser les propriétés suivantes pour détailler les marges :

| | |
|--------------------------|--------------------------------|
| margin-left:5%; | pour la marge à gauche. |
| margin-right:5%; | pour la marge à droite. |
| margin-top:5%; | pour la marge en haut. |
| margin-bottom:5%; | pour la marge en bas. |

Exemple 2 :

```
body
{
    margin-left:10%; /*pour la marge à gauche.*/
    margin-right:10%; /*pour la marge à droite.*/
    margin-top:5%; /*pour la marge en haut.*/
    /*margin-bottom:5%; pour la marge en bas.*/ (commentaire)
}
```

Comme on peut utiliser les propriétés suivantes :

```
body
{
    margin: 15% 30% 15% 20%; /* Top - Right - Bottom - Left */
}
```

```
body
{
    margin: 15% 30%; /* Top / Bottom - Right /Left */
}
```

```
body
{
    margin: 15% 30%; 15% /* Top - Right / Left - Bottom*/
}
```

Les bordures d'un document HTML :

Pour délimiter la zone d'un élément HTML ou d'un document HTML on peut utiliser la propriété Border et ses descendants.

Exemple 1 :

```
body
{
    border-style: solid;
```

```
border-width: 2px;
border-color: blue;
}
```

On peut regrouper ces propriétés en une seule : **border :solid 2px blue ;**

Pour détailler les bordures du document HTML, on utilise les propriétés suivantes :

```
border-left-style;;
border-right-style;;
border-top-style;
border-bottom-style;;
```

```
border-left-color;;
border-right-color;;
border-top-color;;
border-bottom-color;;
```

```
border-left-width;;
border-right-width;;
border-top-width;;
border-bottom-width;;
```

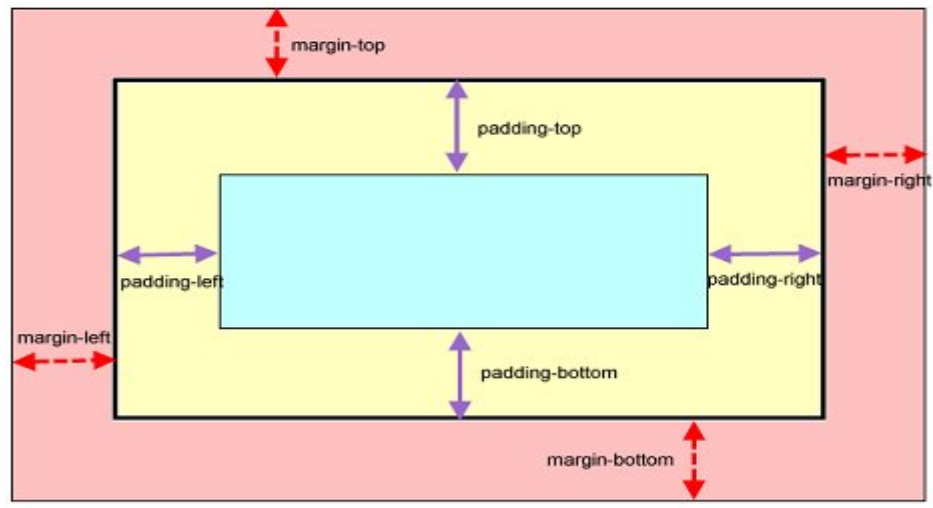
```
border-bottom-left-radius ;
border-top-left-radius ;
border-bottom-right-radius ;
border-top-right-radius ;
```

```
border-left;;
border-right;;
border-top;;
border-bottom;;
```

```
border: solid 2px black;
border-radius: 25px 25px 25px 25px;
border-width : 2px 2px 3px 5px; /* top right bottom left */
```

Les marges intérieures ou les padding d'un document HTML :

Ce sont les espaces internes qui séparent un élément HTML par rapport à son contenu.



Exemple :

```
body
{
  border:solid 2px black;
  padding: 5%;
}
```

Comme on a vu pour les marges (les marges extérieures on peut utiliser les descendants de la propriété padding qui sont :

Padding-left , padding-top, padding-right, padding-bottom

De même on peut utiliser les mêmes décompositions et les appréciations de la propriété margin.

```
Padding -left:5%;    pour la marge interne à gauche.
Padding -right:5%;   pour la marge interne à droite.
Padding -top:5%;     pour la marge interne en haut.
Padding -bottom:5%;  pour la marge interne en bas.
```

Exemple 2 :

```
body
{
  Padding -left:10%; /*pour la marge interne à gauche.*/
  Padding -right:10%; /*pour la marge interne à droite.*/
  Padding -top:5%;    /*pour la marge interne en haut.*/
  Padding -bottom:5%; /*pour la marge interne en bas.*/
}
```

Comme on peut utiliser les propriétés suivantes :

```
body
{
```

```
padding: 15% 30% 15% 20%; /* Top - Right - Bottom - Left */

}
```

```
body
{
padding: 15% 30%; /* Top / Bottom - Right /Left */

}
```

```
body
{

padding: 15% 30%; 15% /* Top - Right / Left - Bottom*/

}
```

Les bordures d'un document HTML :

Pour délimiter la zone d'un élément HTML ou d'un document HTML on peut utiliser la propriété Border et ses descendants.

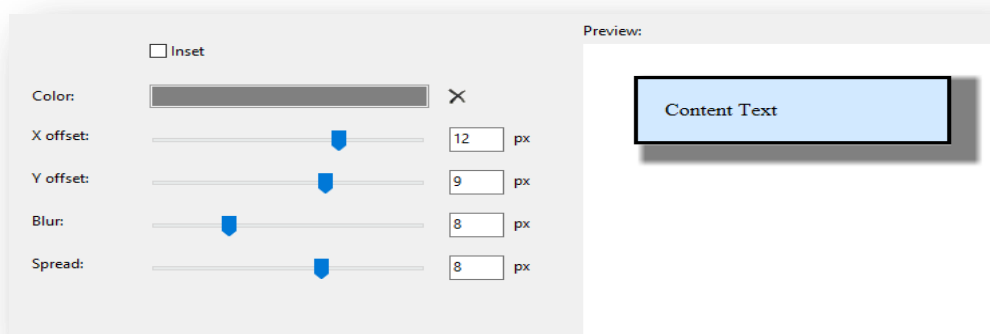
Exemple 1 :

```
body
{
border-style: solid;
border-width: 2px;
border-color: blue;
}
```

L'ombre d'un document HTML :

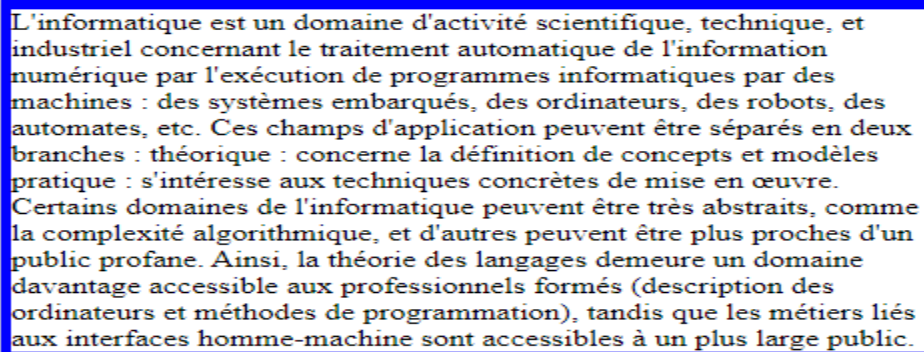
La propriété box-shadow permet d'appliquer un ombre à document HTML et à une boîte en général.

Ces valeurs sont :



```
body
{
    margin:10px;
    border-style: solid;
    border-width : 15px 18px 30px 5px;/* top right bottom left */
    border-color: blue;
    box-shadow: 12px 9px 8px 8px #808080 ;
}
```

Le résultat est :



L'informatique est un domaine d'activité scientifique, technique, et industriel concernant le traitement automatique de l'information numérique par l'exécution de programmes informatiques par des machines : des systèmes embarqués, des ordinateurs, des robots, des automates, etc. Ces champs d'application peuvent être séparés en deux branches : théorique : concerne la définition de concepts et modèles pratique : s'intéresse aux techniques concrètes de mise en œuvre. Certains domaines de l'informatique peuvent être très abstraits, comme la complexité algorithmique, et d'autres peuvent être plus proches d'un public profane. Ainsi, la théorie des langages demeure un domaine davantage accessible aux professionnels formés (description des ordinateurs et méthodes de programmation), tandis que les métiers liés aux interfaces homme-machine sont accessibles à un plus large public.

Exemple d'application : Tp1 , exercice 2.

2) Les styles ou formatage d'un texte

- **font-family** : Permet de définir la police du texte.

Exemple :

```
p
{
  font-family: Arial, Helvetica, sans-serif;
}
```

Les valeurs de la propriété **font-family** en commençant par la première ; si cette valeur n'est prise en charge le navigateur passe à la deuxième etc....

Remarque : pour charger un font qui n'est pas pris en charge en va voir la règle [@font-face](#)

⁴

- **font-size** : définit la taille de la police qui en px ou em.

```
p
{
  font-family: Arial, Helvetica, sans-serif;
  font-size: 20px ;
}
```

Comme elle peut prendre des valeurs comme :

larger smaller xx-small x-small small x-large xx-large

- **font-style**: définit le style du texte (normal italic oblique).

```
p
{
  font-family: Arial, Helvetica, sans-serif;
  font-size: 20px;
  font-style: italic;
}
```

⁴ @font-face {
 font-family: "Bitstream Vera Serif Bold";
 src: url("/static/styles/libs/font-awesome/fonts/fontawesome-webfont.fdf491ce5ff5.woff");
 }
 body {
 font-family: "Bitstream Vera Serif Bold", serif;
 }

- **font-weight** : permet de définir le style du texte s'il est en gras (ces valeurs sont : bold bolder)

text-decoration : définit le soulignement du texte ; ces valeurs sont: underline dotted; (underline overline double dotted)

- **font-variant**: normal small-caps ;
- **text-transform**: définit la casse du texte (capitalize uppercase lowercase full-width);
- **color** : permet de changer la couleur du texte ; sa valeur peut être le nom de la couleur ou un code précédé par #.

Exemple :

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

Ou

```
p
{
  color: #CCCC48;
}
```

Ou

```
p
{
  color: rgb(255,120,180);
}
```

text-shadow : définit l'ombre d'un texte.



Exemple :

```
p
{
  text-align: center;
  line-height: 20px;
  text-shadow: 4px 5px 5px #808080;
}
```

Soulignement d'un texte :

```
text-decoration: underline ;
text-decoration-color: #46CC4B;
text-decoration-line : double ;
text-decoration: overline solid #e4087e;
```

(Certaines de ces propriétés ne sont pas prises en charge par des navigateurs ; vous pouvez utiliser la propriété **border-button**)

3) Mise en forme d'un paragraphe en CSS3 :

- **text-align** : permet de définir l'alignement du texte d'un paragraphe, ces valeurs sont : **left right center justify start end start end**
- **line-height** définit l'interligne d'un paragraphe ; sa valeur est en px.

```
p
{
  text-align: center;
  line-height: 20px;
}
```

Exercices d'application TP1 Exercice 1 Partie 3:

5) Les Modèle de boites en CSS3

Définition :

Chaque élément HTML est considéré comme une boîte rectangulaire quel que soit cet élément en block ou inline.

- Les boites en inline :

La boîte inline ne crée pas de retour à la ligne; et les autres seront sur la même que la première. Certaines propriétés ne s'appliquent sur ces types de boîtes.

Exemples de boîte inline :<label> <a>

Exemples de propriétés non applicables : width, height.

```
<style>
  label
  {
    width:70px;
    height:40px;
    background:red;
  }
  span
  {
    width:70px;
    height:40px;
    background:green;
  }
</style>
</head>

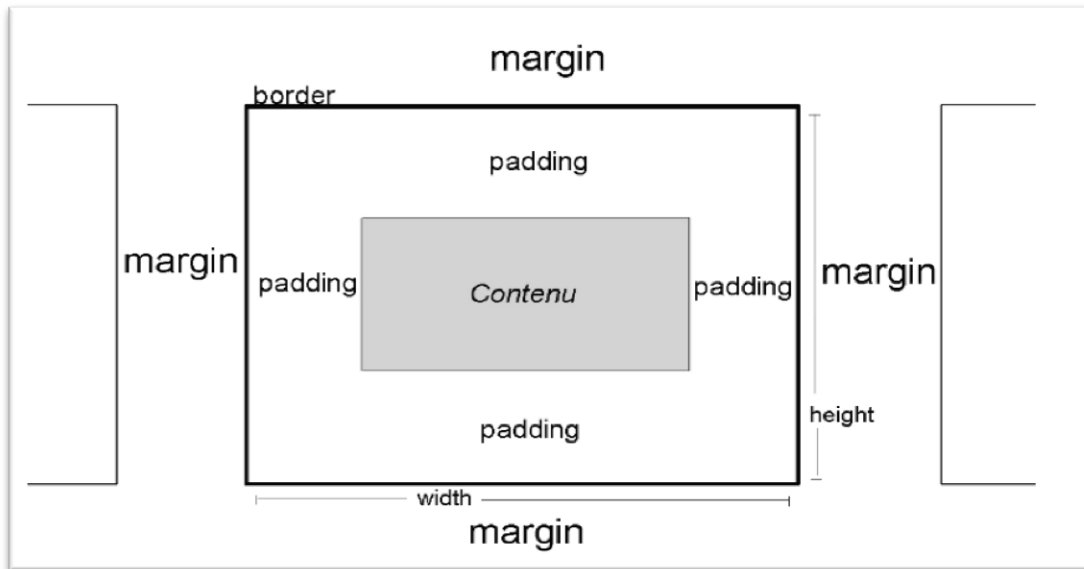
<body>
<label>Bonjour</label><span>Mes amis</span>
</body>
```

- Les boites en block ;

Ces types de boîtes appliquent des retours à la ligne ; on peut appliquer toutes les propriétés CSS sur ses Boîtes.

Exemple :<p><h1>...

Chaque boîte est caractérisée par :



margin : Il s'agit de la marge entre les différentes boîtes de la page HTML. Donc de l'espace à l'extérieur de ces boîtes.

border : Il s'agit de la bordure de la boîte. Par défaut, elle est invisible.

padding : Il s'agit de l'espace à l'intérieur de la boîte. Donc l'espace entre le contenu de la boîte et sa bordure.

width : Il s'agit de la **largeur** de la boîte.

height : Il s'agit de la **hauteur** de la boîte.

Exemple :

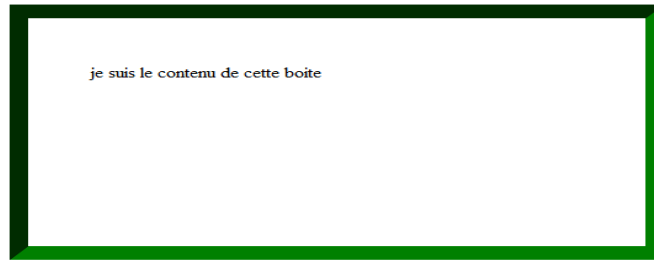
```
div
{
  border: solid 1px black;
  width: 50%;
  margin: 30px; ou bien :30% (de la largeur de la page);
  padding: 50px;
  height: 50px;
}
```

Remarque :

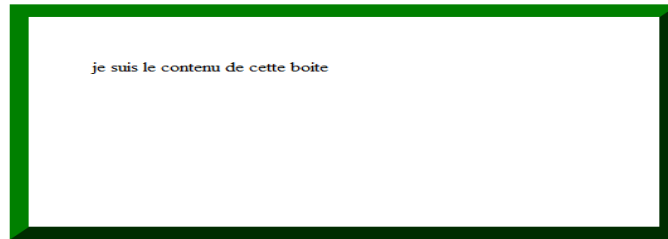
Pour les bordures on peut appliquer les valeurs suivantes pour la propriété border-style : inset ou outset

```
div
{
  border: solid 15px green ;
  border-style: inset;
  width:50%;
  margin:30px;
  padding: 50px;
  height:150px;
}
```


Résultat : propriété **inset** :



Propriété **outset** :



Résumé :

1. Largeur réelle occupée par le bloc :

margin-left + border-left + padding-left + width + padding-right + border-right + margin-right

2. Hauteur réelle occupée par le bloc :

margin-top + border-top + padding-top + height + padding-bottom + border-bottom + margin-bottom

Dépassement de contenu ou overflow :

La propriété **Overflow** d'une boîte permet de gérer le dépassement du contenu d'un élément dans son bloc.

Exemple :

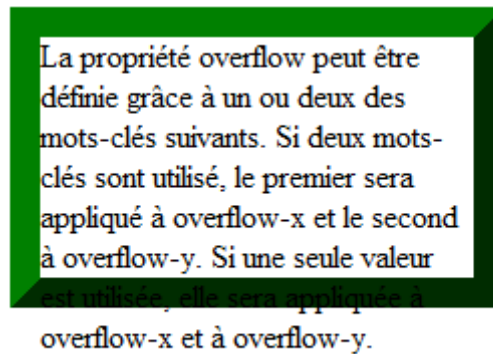
```
<div>
La propriété overflow peut être définie grâce à un ou deux
des mots-clés suivants. Si deux mots-clés sont utilisés, le
premier sera appliqué à overflow-x et le second à
overflow-y. Si une seule valeur est utilisée, elle sera
appliquée à overflow-x et à overflow-y.
</div>
```

Les propriétés de ce block sont :

```
div
{
```

```
border: solid 15px green ;
border-style: outset;
box-sizing: border-box;
width:30%;
height:150px;
}
```

Sur le navigateur le résultat est comme suit :



On a un déplacement du contenu de la boîte c'est vraiment le **OverFlow**.

Les valeurs de cette propriété sont : **visible hidden scroll auto**.

| | |
|---------|--|
| Visible |  |
| Hidden |  |
| Scroll |  |
| Auto |  |

Selon le contenu de la boîte

On peut appliquer aussi le déplacement de contenu selon les axes x ou y en utilisant les propriétés : **overflow-x** et **overflow-y**.

Remarque :

La valeur **inherit** d'une propriété permet d'hériter des valeurs de propriétés de l'élément HTML **parent** ;

Exemple :

Code HTML :

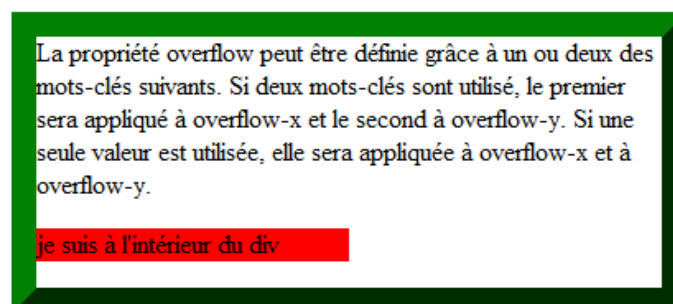
```
<div>
La propriété overflow peut être définie grâce à un ou deux des mots-clés
suivants. Si deux mots-clés sont utilisés, le premier sera appliqué à overflow-x
et le second à overflow-y. Si une seule valeur est utilisée, elle sera appliquée à
overflow-x et à overflow-y.
<p>
je suis à l'intérieur du div
</p>
</div>
```

Code CSS :

```
div
{
border: solid 15px green ;
border-style: outset;
box-sizing: border-box;
width:50%;
}
p
{
width: inherit;
background:red;
}
```

Le paragraphe **p** a 50% de la largeur de son parent qui est le **Div**

Le résultat est :



La valeur **important** définit la priorité d'une valeur par rapport à une valeur de la même propriété appliquée par la norme W3C.

Selon la norme W3C la valeur d'une propriété d'un sélecteur d'identifiant a la priorité de celle des sélecteurs de type ou de classes.

Exemple :

Code HTML :

```
<p>Je suis le pargarphe 1</p>
<br/>
<p id="p1">Je suis le pargarphe 2</p>
```

Code CSS3 :

```
p
{
  background:red;
}
#p1
{
  background:green;
}
```

Le résultat est :

Je suis le pargarphe 1

Je suis le pargarphe 2

La couleur verte revient à que le sélecteur d'identifiant a la priorité par rapport au selecteur de type.

Modifiant le code Css3 par :

```
p
{
  background:red !important;
}
```

Le résultat devient :

Je suis le pargarphe 1

Je suis le pargarphe 2

Exemples d'application : TP2-TP3

Les feuilles de style des images en CSS3.

- Insérer une image en utilisant le langage HTML :

```

```

Parmi les propriétés de l'image :

```
width: 150px;
height:150px;
border: solid 15px green;
border-style: inset;
padding:15px;
border-radius: 25px;
box-shadow: 8px 8px 3px 1px #778014;
```

- Insérer une image en utilisant le langage CSS3 :
Soit la boîte div qui doit être non vide:

```
div
{
  /* les propriétés du div */
  width:      350px;
  height:     350px;
  background:  green;
  text-align:  center;
  color:      white;
  font-size:  25px;
  display:    table-cell;      /*voir plus loin*/
  vertical-align:  middle;
  /* Les propriétés de l'image */
  background-image:    url(images/Capture.PNG);
  background-repeat:   no-repeat;
  background-size:     70% 70% ;
  background-position: center;
}
```

Le résultat :



Pour créer un aperçu d'une image on peut utiliser le code HTML suivant :

```
<a href="images/Capture.PNG">

</a>
```

Et le code CSS3 :

```
img
{
width: 150px;
height:160px;
}
```

Remarque importante :

La propriété display permet de définir le type d’affichage d’un élément HTML ; parmi ses valeurs :

Block : permet de changer l’état d’un élément en inline en block ; dans ce cas on peut appliquer les propriétés des éléments en block sur ces éléments.

Exemple :

```
<p>
Adresse. Siège :<span id="sp1"> Intersection Route nationale n°11 et Route
BO 50, BP 40207 / 20 270 Route Nouaceur, Sidi Maârouf -
Casablanca</span>&nbsp;&nbsp;&nbsp;&nbsp;<span id="sp2">Téléphone:+212
(05) 22 63 44 44</span>&nbsp;&nbsp;&nbsp;&nbsp;<span id="sp3">E-mail.
DRIF@ofppt</span> ...
</p>
```

Le code CSS3 appliqué :

```
#sp1
{
color:red;
}
#sp2
{
color:green;
}
#sp3
{
color:blue;
}
p
{
background:yellow;
height: 70px;
width:650px;
line-height: 30px;
}
```

En appliquant les propriétés width et height sur les span, on remarque qu’on n’a pas de modification ou d’application des attributs sur la balise span qui est inline.

Maintenant, on change l’état des span en block ; en ajoutant la ligne css : display : block.

```
span
{
  display: block;
  width :150px ;
  height :60px ;
}
```

Le paragraphe devient :

Adresse. Siège :
 Intersection Route nationale n°11 et Route BO 50, BP 40207 / 20 270 Route Nouaceur, Sidi Maârouf -
 Casablanca

Téléphone: +212 (05) 22 63 44 44

E-mail. DRIF@ofppt

Inline : permet de changer l'état d'un élément en block en inline ; dans ce cas l'élément HTML en question perd certaines propriétés.

Exemple :

```
<p>Je suis le paragraphe 1</p>
<p>Je suis le paragraphe 2</p>
<p>Je suis le paragraphe 3</p>
<p>Je suis le paragraphe 4</p>
```

La balise `<p>` est en block ; il marque un retour à la ligne ; le résultat est :

Je suis le paragraphe 1

Je suis le paragraphe 2

Je suis le paragraphe 3

Je suis le paragraphe 4

Si on applique le code CSS suivant sur les paragraphes ci-dessous ; les paragraphes deviennent inline et perdent certaines de leurs mises en forme.

```
p
{
  display:inline;
}
```

inline-block : cette permet de rendre des balises inline en block ; ces balises auront les propriétés des balises en block ; de même les balises en block deviennent inline en gardant les propriétés des balises en block.

Exemple :

Des balises inline :

Adresse. Siège : Intersection Route nationale n°11 et Route BO 50, BP 40207 / 20 270 Route Nouaceur, Sidi Maârouf - Casablanca Téléphone:+212 (05) 22 63 44 44 E-mail. DRIF@ofppt ...

En appliquant la propriété display :

```
span
{
  display: inline-block;
  width:150px;
  height:30;
  background:red;
}
```

L'exécution devient :

Adresse. Siège : Intersection Route nationale n°11 et Route BO 50, BP 40207 / 20 270 Route Nouaceur, Sidi Maârouf - Casablanca Téléphone:+212 (05) 22 63 44 44 E-mail. DRIF@ofppt ...

De même pour les balises en block :

```
<p>Je suis le paragraphe 1</p>
<p>Je suis le paragraphe 2</p>
<p>Je suis le paragraphe 3</p>
<p>Je suis le paragraphe 4</p>
```

En appliquant la propriété display :

```
p
{
  display: inline-block;
  width:150px;
  height:30;
  background:red;
}
```

L'exécution devient :



table-cell : Permet de convertir une boîte d'un élément HTML en case de tableau ; dans ce cas on peut utiliser la propriété : **vertical-align** :

Exemple :

Code HTML :

```
<p>paragraphe 1</p>
```

Code Css :

```
p
{
  width:600px;
  height:80px;
  background: red;
  text-align:center;
  display: table-cell;
  vertical-align: middle;
}
```

Exemple d'application TP2-exercice 4

Les feuilles de style d'une liste non ordonnée :

Soit la liste à puces suivante :

```
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JAVASCRIPT</li>
  <li>BOOTSRAP</li>
</ul>
```

Les éléments HTML **ul** et **li** sont des éléments en block.

Tester avec le code css suivant :

```
ul
{
  background: #AACC1E;
}
li
{
  background: #2ACACC;
}
```

Parmi les propriétés de la liste non ordonnée :

list-style-type: permet de définir le type de puce de la liste non ordonnée ; ses valeurs sont : **none circle square disc**

list-style-image: permet de définir une image comme puce de la liste; sa valeur est :

url(images/fleche.PNG)

list-style-position: permet de décaler le texte par rapport au puce de la liste ses valeurs sont **inside** ou **outside**. Sa valeur par défaut est **outside**;

```
<ul>
  <li>HTML: c'est un langage qui permet de définir la structure d'un
document HTML</li>
  <li>CSS: c'est un langage qui permet de définir la mise en forme des
éléments d'un document HTML"</li>
  <li>JAVASCRIPT</li>
  <li>BOOTSRAP</li>
</ul>
```

list-style-position: outside ou par défaut: le résultat sera selon le code CSS suivant:

```
ul
{
  background: #AACC1E;
```

```
width:200px;
list-style-position:outside;
}
```

Le résultat est :

- HTML: c'est un langage qui permet de définir la structure d'un document HTML
- CSS: c'est un langage qui permet de définir la mise en forme des éléments d'un document HTML"
- JAVASCRIPT
- BOOTSRAP

list-style-position: inside :

- HTML: c'est un langage qui permet de définir la structure d'un document HTML
- CSS: c'est un langage qui permet de définir la mise en forme des éléments d'un document HTML"
- JAVASCRIPT
- BOOTSRAP

Exercice d'application : créer un Menu Vertical et le convertir en Menu Horizontal.

Exemple de menu vertical :

Code HTML :

```
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JAVASCRIPT</li>
  <li>BOOTSRAP</li>
</ul>
```

Code CSS :

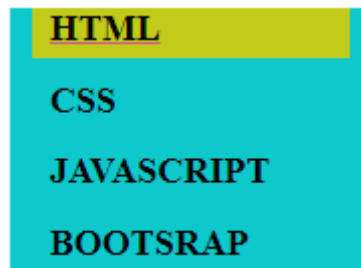
```
ul
{
  list-style: none;
  background: #0EC9CC;
  width: 180px;
  padding:0px;
  margin:0px;
}
li
{
  margin:6%;
  height:25px;
```

```

    cursor:pointer;
    font-size:18px;
    font-weight:bold;
    padding-left:5%;
}
li:hover
{
    background: #C1CC1A;
    text-decoration: underline solid #CC5496;
}

```

Le résultat est :



Menu Horizontal :

```

ul
{
    list-style: none;
    background: #0EC9CC;
    padding:0px;
    margin:0px;
}
li
{
    cursor:pointer;
    font-size:18px;
    font-weight:bold;
    display:inline-block;
    padding-left:2%;
    padding-right:2%;
    height:50px;
    padding-top:2%;
}
li:hover
{
    background: #C1CC1A;
    text-decoration: underline solid #CC5496;
}

```

HTML**CSS****JAVASCRIPT****BOOTSTRAP****Exemples d'application TP5.****Les feuilles de style d'un tableau.**

Les balises d'un tableau sont :

Balises élémentaires

- <table> : crée un tableau
- <caption> : légende du tableau (optionnel)
- <tr> : ligne du tableau
- <th> : cellules d'en-tête du tableau
- <td> : cellules du tableau

Balises complémentaires

- <thead> : groupe de cellules d'en-tête
- <tbody> : groupe de cellules de corps de tableau
- <tfoot> : groupe de cellules de pied de tableau

Pour appliquer un style d'un élément on utilise le tag de cet élément.

| Balise HTML | Boîte | | | | | Fond | Typo- graphie | Alignement |
|---|-------|--------------------|--------------------|--------------------|---------|--------|------------------|------------|
| | Cadre | Marges externes | Marges internes | Largeur | Hauteur | | | |
| TABLE | Oui | Oui | | Oui | Oui | Oui | Oui | Oui |
| CAPTION | Oui | Ne pas utiliser | Oui | Ne pas utiliser | Oui | Oui | Oui | Oui |
| COLGROUP | | | Pas NS | Oui | | Pas NS | Oui | Pas NS |
| COL | | | Pas NS | Oui | | Pas NS | Oui | Pas NS |
| THEAD, TBODY, TFOOT | | | | | | Oui | Oui | Oui |
| TR | | | | | Oui | Oui | Oui | Oui* |
| TH, TD | Oui | | Oui | Oui | Oui | Oui | Oui | Oui |
| * l'alignement horizontal ne fonctionne pas avec NS | | | | | | | | |

Exemple1 : Parmi les propriétés de la balise <table> :

Le résultat :

| Code | Intitle | MasseH |
|------|---------|--------|
| a1 | b1 | 20 |
| a2 | b2 | 25 |
| a3 | b3 | 60 |

Exemple2 : Parmi les propriétés de la balise <th> <td> :

```
th,td
{
border: solid 5px red;
background: #B8CC1A;
padding: 05px;
text-align: center;
}
```

Le résultat est :

| Code | Intitle | MasseH |
|------|---------|--------|
| a1 | b1 | 20 |
| a2 | b2 | 25 |
| a3 | b3 | 60 |

Pour les balises <th> et <td> on peut appliquer la propriété : **vertical-align** pour aligner le contenu des cases verticalement ; ces valeurs sont : **top baseline sub super top text-top middle middle bottom text-bottom**

Exemple :

```
th
{
vertical-align: middle ;
}
```

La propriété **border-collapse** détermine si les bordures d'un tableau sont séparées ou fusionnées. Ces valeurs sont : **collapse separate**

Pour fusionner les bordures du tableau ; on ajoute la ligne à la balise <table> :

```
table
{
border: solid 5px red;
width: 50%;
margin: 0px auto; /* V x H centrer horizontalement le tableau */
}
```

```
border-collapse: collapse separate;
}
```

Remarque : on peut mettre des images en arrière-plan des case du tableau (utiliser `background-image:url(image.jpg);`);

Exemple d'application :

| | Débutants | Experts | Gourous |
|---------------|--|--|---------|
| Tristan Nitot | Introduction à l'accessibilité | | |
| | Interview de Chantal Laplanche, Responsable Accessibilité Web au Ministère de la Culture | | |
| Laurent Denis | Un exemple de feuille de style d'impression | | |
| | | Commenter le document de travail W3C XHTML 2 | |
| | Débutants | Experts | Gourous |

Réponse : Voir la page : https://openweb.eu.org/articles/tableaux_css

6) La mise en page d'un document HTML

Disposition d'un élément HTML en CSS3.

Le CSS permet de définir le comportement ou la disposition d'un élément HTML par rapport aux autres éléments HTML dans un document HTML;

Pour définir le comportement d'un élément HTML on utilise la propriété display ; cette propriété a les valeurs suivantes :

None : Permet de supprimer un élément HTML d'un document HTML : il ne laisse d'espace

Exemple :

```
p
{
  display:none;
}
```

Au contraire de la propriété **visibility** dont la valeur est **hidden** ; permet de masquer un élément HTML en gardant son espace.

Code HTML :

```
<h1>Je suis un paragraphe1</h1>
<div>Je suis un paragraphe2</div>
<h2>Je suis un paragraphe3</h2>
```

Code CSS3 :

```
div
{
  visibility:hidden;
}
```

Et

```
div
{
  display:none;
}
```

Inline : permet de rendre un élément en block en inline voir plus en haut.

```
span
{
  display: inline;
}
```

Block : permet de rendre un élément inline en en block voir plus en haut.

```
div
{
  display: block;
}
```

inline-block : permet de rendre un élément **inline** ou en block en **inline** qui auront l'application des propriétés des éléments en block. (Voir plus haut).

Table-cell : permet de convertir un élément HTML en case d'un tableau (Voir plus haut).

flex : permet de rendre les éléments d'un block flexibles ; en effet on peut les afficher de différentes manières.

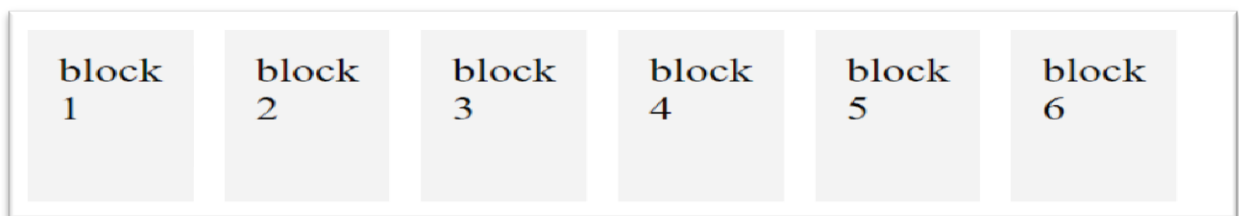
Exemple :

```
<div id="div1">
  <div>block 1</div>
  <div>block 2</div>
  <div>block 3</div>
  <div>block 4</div>
  <div>block 5</div>
  <div>block 6</div>
</div>
```

Le code Css3 :

```
#div1
{
  display: flex;
  width: 500px;
  height: 250px;
}
#div1 div
{
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
```

Le résultat :



Pour appliquer la flexibilité de ces éléments HTML, on peut ajouter des propriétés en relation avec flex;

```
#div1
{
```

```
display:flex;
FLEX-DIRECTION:row-reverse;
width:500px;
height:250px;
}
```

FLEX-DIRECTION qui aura les valeurs : **row row-reverse column column-reverse**.

FLEX-WRAP : qui va pouvoir donner la possibilité aux éléments flexibles d'aller à la ligne ou en colonne s'ils dépassent de leur conteneur. Ces valeurs sont : **nowrap wrap wrap-reverse**.

Pour plus de détails voir le site de test des valeurs de flexibilité : <https://flexbox.help/>

Exemples d'application : TP5

1. Le positionnement CSS3.

Le navigateur affiche les éléments HTML dans l'ordre du flux Normal ; il parcourt logiquement le document du début à la fin et affiche les éléments HTML élément par élément.

La propriété **position** permet de définir la position de ces éléments HTML selon ses valeurs qui sont : **static relative absolute fixed sticky**

Soir le code HTML :

```
<p id="p1">Paragraphe 1</p>
<p id="p2">Paragraphe 2</p>
<p id="p3">Paragraphe 3</p>
```

Code CSS3 :

```
p
{
width: 120px;
height:80px;
}
#p1
{
background: red;
}
#p2
{
background: green;
}
#p3
{
background: blue;
}
```

Static : Cette valeur est la valeur par défaut ; il intègre l'élément HTML dans le flux normal du document.

Exemple :

```
#p1
{
  background: red;
  position:static;
}
```

fixed : cette valeur permet de retirer un élément HTML du flux normal et la fixer dans un emplacement sans défiler avec le contenu du document ; pour définir la nouvelle position à partir du document, on utilise les propriétés **Top, bottom, right** et **left**.

```
#p1
{
  background: red;
  position:fixed;
  top:40px;
  left: 50px;
}
```

Relative : permet de positionner un élément HTML par rapport à sa position initiale sans influence sur les autres éléments HTML ; utilise les propriétés **Top, bottom, right** et **left** pour définir la nouvelle position de l'élément HTML par rapport à la position initiale ; les autres éléments restent sans influence du décalage de cet élément.

Code CSS3 :

```
#p1
{
  background: red;
  position:relative;
  top:40px;
  left: 50px; }
```

Le résultat est :



Absolute : cette propriété détache l'élément HTML du flux normal duc document et permet de déplacer cet élément dans n'importe quel endroit du document ; avec cette propriété l'élément perd la relation avec son **parent**.

```
#p1
{
  background: red;
```

```
position: absolute;
top: 50px;
left: 80px;
}
```

L'élément HTML prend le début de la page et peut être positionné par les propriétés **Top**, **bottom**, **right** et **left**.

Sticky : permet de décaler un élément HTML vers un emplacement dont la position est définie par les propriétés **Top**, **bottom**, **right** et **left**.

```
#p1
{
width: 100%;
background: red;
position: sticky;
top: 0px;
left: 80px;
}
```

Exemple d'application : TP4 exercice 4 et 5.

2. La position flottante.

Soit le code HTML :

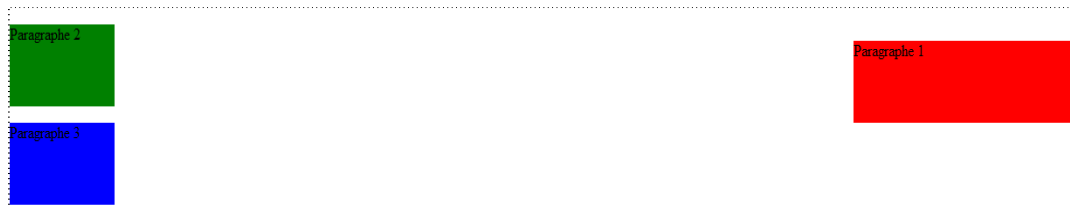
```
<p id="p1">Paragraphe 1</p>
<p id="p2">Paragraphe 2</p>
<p id="p3">Paragraphe 3</p>
```

La propriété **float** permet de détacher un élément HTML du flux normal et le déplacer à droite ou à gauche en utilisant ces valeurs **left** ou **right**.

Exemple :

```
#p1
{
width: 250px;
background: red;
float: right;
}
```

Le résultat est :

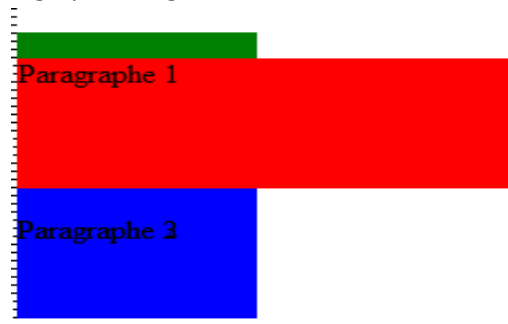


Et le code :

```
#p1
{
width: 250px;
background: red;
float: left;
}
```

}

Permet de mettre le paragraphe 1 à gauche du document :



Et pour déplacer cette boîte, on utilise les propriétés des marges.

Exemple :

```
#p1
{
  width:250px;
  background: red;
  float: left;
  margin-left:120px;
  margin-top:0px;
}
```

3. Mise en page d'un document HTML

Pour créer un document HTML on peut le décomposer en sections en utilisant des balises HTML5.

<header> : c'est la balise qui définit l'entête du document qui peut contenir des informations, logo, image, bannière Boite de connexion ou un formulaire de recherche.....

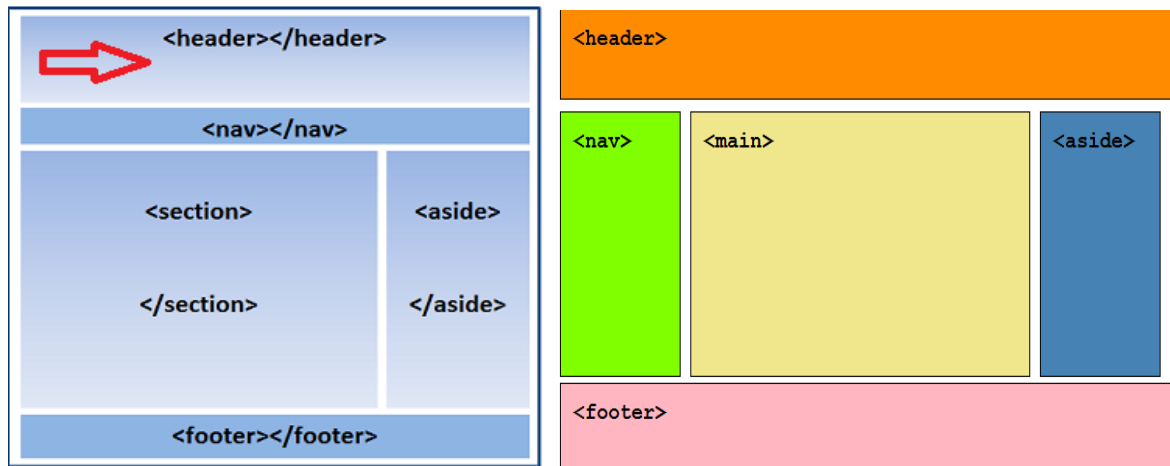
<nav> : c'est la partie qui est réservée au menu Horizontal ou vertical ; elle est destinée à la navigation entre les parties du document ou entre des pages web d'un site.

<section> : c'est la partie du document qui contient généralement le contenu de la page sous la forme des articles ou des thèmes <article>.

<aside> : c'est la partie à droite du document qui contient des informations indépendantes du contenu du site ou de la publicité.

<footer> : c'est la partie au pied du document HTML ; elle contient généralement des informations de l'auteur, la société,.....

Exemple :



Exemple :

Code HTML :

```
<header>
Entete
</header>
<nav>
Menu
</nav>
<section>
section
</section>
<aside>
aside
</aside>
<footer>
foot
</footer>
```

Code CSS3 :

```
header
{
background:blue;
height:80px;
}
nav
{
width: 20%;
background:green;
```

```

height:600px;
float:left;
}
section
{
width :60%;
background:yellow;
height:600px;
float:left;
}
aside
{
width:20%;
height:600px;
background:blue;
float:left;
}
footer
{
clear: left;
height:80px;
background: #292111;
}

```

Le résultat :



La propriété clear indique si un élément HTML peut être situé à côté de l'élément flottant ; ses valeurs sont : none ; left ; right, both !important.

Left : indique que l'élément sera déplacé vers le bas afin de dégager le flottement à gauche.

Right : déplace l'élément vers la droite de l'élément flottant.

Both : indique que l'élément est déplacé vers le bas afin de dégager les flottements à gauche et à droite.

Exemple : TP6 .

7) Les transformations, les transitions et les animations en CSS3

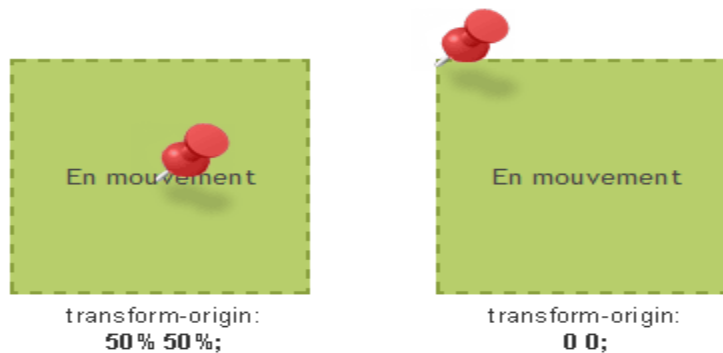
Les transformations en css3

La propriété **transform** permet de modifier les coordonnées d'un élément HTML : pivoter, incliner, mettre à l'échelle ou déplacer.

Pour en faire, en premier on définit le point d'origine de la transformation (épingle).



Le point d'origine est défini par les coordonnées de l'épingle dans un repère (0, x, y).



La fonction *scale()* :

Cette fonction permet d'augmenter la taille d'un élément HTML , en tenant la position du point d'origine.

Exemple :

Code HTML :

```
<p>Je suis un paragraphe en transformation</p>
```

Code CSS3 :

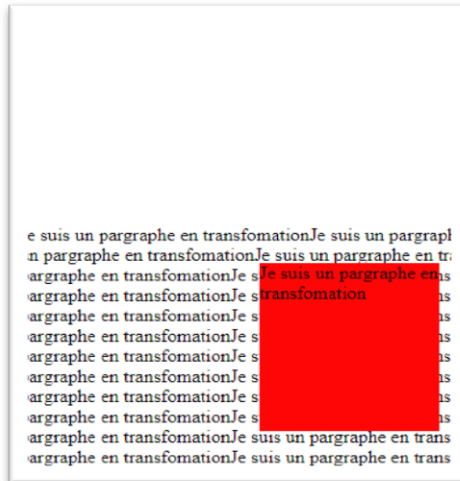
```
p
{
  height:150px;
  width:150px;
  background: red;
  transform-origin: 0% 0%;
}
p:hover
```

```
{
  transform: scale(1.5);
}
```

On peut utiliser les fonctions `scaleX()` et `scaleY()`.

La fonction `translate()`

Cette fonction permet d'effectuer une translation d'un élément HTML sur les axes X et Y ; l'élément en transformation est détaché du flux normal du document HTML en gardant son espace.



Code CSS :

```
p
{
  height:150px;
  width:150px;
  background: red;
}
p:hover
{
  transform: translate(200px, 200px);
}
```

Remarque : la translation peut être aussi sur l'axe des X ou sur l'axe des Y ; en utilisant les fonction `translateX()` ou `translateY()`

La fonction `rotate()` :

Cette fonction permet d'effectuer une rotation d'un élément HTML par rapport à une position d'origine, cette rotation s'exprime en degré et qui peut être négative ou positive.

Exemple :

Code CSS.

```
p
{
  height:150px;
  width:150px;
  background: red;
  transform-origin:50% 50%;
}
p:hover
{
  transform: rotate(45deg);
}
```

```
p:hover
{
  transform: rotatey(45deg) rotatex(50deg);
}
```

Remarque :

La rotation d'un élément HTML peut s'effectuer autour de l'axe X, ou Y ou Z.

La fonction skew()

Permet d'incliner un élément HTML en X ou Y .

Exemple :

```
p
{
  height:150px;
  width:150px;
  background: red;
}
p:hover
{
  transform:skewy(20deg);
}
```

Le résultat est:



Remarque : On peut utiliser les fonctions `skewX()`, `skewY()`.

Les transitions en css3

Une transition est une propriété CSS3 qui permet de modifier une valeur d'une propriété CSS3 ou plusieurs d'un élément HTML pendant une durée de temps.

La propriété transition regroupe les 4 autres propriétés suivantes :

- transition-property
- transition-duration
- transition-delay
- transition-timing-function.

transition-property :

Permet de définir la propriété sur laquelle on applique la transition.

```
p
{
  height:150px;
  width:150px;
  background: red;
  transition-property: width, height;
  transition-duration: 3s;
}
p:hover
{
  width:300px;
  height:600px;
}
```

Cette transition est appliquée sur les propriétés `width` et `height`.

```
transition-property: width, height;
```

transition-duration :

Cette propriété définit la durée de la transition qui en seconde.

```
transition-duration: 3s;
```

transition-delay :

permet de définir le temps de retard avant le déclenchement de l'événement .

```
transition-delay: 1s;
```

Après 1s du `hover` l'évènement se déclenche.

transition-timing-function :

Cette propriété permet de définir la cadence de la réalisation de la transition autre la vitesse de changement de transition au long de la durée ; les valeurs de cette propriété sont :

ease linear ease-in ease-out ease-in-out

Exemple :

```
transition-timing-function: ease ;
```

On peut regrouper les valeurs de la transition comme suit :

```
transition: width 3s, height 3s;
```

Les animations en css3 :

Une animation est un ensemble de transition pendant une durée de temps; cette animation est entre deux états de mise en forme d'un élément HTML.

Une animation est caractérisée par :

- animation-name
- animation-duration
- animation-delay
- animation-iteration-count
- animation-direction
- animation-timing-function
- animation-fill-mode
- animation

animation-name :

Permet de définir le nom de l'animation.

```
p
{
  width: 100px;
  height:100px;
  background: red;
  border-radius:50px;
  animation-name: exemple ;
  animation-duration: 5s;
}
```

animation-duration :

Permet de définir la durée de l'animation qui en seconde.

```
animation-duration: 5s;
```

Création de l'animation :

La directive ou la règle **@Keyframes** permet de définir les étapes de l'animation.

Exemple 1: déplacement d'une boule.

```
@Keyframes exemple
```

```

{
  from{
    margin-left: 200px;
  }
  to {
    margin-left: 500px;
  }
}

```

Exemple 2: Boule roulante.

```

p
{
  width: 100px;
  height:100px;
  background-image:linear-gradient(to bottom, #5CC3FF,
#BF51B4) ;
  border-radius:50px;
  transform-origin: 50% 50%;
  animation-name: exemple;
  animation-duration: 5s;
}
@Keyframes exemple
{
  from{
    margin-left: 200px;
    transform: rotate(0deg);
  }
  to {
    margin-left: 500px;
    transform: rotate(360deg);
  }
}
}

```

Exemple3 : Texte Animé.

```

p
{
  color: #28A3CC;
  font-size:30px;
  font-weight: bold;
  animation-name:exemple3;
  animation-duration: 5s;
}

```

```

}
@keyframes exemple3
{
  from{
    margin-left:0px;
  }
  to
  {
    margin-left:700px;
  }
}

```

Les autres propriétés de l'animation :

animation-delay :

C'est le temps de retard avant le déclenchement de l'animation en seconde.

```
animation-delay: 1s;
```

animation-iteration-count :

Définit le nombre d'itération de l'animation ; ses valeurs sont : **infinite** ou un nombre fixe d'itération.

```
animation-iteration-count : infinite;
```

animation-direction :

Définit la direction de l'animation ; ses valeurs sont : **normal alternate**

```
animation-direction: alternate;
```

animation-timing-function :

Définit la vitesse de changement de l'animation pendant la durée de temps. Ses valeurs sont : **ease linear ease-in ease-out ease-in-out**

animation-fill-mode :

Définit le style d'un élément HTML lorsque l'animation ne fonctionne pas (en avant et après). Ses valeurs sont :

Forwards : garder son style ou son état après l'animation.

Backwards : garder son style ou son état avant l'animation

Exemple :

```

p
{
  color: #28A3CC;
  font-size:30px;
  font-weight: bold;
  animation-name:exemple3;
  animation-duration: 5s;
  animation-delay: 1s;
  animation-fill-mode: backwards ;
}
@keyframes exemple3
{
  from{
    margin-left:0px;
    color:red;
  }
  to
  {
    margin-left:500px;
    color:blue;
  }
}

```

On peut utiliser la propriété animation qui aura la combinaison des valeurs pour appliquer une animation :

```

p
{
  color: #28A3CC;
  font-size:30px;
  font-weight: bold;
  animation: exemple3 5s infinite;
  animation-fill-mode: backwards ;
}

```

Remarque : Dans une animation, on peut utiliser le code sous la forme suivante :

```

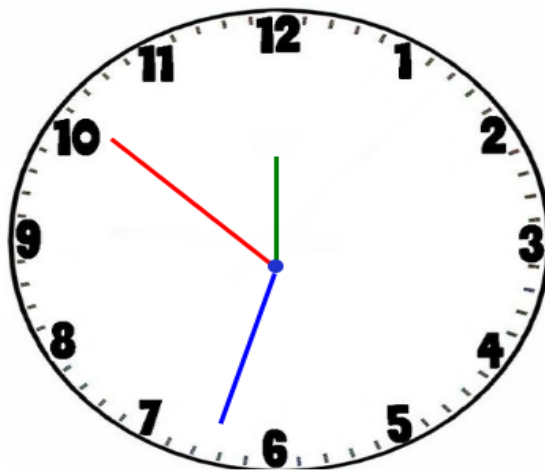
p
{
  width: 100px;
  height:100px;
  background-image:linear-gradient(to bottom, #5CC3FF,
#BF51B4) ;
  border-radius:50px;
  animation-name: exemple;
}

```



```
animation-duration: 15s;
}
@Keyframes exemple
{
  %0{
    margin-left: 200px;
    transform: rotate(0deg);
    margin-top:0px;
  }
  25% {
    margin-left: 500px;
    transform: rotate(360deg);
    margin-top:0px;
  }
  50%{
    margin-left: 500px;
    transform: rotate(360deg);
    margin-top:400px;
  }
  75%{
    margin-left: 0px;
    transform: rotate(360deg);
    margin-top:400px;
  }
}
```

Travail à faire : Créer une horloge analogique en utilisant seulement HTML5 et CSS3.



Bibliographies

<http://css.mammothland.net/feuille-de-style-css-debutant.php>

<https://www.commentcamarche.net/contents/229-declaration-d-une-feuille-de-style-css>

<https://yesyouweb.com/css3-guide->

<reference/#:~:text=La%20colonne%20CSS%20dans%20nos,%3A%20CSS1%2C%20CSS2%20ou%20CSS3.>

<https://developer.mozilla.org/fr/docs/Apprendre/CSS>

https://openweb.eu.org/articles/tableaux_css

<https://www.alsacreations.com/article/lire/1418-css3-transformations-2d.html>