

Support de cours

Cours complet de CSS

Réaliser par : Nene SYLLA

I. INTRODUCTION A CSS :

CSS (*Cascading Style Sheets*), c'est cet autre langage qui vient compléter le HTML.

A quoi sert le CSS ?

- ✓ C'est lui qui vous permet de choisir la couleur de votre texte.
- ✓ Lui qui vous permet de sélectionner la police utilisée sur votre site.
- ✓ Lui encore qui permet de définir la taille du texte, les bordures, le fond...
- ✓ Et aussi, c'est lui qui permet de faire la mise en page de votre site. Vous pourrez dire : je veux que mon menu soit à gauche et occupe telle largeur, que l'en-tête de mon site soit calé en haut et qu'il soit toujours visible, etc.

Où écrit-on le CSS ?

On peut écrire du CSS à 3 endroits différents, selon ce qu'on préfère :

- directement dans les balises du fichier **HTML** *via* un attribut **style** (*méthode la moins recommandée*) ;
- dans l'en-tête **<head>** du fichier HTML ;
- dans un fichier **.css** (méthode la plus recommandée).

1. Directement dans les balises (non recommandé)

Vous pouvez ajouter un attribut style à n'importe quelle balise. Vous insérez votre code CSS directement dans cet attribut:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Premiers tests du CSS</title>
  </head>
  <body>
    <h1>Mon super site</h1>
    <p style="color: blue;">Bonjour et bienvenue sur mon site !</p>
    <p>Pour le moment, mon site est un peu <em>vide</em>.
    Patientez encore un peu !</p>
  </body>
</html>
```

Ici, seul le texte du premier paragraphe, dont la balise contient le code CSS, sera coloré en bleu.

2. Dans l'en-tête <head> du fichier HTML

Il existe une autre méthode pour utiliser du CSS dans ses fichiers HTML : cela consiste à insérer le code CSS directement dans une balise <style> à l'intérieur de l'en-tête <head>.

Voici comment on peut obtenir exactement le même résultat avec un seul fichier .html qui contient le code CSS :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <style>
      p
      {
        color: blue;
      }
    </style>
    <title>Premiers tests du CSS</title>
  </head>
  <body>
    <h1>Mon super site</h1>
    <p>Bonjour et bienvenue sur mon site !</p>
    <p>Pour le moment, mon site est un peu <em>vide</em>.
    Patientez encore un peu !</p>
  </body>
</html>
```

3. Dans un fichier .css (recommandé)

On écrit le plus souvent le code CSS dans un fichier spécial ayant l'extension .css (contrairement aux fichiers HTML qui ont l'extension .html). C'est la méthode la

plus pratique et la plus souple. Cela nous évite de tout mélanger dans un même fichier.

Nous allons partir du fichier HTML suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Premiers tests du CSS</title>
  </head>
  <body>
    <h1>Mon super site</h1>
    <p>Bonjour et bienvenue sur mon site !</p>
    <p>Pour le moment, mon site est un peu <em>vide</em>.
    Patientez encore un peu !</p>
  </body>
</html>
```

<link rel="stylesheet" href="style.css" /> : c'est elle qui indique que ce fichier HTML est associé à un fichier appelé style.css et chargé de la mise en forme.

Enregistrez ce fichier sous le nom que vous voulez (par exemple page.html). Pour le moment, rien d'extraordinaire à part la nouvelle balise que nous avons ajoutée. Maintenant, créez un *nouveau* fichier vide dans votre éditeur de texte et copiez-y ce bout de code CSS :

```
P
{
  color: blue;
}
```

Enregistrez le fichier en lui donnant un nom qui se termine par .css, comme style.css. Placez ce fichier .css dans le même dossier que votre fichier .html.

Il est inutile d'ouvrir directement le fichier style.css dans le navigateur. Il faut ouvrir le fichier page.html (il fera automatiquement appel au fichier style.css).

SYNTAXE GÉNÉRALE :

```
Nom_de_balises  
{  
    Propriété : valeur ;  
}
```

EXPLICATION :

Dans un code CSS comme celui-ci, on trouve trois éléments différents :

- **Des noms de balises** : on écrit les noms des balises dont on veut modifier l'apparence. Par exemple, pour modifier l'apparence de tous les paragraphes <p>, nous devons écrire p.
- **Des propriétés CSS** : les « effets de style » de la page sont rangés dans des propriétés. Il y a par exemple la propriété color qui permet d'indiquer la couleur du texte, font-size qui permet d'indiquer la taille du texte, etc. Il y a beaucoup de propriétés CSS.
- **Les valeurs** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété color, il faut indiquer le nom de la couleur. Pour font-size, il faut indiquer quelle taille on veut, etc.

Des commentaires dans du CSS

Comme en HTML, il est possible de mettre des commentaires. Les commentaires ne seront pas affichés, ils servent simplement à indiquer des informations pour vous.

D'ailleurs, vous allez vous en rendre compte, en général le fichier HTML est assez court et la feuille CSS assez longue (si elle contient tous les éléments de style de votre site, c'est un peu normal). Notez qu'il est possible de créer plusieurs fichiers CSS pour votre site si vous ressentez le besoin de séparer un peu votre code CSS (en fonction des différentes sections de votre site, par exemple).

Pour faire un commentaire, c'est facile ! Tapez `/*`, suivi de votre commentaire, puis `*/` pour terminer votre commentaire.

Vos commentaires peuvent être écrits sur une ou plusieurs lignes.

II. LES ATTRIBUTS CLASS ET ID :

Ce sont des attributs HTML qui ont été créés pour donner différents styles à des éléments de même type, et également le même style à différents éléments choisis. Les attributs class et id sont quasiment identiques.

- **L'attribut class :**

Si vous utilisez des class, lorsque vous définirez leurs propriétés dans le fichier CSS, il faudra faire précéder le nom de l'id par un point (.) :

Soit l'exemple :

Code HTML :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Premiers tests du CSS</title>
  </head>
  <body>
    <h1>Mon super site</h1>
    <p class="introduction">Bonjour et bienvenue sur mon site !</p>
    <p>Pour le moment, mon site est un peu <em>vide</em>.
    Patientez encore un peu !</p>
  </body>
</html>
```

Code CSS :

```
.introduction
{
  color: blue;
}
```

- **L'attribut id :**

L'attribut id fonctionne exactement de la même manière que class, à un détail près: il ne peut être utilisé *qu'une fois* dans le code. L'utilisation de l'attribut id vous sera utile si vous faites du JavaScript plus tard pour reconnaître certaines balises.

En pratique, nous ne mettrons des id que sur des éléments qui sont uniques dans la page, comme par exemple le logo :

```

```

Si vous utilisez des id, lorsque vous définirez leurs propriétés dans le fichier CSS, il faudra faire précéder le nom de l'id par un dièse (#) :

```
#logo
{
    /* Indiquez les propriétés CSS ici */
}
```

- **Le sélecteur universel :**

Un sélecteur universel sélectionne toutes les balises sans exceptions. Elle est caractérisée par un **ASTÉRIX(*)**.

```
*{
  
}
```

EN RÉSUMÉ :

- CSS est un autre langage qui vient compléter le HTML. Son rôle est de mettre en forme votre page web.
- Il faut être vigilant sur la compatibilité des navigateurs avec certaines fonctionnalités récentes de CSS3. Quand un navigateur ne connaît pas une instruction de mise en forme, il l'ignore simplement.

- On peut écrire le code CSS à plusieurs endroits différents, le plus conseillé étant de créer un fichier séparé portant l'extension .css (exemple : style.css).
- En CSS, on sélectionne quelles portions de la page HTML on veut modifier et on change leur présentation avec des propriétés CSS :

Code CSS :

```
balise1
{
  propriete1: valeur1;
  propriete2: valeur2;
}
```

- Il existe de nombreuses façons de sélectionner la portion de la page que l'on veut mettre en forme. Par exemple, on peut viser :
 - toutes les balises d'un même type, en écrivant simplement leur nom (h1 par exemple) ;
 - certaines balises spécifiques, auxquelles on a donné des noms à l'aide des attributs class ou id (.nomclasse ou #nomid) ;

III. FORMATAGE DU TEXTE :

Le formatage du texte en CSS ne veut pas dire destruction de toutes les données, mais plutôt signifie la modification de l'apparence du texte.

La taille

Pour modifier la taille du texte, on utilise la propriété CSS **font-size**. Mais comment indiquer la taille du texte ? Plusieurs techniques vous sont proposées :

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

1. Une taille absolue :

Pour indiquer une taille absolue, on utilise généralement les pixels. Pour avoir un texte de 16 pixels de hauteur, vous devez donc écrire :

```
font-size: 16px;
```

EXEMPLE :

```
P
{
    font-size: 14px; /* Paragraphes de 14 pixels */
}
h1
{
    font-size: 40px; /* Titres de 40 pixels */
}
```

2. Une valeur relative :

C'est la méthode recommandée car le texte s'adapte alors plus facilement aux préférences de tous les visiteurs.

Il y a plusieurs moyens d'indiquer une valeur relative. Vous pouvez par exemple écrire la taille avec des mots en anglais comme ceux-ci :

- xx-small : minuscule ;
- x-small : très petit ;
- small : petit ;
- medium : moyen ;
- large : grand ;
- x-large : très grand ;
- xx-large : euh... gigantesque.

Vous pouvez tester l'utilisation de ces valeurs dans votre code CSS :

```
P
{
    font-size: small;
```

```
}  
h1  
{  
    font-size: large;  
}
```

La police

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

La bonne nouvelle, c'est que depuis CSS3, il est possible de faire télécharger automatiquement une police au navigateur.

1. Modifier la police utilisée :

La propriété CSS qui permet d'indiquer la police à utiliser est **font-family**.

SYNTAXE :

```
balise  
{  
    font-family: police;  
}
```

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

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

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

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

Il existe aussi une autre police par défaut appelée sans-serif. La différence entre les deux est la présence de petites pattes de liaison en bas des lettres, que la police sans-serif n'a pas.

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

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

EXEMPLE :

```
P
{
    font-family: Impact, "Arial Black", Arial, Verdana, sans-serif;
}
```

Italique, gras, souligné...

Il existe en CSS une série de propriétés classiques de mise en forme du texte.

1. Mettre en italique :

****, est faite pour insister sur des mots. Cela veut dire que les mots qu'elle entoure sont assez importants.

Pour représenter cette importance, la plupart des navigateurs choisissent d'afficher le texte en italique, mais ce n'est pas une obligation.

Le CSS lui, permet de dire réellement : « Je veux que ce texte soit en italique ». Rien ne vous empêche, par exemple, de décider que tous vos titres seront en italique.

Concrètement, en CSS, pour mettre en italique, on utilise `font-style` qui peut prendre trois valeurs :

- ***italic*** : le texte sera mis en italique.
- ***oblique*** : le texte sera passé en oblique (les lettres sont penchées, le résultat est légèrement différent de l'italique proprement dit).
- ***normal*** : le texte sera normal (par défaut). Cela vous permet d'annuler une mise en italique. Par exemple, si vous voulez que les textes entre `` ne soient plus en italique, vous devrez écrire :

```
em
{
  font-style: normal;
}
```

Ainsi, dans l'exemple suivant, je me sers de `font-style` pour mettre en italique tous mes titres `<h2>` :

```
h2
{
  font-style: italic;
}
```

2. Mettre en gras :

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

- ***bold*** : le texte sera en gras ;
- ***normal*** : le texte sera écrit normalement (par défaut).

Voici par exemple comment écrire les titres en gras :

```
h1
{
  font-weight: bold;
}
```

3. *Soulignement et autres décorations :*

La propriété CSS associée porte bien son nom : text-decoration. Elle permet, entre autres, de souligner le texte, mais pas seulement. Voici les différentes valeurs qu'elle peut prendre :

- underline : souligné.
- line-through : barré.
- overline : ligne au-dessus.
- blink : clignotant. Ne fonctionne pas sur tous les navigateurs (Internet Explorer et Google Chrome, notamment).
- none : normal (par défaut).

Ce CSS va vous permettre de tester les effets de text-decoration :

```
h1
{
    text-decoration: blink;
}
.souligne
{
    text-decoration: underline;
}
.barre
{
    text-decoration: line-through;
}.
.ligne_dessus
{
    text-decoration: overline;
}
```

L'alignement

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

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

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

Regardez les différents alignements sur cet exemple :

```
h1
{
  text-align: center;
}
p
{
  text-align: justify;
}
.signature
{
  text-align: right;
}
```

Les flottants

Le CSS nous permet de faire flotter un élément autour du texte. On dit aussi qu'on fait un « habillage ».

float (« flottant » en anglais). Cette propriété peut prendre deux valeurs très simples :

- left : l'élément flottera à gauche.
- right : l'élément flottera à droite !

L'utilisation des flottants est très simple :

- Vous appliquez un **float** à une balise.
- Puis vous continuez à écrire du texte à la suite normalement.

EXEMPLE :

```
.imageflottante
```

```
{  
  float: left;  
}
```

La couleur et le fond

1. Couleur du texte:

La propriété qui permet de modifier la couleur est **color**.

a. Indiquer le nom de la couleur :

La méthode la plus simple et la plus pratique pour choisir une couleur consiste à taper son nom en anglais.

Le seul défaut de cette méthode est qu'il n'existe que seize couleurs dites « standard ». D'autres couleurs officielles existent mais, comme elles ne fonctionneront pas forcément de la même manière sur tous les navigateurs.

EXEMPLE :

```
h1  
{  
  color : maroon ;  
}
```

b. La notation hexadécimale :

Seize couleurs, c'est quand même un peu limite quand on sait que la plupart des écrans peuvent en afficher seize *millions*.

Heureusement, il existe en CSS plusieurs façons de choisir une couleur parmi toutes celles qui existent. La première que je vais vous montrer est la notation hexadécimale. Elle est couramment utilisée sur le Web mais il existe aussi une autre méthode que nous verrons plus loin.

Un nom de couleur en hexadécimal, cela ressemble à : #FF5A28. Pour faire simple, c'est une combinaison de lettres et de chiffres qui indiquent une couleur.

On doit toujours commencer par écrire un dièse (#), suivi de six lettres ou chiffres allant de 0 à 9 et de A à F.

Ces lettres ou chiffres fonctionnent deux par deux. Les deux premiers indiquent une quantité de rouge, les deux suivants une quantité de vert et les deux derniers une quantité de bleu. En mélangeant ces quantités (qui sont les composantes Rouge-Vert- Bleu de la couleur) on peut obtenir la couleur qu'on veut.

Ainsi, #000000 correspond à la couleur noire et #FFFFFF à la couleur blanche.

Voici par exemple comment on fait pour appliquer aux paragraphes la couleur blanche en hexadécimal :

```
P
{
  color: #FFFFFF;
}
```

c. La méthode RGB :

Que signifie RGB ? En anglais, Rouge-Vert-Bleu s'écrit Red-Green-Blue, ce qui s'abrège en « RGB ». Comme avec la notation hexadécimale, pour choisir une couleur, on doit définir une quantité de rouge, de vert et de bleu.

Voici la marche à suivre pour utiliser la méthode RGB :

1. Lancez le logiciel Paint depuis le menu Démarrer.
2. Rendez-vous dans la section Modifier les couleurs, comme indiqué à la figure suivante.
3. Une fenêtre s'ouvre. Dans la zone qui apparaît à droite, faites bouger les curseurs pour sélectionner la couleur qui vous intéresse. Supposons que vous soyez pris d'une envie folle d'écrire vos titres <h1> en rose bonbon (supposons seulement). Sélectionnez la couleur dans la fenêtre, comme à la figure suivante.
4. Relevez les quantités de Rouge-Vert-Bleu correspondantes, indiquées en bas à droite de la fenêtre (ici 240-96-204).

Recopiez ces valeurs dans cet ordre dans le fichier CSS, comme dans le code ci-dessous :

```
P
{
  color: rgb(240,96,204);
}
```



```
}
```

2. Couleur de fond :

Pour indiquer une couleur de fond, on utilise la propriété CSS **background-color**. Elle s'utilise de la même manière que la propriété **color**, c'est-à-dire que vous pouvez taper le nom d'une couleur, l'écrire en notation hexadécimale ou encore utiliser la méthode RGB.

Pour indiquer la couleur de fond de la page web, il faut travailler sur la balise **<body>**. Eh oui, **<body>** correspond à l'ensemble de la page web, c'est donc en modifiant sa couleur de fond que l'on changera la couleur d'arrière-plan de la page.

EXEMPLE :

```
/* On travaille sur la balise body, donc sur TOUTE la page */  
body  
{  
    background-color: black; /* Le fond de la page sera noir */  
    color: white; /* Le texte de la page sera blanc */  
}
```

3. Images de fond :

Tout comme pour la couleur de fond, n'oubliez pas que l'image de fond ne s'applique pas forcément à la page entière. On peut aussi mettre une image de fond derrière les titres, paragraphes, etc.

Appliquer une image de fond :

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

EXEMPLE :

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

NB : Votre fond n'est pas forcément en PNG, il peut aussi être en JPEG ou en GIF.

Options disponibles pour l'image de fond :

On peut compléter la propriété **background-image** que nous venons de voir par plusieurs autres propriétés qui permettent de changer le comportement de l'image de fond :

a. background-attachment : fixer le fond

La propriété CSS **background-attachment** permet de « fixer » le fond. L'effet obtenu est intéressant car on voit alors le texte « glisser » par-dessus le fond. Deux valeurs sont disponibles :

- **fixed** : l'image de fond reste fixe ;
- **scroll** : l'image de fond défile avec le texte (par défaut).

EXEMPLE :

```
body
{
    background-image: url("neige.png");
    background-attachment: fixed; /* Le fond restera fixe */
}
```

b. background-repeat : répétition du fond

Par défaut, l'image de fond est répétée en mosaïque. Vous pouvez changer cela avec la propriété **background-repeat** :

- **no-repeat** : le fond ne sera pas répété. L'image sera donc unique sur la page.
- **repeat-x** : le fond sera répété uniquement sur la première ligne, horizontalement.
- **repeat-y** : le fond sera répété uniquement sur la première colonne, verticalement.
- **repeat** : le fond sera répété en mosaïque (par défaut).

EXEMPLE :

```
body
```

```
{  
  background-image: url("soleil.png");  
  background-repeat: no-repeat;  
}
```

c. background-position : position du fond

On peut indiquer où doit se trouver l'image de fond avec background-position. Cette propriété n'est intéressante que si elle est combinée avec background-repeat: no-repeat; (un fond qui ne se répète pas).

Vous devez donner à background-position deux valeurs en pixels pour indiquer la position du fond par rapport au coin supérieur gauche de la page (ou du paragraphe, si vous appliquez le fond à un paragraphe). Ainsi, si vous tapez :

```
background-position: 30px 50px;
```

Votre fond sera placé à 30 pixels de la gauche et à 50 pixels du haut. Il est aussi possible d'utiliser ces valeurs en anglais :

- top : en haut ;
- bottom : en bas ;
- left : à gauche ;
- center : centré ;
- right : à droite.

Il est possible de combiner ces mots. Par exemple, pour aligner une image en haut à droite, vous taperez :

```
background-position: top right;
```

```
body  
{  
  background-image: url("soleil.png");  
  background-attachment: fixed; /* Le fond restera fixe */  
  background-repeat: no-repeat; /* Le fond ne sera pas répété */
```

```
background-position: top right; /* Le fond sera placé en haut à droite */
}
```

4. Combiner les propriétés :

Si vous utilisez beaucoup de propriétés en rapport avec le fond (comme c'est le cas sur ce dernier exemple), vous pouvez utiliser une sorte de « super-propriété » appelée `background` dont la valeur peut combiner plusieurs des propriétés vues précédemment : `background-image`, `background-repeat`, `background-attachment` et `background-position`.

On peut donc tout simplement écrire :

```
body
{
    background: url("soleil.png") fixed no-repeat top right;
}
```

5. Plusieurs images de fond :

Depuis CSS3, il est possible de donner plusieurs images de fond à un élément. Pour cela, il suffit de séparer les déclarations par une virgule, comme ceci :

```
body
{
    background: url("soleil.png") fixed no-repeat top right,
    url("neige.png") fixed;
}
```

La transparence

Le CSS nous permet de jouer très facilement avec les niveaux de transparence des éléments ! Pour cela, nous allons utiliser des fonctionnalités de CSS3 : la propriété `opacity` et la notation `RGBA`.

a. La propriété *opacity* :

La propriété *opacity*, très simple, permet d'indiquer le niveau d'opacité (c'est l'inverse de la transparence).

- Avec une valeur de 1, l'élément sera totalement opaque : c'est le comportement par défaut.
- Avec une valeur de 0, l'élément sera totalement transparent.

Il faut donc choisir une valeur comprise entre 0 et 1. Ainsi, avec une valeur de 0.6, votre élément sera opaque à 60%... et on verra donc à travers !

Voici comment on peut l'utiliser :

```
P
{
  opacity: 0.6;
}
```

Voici un exemple qui va nous permettre d'apprécier la transparence :

```
body
{
  background: url('neige.png');
}
P
{
  background-color: black;
  color: white;
  opacity: 0.3;
}
```

NB : Si vous appliquez la propriété *opacity* à un élément de la page, tout le contenu de cet élément sera rendu transparent (même les images, les autres blocs à l'intérieur, etc.).

EN RÉSUMÉ :

- On change la couleur du texte avec la propriété `color`, la couleur de fond avec `background-color`.
- On peut indiquer une couleur en écrivant son nom en anglais (`black`, par exemple), sous forme hexadécimale (`#FFC8D3`) ou en notation RGB (`rgb(250,25,118)`).
- On peut ajouter une image de fond avec `background-image`. On peut choisir de fixer l'image de fond, de l'afficher en mosaïque ou non, et même de la positionner où on veut sur la page.
- On peut rendre une portion de la page transparente avec la propriété `opacity`.

Les bordures et les ombres

Ici, nous allons nous intéresser aux bordures et aux effets d'ombrage que l'on peut appliquer, aussi bien sur le texte que sur les blocs qui constituent notre page.

a. Bordures standard :

Le CSS vous offre un large choix de bordures pour décorer votre page. De nombreuses propriétés CSS vous permettent de modifier l'apparence de vos bordures : `border-width`, `border-color`, `border-style`...

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

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

- inset : effet 3D global enfoncé ;
- outset : effet 3D global surélevé.

EXEMPLE :

```
h1
{
  border: 3px blue dashed;
}
```

En haut, à droite, à gauche, en bas... :

Dans ce cas, vous devrez utiliser ces quatre propriétés :

- border-top : bordure du haut ;
- border-bottom : bordure du bas ;
- border-left : bordure de gauche ;
- border-right : bordure de droite.

Il existe aussi des équivalents pour paramétrer chaque détail de la bordure si vous le désirez : border-top-width pour modifier l'épaisseur de la bordure du haut, border-top-color pour la couleur du haut, etc.

Ce sont aussi des super-propriétés, elles fonctionnent comme border mais ne s'appliquent donc qu'à un seul côté.

Pour ajouter une bordure uniquement à gauche et à droite des paragraphes, on écrira donc :

```
P
{
  border-left: 2px solid black;
  border-right: 2px solid black;
}
```

On peut modifier les bordures de n'importe quel type d'élément sur la page. Nous l'avons fait ici sur les paragraphes mais on peut aussi modifier la bordure des images, des textes importants comme , etc.

b. Bordures arrondies :

Les bordures arrondies, c'est un peu le Saint Graal attendu par les webmasters depuis des millénaires (ou presque). Depuis que CSS3 est arrivé, il est enfin possible d'en créer facilement !

La propriété `border-radius` va nous permettre d'arrondir facilement les angles de n'importe quel élément. Il suffit d'indiquer la taille (« l'importance ») de l'arrondi en pixels :

```
P
{
    border-radius: 10px;
}
```

On peut aussi préciser la forme de l'arrondi pour chaque coin. Dans ce cas, indiquez quatre valeurs :

```
P
{
    border-radius: 10px 5px 10px 5px;
}
```

Les valeurs correspondent aux angles suivants dans cet ordre :

- en haut à gauche ;
- en haut à droite ;
- en bas à droite ;
- en bas à gauche.

Enfin, il est possible d'affiner l'arrondi de nos angles en créant des courbes elliptiques (figure suivante). Dans ce cas, il faut indiquer deux valeurs séparées par une barre oblique (slash, caractère /). Le mieux est certainement de tester pour voir l'effet :

```
P
{
```



```
border-radius: 20px / 10px;  
}
```

Les ombres

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

Nous allons ici découvrir deux types d'ombres :

- les ombres des boîtes ;
- les ombres du texte.

1. *box-shadow* : les ombres des boîtes

La propriété **box-shadow** s'applique à tout le bloc et prend quatre valeurs dans l'ordre suivant :

- le décalage horizontal de l'ombre ;
- le décalage vertical de l'ombre ;
- l'adoucissement du dégradé ;
- la couleur de l'ombre.

Par exemple, pour une ombre noire de 6 pixels, sans adoucissement, on écrira :

```
P  
{  
    box-shadow: 6px 6px 0px black;  
}
```

On peut aussi rajouter une cinquième valeur facultative : **inset**. Dans ce cas, l'ombre sera placée à l'intérieur du bloc, pour donner un effet enfoncé :

```
P  
{  
    box-shadow: 6px 6px 6px black inset;  
}
```

NB : La propriété box-shadow fonctionne sur tous les navigateurs récents, IE9 inclus. Pour certains navigateurs, en particulier les navigateurs mobiles, il faut encore rajouter un préfixe. Ainsi, il faudra écrire une version -webkitbox-shadow pour que cela fonctionne sur les navigateurs Android et iOS.

2. text-shadow : l'ombre du texte

Avec text-shadow, vous pouvez ajouter une ombre directement sur les lettres de votre texte ! Les valeurs fonctionnent exactement de la même façon que box-shadow : décalage, adoucissement et couleur.

```
p
{
    text-shadow: 2px 2px 4px black;
}
```

EN RÉSUMÉ :

- On peut appliquer une bordure à un élément avec la propriété border. Il faut indiquer la largeur de la bordure, sa couleur et son type (trait continu, pointillés...).
- On peut arrondir les bordures avec border-radius.
- On peut ajouter une ombre aux blocs de texte avec box-shadow. On doit indiquer le décalage vertical et horizontal de l'ombre, son niveau d'adoucissement et sa couleur.
- Le texte peut lui aussi avoir une ombre avec text-shadow.

Création d'apparences dynamiques

C'est une de ses forces : le CSS nous permet aussi de modifier l'apparence des éléments de façon dynamique, c'est-à-dire que des éléments peuvent changer de forme une fois que la page a été chargée.

1. Au survol :

Nous allons découvrir plusieurs pseudo-formats CSS. Le premier que nous allons étudier s'appelle **:hover**.

Comme tous les autres pseudo-formats que nous allons voir, c'est une information que l'on rajoute après le nom de la balise (ou de la classe) dans le CSS, comme ceci :

```
a:hover
{
}
```

:hover signifie « **survol**er ». **a:hover** peut donc se traduire par : « Quand la souris est sur le lien » (quand on pointe dessus).

À partir de là, c'est à vous de définir l'apparence que doivent avoir les liens lorsqu'on pointe dessus.

EXEMPLE :

```
a /* Liens par défaut (non survolés) */
{
  text-decoration: none;
  color: red;
  font-style: italic;
}
a:hover /* Apparence au survol des liens */
{
  text-decoration: underline;
  color: green;
}
```

On a défini ici deux versions des styles pour les liens :

- pour les liens par défaut (non survolés) ;
- pour les liens au survol.

2. Au clic et lors de la sélection

Vous pouvez interagir encore plus finement en CSS. Nous allons voir ici que nous pouvons changer l'apparence des éléments lorsque l'on clique dessus et lorsqu'ils sont sélectionnés !

:active : au moment du clic

Le pseudo-format **:active** permet d'appliquer un style particulier *au moment du clic*. En pratique, il n'est utilisé que sur les liens.

Le lien gardera cette apparence très peu de temps : en fait, le changement intervient lorsque le bouton de la souris est enfoncé.

En clair, ce n'est pas forcément toujours bien visible.

On peut par exemple changer la couleur de fond du lien lorsque l'on clique dessus :

```
a:active /* Quand le visiteur clique sur le lien */
{
    background-color: #FFCC66;
}
```

:focus : lorsque l'élément est sélectionné

Là, c'est un peu différent. Le pseudo-format **:focus** applique un style *lorsque l'élément est sélectionné*.

Une fois que vous avez cliqué, le lien reste « sélectionné » (il y a une petite bordure en pointillés autour). C'est cela, la sélection.

Ce pseudo-format pourra être appliqué à d'autres balises HTML que nous n'avons pas encore vues, comme les éléments de formulaires.

EXEMPLE :

```
a:focus /* Quand le visiteur sélectionne le lien */
{
    background-color: #FFCC66;
}
```

3. Lorsque le lien a déjà été consulté :

Il est possible d'appliquer un style à un lien vers une page qui a déjà été vue. Par défaut, le navigateur colore le lien en un violet.

Vous pouvez changer cette apparence avec **:visited** (qui signifie « visité »). En pratique, sur les liens consultés, on ne peut pas changer beaucoup de choses à part la couleur.

```
a:visited /* Quand le visiteur a déjà vu la page concernée */  
{  
    color: #AAA; /* Appliquer une couleur grise */  
}
```

EN RÉSUMÉ :

- En CSS, on peut modifier l'apparence de certaines sections dynamiquement, après le chargement de la page, lorsque certains événements se produisent. On utilise pour cela les pseudo-formats.
- Le pseudo-format **:hover** permet de changer l'apparence au survol (par exemple : **a:hover** pour modifier l'apparence des liens lorsque la souris pointe dessus).
- Le pseudo-format **:active** modifie l'apparence des liens au moment du clic, **:visited** lorsqu'un lien a déjà été visité.
- Le pseudo-format **:focus** permet de modifier l'apparence d'un élément sélectionné.