



Les cahiers d'Exercices en Programmation : Le langage CSS

Apprenez et entraînez vos acquis

- De très nombreux exercices à réaliser par vous-même
- Des corrigés regroupés à la fin du cahier et expliqués Pas à Pas.

AVANT-PROPOS

Ce livre est un cahier d'exercices : il vous propose des énoncés d'exercices et leurs corrigés. Vous allez apprendre le logiciel en vous entraînant à travers des exercices regroupés par thème.

Chaque énoncé vous présente l'exercice à réaliser. Vous trouverez à la fin du cahier le corrigé de chaque exercice. Certaines explications peuvent-être présentes.

METHODOLOGIE

Lors de la réalisation des exercices, vous pourrez remédier à certain problème à l'aide des corrections à la fin du cahier.

Après avoir réalisé tous les exercices de chaque chapitre vous allez pouvoir vérifier les compétences acquises à l'aide du tableau des objectifs.

Celui-ci sert à la cotation du professeur (grille d'évaluation).

Des **légendes** ou **recommandations** peuvent être présentes dans certains exercices. Celles-ci vous aideront dans vos recherches. Elles ne doivent pas être reproduites dans votre travail.

Chaque point de matière acquis dans un exercice peut être utilisé dans des exercices suivants sans explication.

Table des matières

Chapitre 1 : Introduction au CSS	1
Chapitre 2 : CSS, des débuts difficiles.....	2
Chapitre 3 : CSS, la prise en charge des navigateurs	3
Chapitre 4 : Où écrit-on le CSS ?	4
Chapitre 5 : Appliquer un style : sélectionner une balise	7
Appliquer un style : class et id.....	7
Les balises universelles.....	9
Chapitre 6 : Appliquer un style : les sélections avancées.....	11
Les sélecteurs que vous connaissez déjà.....	11
Les sélecteurs avancés	12
En résumé.....	13
Chapitre 7 : Formatage du texte	14
La taille.....	14
La police.....	17
Italique, gras, souligné.....	18
L'alignement	19
Les flottants	20
Stopper un flottant	21
En résumé.....	22
Chapitre 8 : La couleur et le fond	23
Indiquer le nom de la couleur	23
Couleur de fond.....	24
Appliquer une image de fond	25
En résumé.....	28
Chapitre 9 : Les bordures et les ombres.....	29
Bordures standard	29
Bordures à droite, à gauche, en bas et en haut	30

Des bordures arrondies.....	31
Les ombres.....	32
En résumé.....	34
Chapitre 10 : Création d'apparences dynamiques.....	35
Au survol.....	35
Au clic et lors de la sélection.....	36
Lorsque le lien a déjà été consulté	37
En résumé.....	37
Chapitre 11 : Exercices	38
Chapitre 11 : Solutions	40
Bibliographie.....	42

Chapitre 1 : Introduction au CSS

Pour définir la présentation d'un document web écrit en HTML, on utilise un autre langage en complément, les feuilles de styles CSS (Cascading Style Sheet ou feuille de style en cascade).

Les feuilles de styles CSS permettent de définir les couleurs, les polices d'écriture, le positionnement des éléments.

Les feuilles peuvent indifféremment être placées dans les pages où on souhaite appliquer les styles, ou dans un fichier externe.

Cette dernière méthode permet de gagner un temps précieux car pour modifier le design complet du site, seule la mise à jour du fichier externe CSS est nécessaire. Il faut toutefois être prudent, car certains navigateurs comme Internet Explorer ne respectent pas entièrement les normes et provoquent dans certains cas un rendu différent des navigateurs les respectant.

Voici un petit comparatif :

HTML
(pas de CSS)



HTML + CSS



Grâce au HTML, nous avons pu rédiger le contenu de notre site

mais il est brut. Le CSS vient compléter ce code pour mettre en forme tout cela et donner au contenu l'apparence que l'on souhaite.

Chapitre 2 : CSS, des débuts difficiles

Il faut savoir qu'aux débuts du Web, CSS n'existait pas. En fait, il n'y avait initialement que le langage HTML.

Le HTML est né en 1991 et CSS en 1996. Alors, vous vous dites sûrement : comment faisait-on la mise en forme de 1991 à 1996 ? Eh bien, uniquement en HTML ! Il y avait en effet des balises HTML dédiées à la mise en forme.

Exemple : `` permettait de définir la couleur du texte.

Cependant, les pages HTML commençaient à devenir assez complexes. Il y avait de plus en plus de balises et c'était un joyeux mélange entre le fond et la forme, qui rendait la mise à jour des pages web de plus en plus complexe. C'est pour cela que l'on a créé le langage CSS.

Cependant, le CSS n'a pas été adopté immédiatement par les webmasters, loin de là. Il fallait se défaire de certaines mauvaises habitudes et cela a pris du temps. Encore aujourd'hui, on peut trouver des sites web avec des balises HTML de mise en forme, anciennes et obsolètes, comme `` !

Chapitre 3 : CSS, la prise en charge des navigateurs

Tout comme le HTML, le CSS a évolué. Je vous avais indiqué qu'il y avait quatre versions importantes de CSS :

- CSS 1 ;
- CSS 2.0 ;
- CSS 2.1 ;
- CSS 3.

En fait, la version CSS 3 n'est pas encore totalement finalisée (ce n'est pas encore une version officielle). Cependant, elle est bien avancée et aujourd'hui déjà bien prise en charge par de nombreux navigateurs, on peut déjà s'en servir. Il serait dommage de passer à côté car CSS 3 apporte de nombreuses fonctionnalités à CSS (leur nombre double par rapport à CSS 2.1 !). Nous nous baserons donc dans ce cours sur CSS 3, qui reprend et complète la plupart des fonctionnalités de CSS 2.1.

Ce sont les navigateurs web qui font le travail le plus complexe : ils doivent *lire* le code CSS et *comprendre* comment afficher la page.

Au début des années 2000, Internet Explorer était le navigateur le plus répandu mais sa gestion du CSS est longtemps restée assez médiocre (pour ne pas dire carrément mauvaise). C'était la grande époque de la version 6 (IE6).

Depuis, de nombreux navigateurs sont arrivés et ont chahuté Internet Explorer : Mozilla Firefox bien sûr, mais aussi Google Chrome. Et je ne vous parle pas du succès des Mac et iPhone avec leur navigateur Safari. Cela a incité Microsoft à réagir et publier (après une longue période d'inactivité) IE 7, puis IE 8 et IE 9, 10, 11... Et maintenant Edge.

Que faut-il retenir de tout cela ? Que les navigateurs ne connaissent pas toutes les propriétés CSS qui existent. Plus le navigateur est vieux, moins il connaît de fonctionnalités CSS.

Chapitre 4 : Où écrit-on le CSS ?

Vous avez le choix car on peut écrire du code en langage CSS à trois endroits différents :

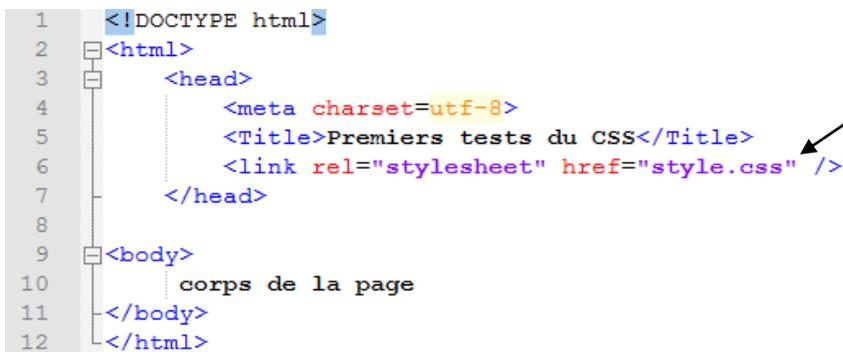
- dans un fichier .css (méthode la plus recommandée, nous emploierons cette méthode dans le cours) ;
- dans l'en-tête <head> du fichier HTML ;
- directement dans les balises du fichier HTML via un attribut style (méthode la moins recommandée).

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.

Reprendre le fichier HTML modèle et ajouter cette ligne et modifier le titre.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset=utf-8>
5     <Title>Premiers tests du CSS</Title>
6     <link rel="stylesheet" href="style.css" />
7   </head>
8
9   <body>
10     corps de la page
11   </body>
12 </html>
```



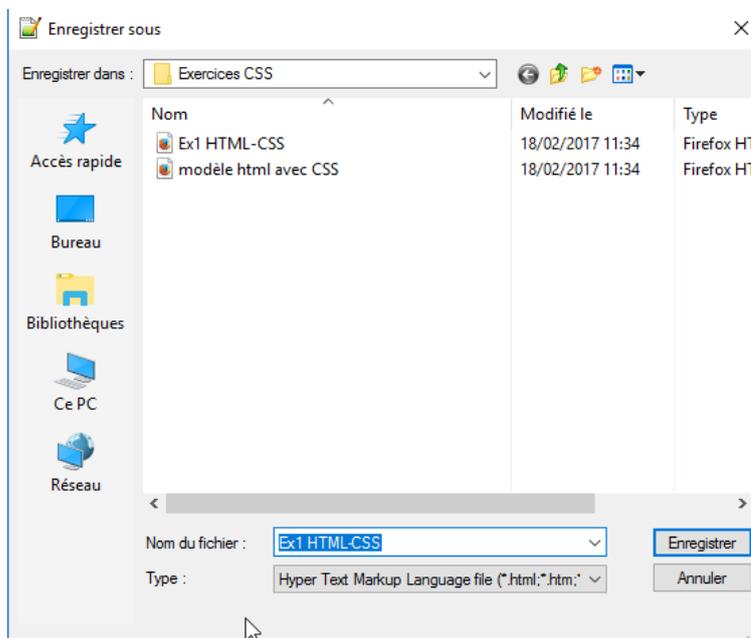
Le contenu de cette ligne 6, `<link rel="stylesheet" href="style.css" />` indique que ce fichier HTML est associé à un fichier appelé `style.css` et chargé de la mise en forme.

Enregistrer ce modèle sous le nom : `Modèle avec HTML et CSS`.

Introduire le code suivant dans le corps de la page :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset=utf-8>
5     <Title>Premiers tests du CSS</Title>
6     <link rel="stylesheet" href="style.css" />
7   </head>
8
9   <body>
10    <h1>Mon super site</h1>
11    <p>Bonjour et bienvenue sur mon site !</p>
12
13    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>
14  </body>
15 </html>
```

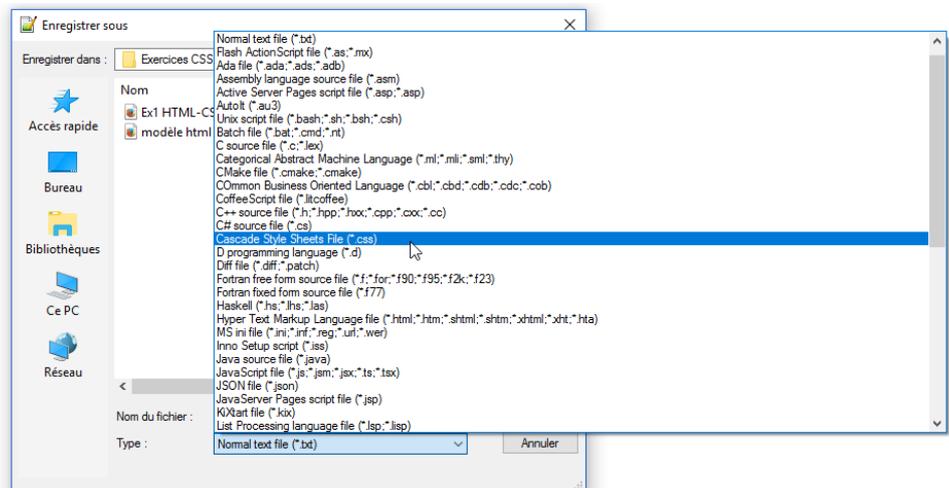
Enregistrer le travail :



Créer maintenant un nouveau fichier à l'aide de votre éditeur.

Enregistrer ce fichier dans le même dossier que ton fichier HTML avec comme

nom **style.css**
attention au type
de ce fichier :
(Cascade Style
Sheets) voir ci-
contre :



Copier dans ce fichier ce bout de code et enregistre ton travail :

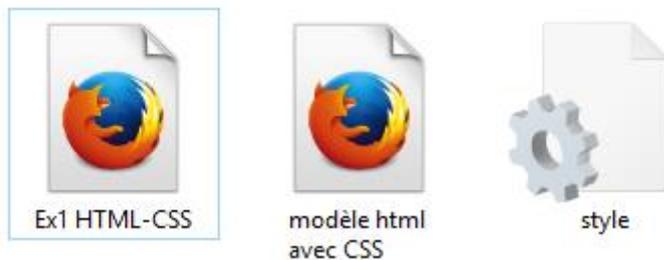
```
1 P
2 { color : Blue;
3 }
```

Les deux fichiers HTML et CSS doivent être ouverts.



```
style.css x Ex1 HTML-CSS.html x
1 P
2 { color : Blue;
3 }
```

Dans l'explorateur, il est possible de voir les 2 fichiers HTML + CSS ainsi que le modèle :



Ouvre maintenant le fichier HTML précédemment créé dans le navigateur pour le tester. C'est magique : le paragraphe est écrit en bleu.

Mon super site

Bonjour et bienvenue sur mon site !

Pour le moment, mon site est un peu vide. Patientez encore un peu !

Il ne faut donc pas ouvrir le fichier CSS mais bien que la page HTML (il fera automatiquement appel au fichier CSS).

Chapitre 5 : Appliquer un style : sélectionner une balise

Appliquer un style : class et id

Ce qui a été montré a quand même un défaut : TOUS les paragraphes possèdent la même présentation (ici, ils seront donc tous écrits en bleu).

Comment faire pour que certains paragraphes seulement soient écrits d'une manière différente ?

Pour résoudre le problème, on peut utiliser ces attributs spéciaux qui fonctionnent sur toutes les balises :

- l'attribut class ;
- l'attribut id.

Que les choses soient claires dès le début : les attributs class et id sont quasiment identiques. Il y a seulement une petite différence que je dévoilerai plus tard.

Pour le moment, et pour faire simple, on ne va s'intéresser qu'à l'attribut class.

C'est un attribut que l'on peut mettre sur n'importe quelle balise, aussi bien titre que paragraphe, image, etc.

En fait, il faut écrire un nom qui sert à identifier la balise. Ce que tu veux, du moment que le nom commence par une lettre.

Par exemple, je vais associer la classe `introduction` à mon premier paragraphe (ligne 11) :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset=utf-8>
5     <Title>Premiers tests du CSS</Title>
6     <link rel="stylesheet" href="style.css" />
7   </head>
8
9   <body>
10    <h1>Mon super site</h1>
11    <p class="introduction">Bonjour et bienvenue sur mon site !</p>
12
13    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>
14  </body>
15 </html>
```

Maintenant

que c'est fait, le paragraphe est identifié. Il a un nom : `introduction`. Utilisons ce

nom dans le fichier `CSS` pour dire : "Je veux que seules les balises qui ont comme nom `introduction` soient affichées en bleu".

Pour faire cela en `CSS`, indiquez le nom de la classe en commençant par un point, comme ci-dessous (modifier le contenu du fichier `style.css` précédent) :

```
1 .introduction
2 { color : Blue;
3 }
```

Résultat :



Et l'attribut `id` alors ?

Lui, il fonctionne exactement de la même manière que `class`, à un détail près : il ne peut être utilisé qu'une fois dans le code.

Quel intérêt ? Il y en a assez peu pour tout vous dire, cela vous sera utile si vous faites du JavaScript plus tard pour reconnaître certaines balises. D'ailleurs, nous avons déjà vu l'attribut `id` dans le chapitre sur les liens (pour réaliser des ancres en HTML). En pratique, nous ne mettrons des `id` que sur des éléments qui sont uniques dans la page, comme par exemple le logo.

Cherchez sur internet une image quelconque et enregistrez-la dans un dossier `images` (sous-dossier ou ton travail se situe).

Ajoute cette ligne à ton fichier HTML :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset=utf-8>
5     <Title>Premiers tests du CSS</Title>
6     <link rel="stylesheet" href="style.css" />
7   </head>
8
9   <body>
10    <h1>Mon super site</h1>
11
12    
13
14    <p class="introduction">Bonjour et bienvenue sur mon site !</p>
15
16    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>
17  </body>
18 </html>
```

Je

suppose qu'à ce niveau, la ligne de codes ne te fait pas trop peur.

Si tu utilises des `id`, lorsque tu définis leurs propriétés dans le fichier CSS, il faudra faire précéder le nom de l'`id` par un dièse (`#`) :

```
1 .introduction
2 { color : Blue;
3 }
4
5 #logo
6 {
7     /* Indiquez les propriétés CSS ici */
8 }
```

Les balises universelles

Voici ces deux balises avec un exemple à appliquer à ton travail :

- ` ` : c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement. Les balises `` et `` sont de la même famille. Cette balise s'utilise donc au milieu d'un paragraphe et c'est celle dont nous allons nous servir pour colorer « bienvenue ».
- `<div> </div>` : c'est une balise de type **block**, qui entoure un bloc de texte. Les balises `<p>`, `<h1>`, etc. sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau « bloc » dans la page et provoquent donc obligatoirement un retour à la ligne. `<div>` est une balise fréquemment utilisée dans la construction d'un design, comme nous le verrons plus tard.

Pour le moment donc, nous allons utiliser plutôt la balise ``. On la place autour de « mon site », on lui ajoute une classe (du nom que l'on veut), on crée le CSS et c'est gagné !

Fichier HTML :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset=utf-8>
5     <Title>Premiers tests du CSS</Title>
6     <link rel="stylesheet" href="style.css" />
7   </head>
8
9   <body>
10    <h1>Mon super site</h1>
11
12    
13
14    <p class="introduction">Bonjour et bienvenue sur mon site !</p>
15
16    <p>Pour le moment, <span class="nom">mon site</span> est un peu <em>vide</em>. Patientez encore un peu !</p>
17
18  </body>
19 </html>
```

Fichier CSS :

```
10 .nom
11 {
12     color: blue;
13 }
```

Ouvre maintenant ton travail à l'aide de ton navigateur et regarde le résultat :



Chapitre 6 : Appliquer un style : les sélections avancées

En CSS, le plus difficile est de savoir cibler le texte dont on veut changer la forme. Pour cibler (on dit « sélectionner ») les éléments de la page à modifier, on utilise ce qu'on appelle des **sélecteurs**.

Les sélecteurs que vous connaissez déjà

Ces sélecteurs, que nous avons vus précédemment, sont de loin les plus couramment utilisés. Il faut les connaître par cœur. Commençons par la base de la base :

Celui-ci veut veul toucher tous les paragraphes ». Après, c'est à vous de dire ce que vous faites à ces paragraphes (vous les écrivez en rouge, par exemple).

```
p {  
  }  
}
```

Tous les titres et tous les textes importants ». Nous avons sélectionné deux balises en un seul coup. On les sépare par une virgule.

```
h1, em  
{  
  }  
}
```

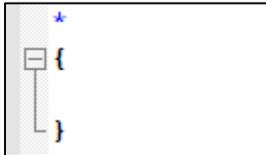
Et enfin, nous avons vu comment sélectionner des balises précises à qui nous avons donné un nom grâce aux attributs `class` et `id` :

```
.class  
{  
  }  
  
#id  
{  
  }  
}
```

Les sélecteurs avancés

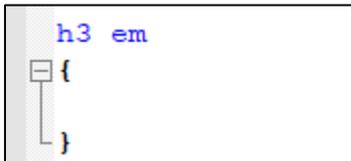
* : sélecteur universel

Sélectionne toutes les balises sans exception. On l'appelle le sélecteur universel.



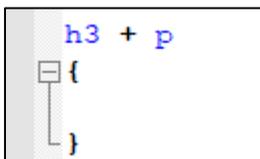
Balise A Balise B : une balise contenue dans une autre

Sélectionne toutes les balises `` situées à l'intérieur d'une balise `<h3>`. Notez qu'il n'y a pas de virgule entre les deux noms de balises.



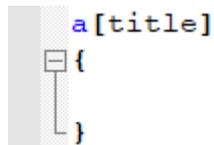
Balise A + Balise B : une balise qui en suit une autre

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



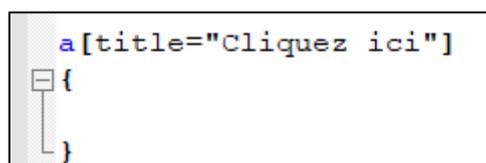
A[attribut] : une balise qui possède un attribut

Sélectionne tous les liens `<a>` qui possèdent un attribut `title`.



A[attribut="Valeur"] : une balise, un attribut et une valeur exacte

Idem, mais l'attribut doit en plus avoir exactement pour valeur « Cliquez ici ».



A[attribut*="Valeur"] : une balise, un attribut et une valeur

Idem, l'attribut doit cette fois contenir dans sa valeur le mot « ici » (peu importe sa position).

```
a[title*="ici"]  
{  
  }  
}
```

Une partie des sélecteurs CSS est présentée ici mais sachez qu'il en existe beaucoup d'autres.

Nous découvrirons certains de ces autres sélecteurs dans la suite de ce cours

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 :
- 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) ;
 - uniquement les balises qui se trouvent à l'intérieur d'autres balises (h3 em).
 - etc.

Chapitre 7 : Formatage du texte

Nous arrivons maintenant à un chapitre qui devrait beaucoup vous intéresser.

Non, le « formatage du texte » n'a rien à voir avec la destruction de toutes les données présentes sur votre disque dur ! Cela signifie simplement que l'on va modifier l'apparence du texte (on dit qu'on le « met en forme »).

Pas de surprise particulière : nous sommes toujours dans le CSS et nous allons réutiliser ce que nous venons d'apprendre dans le chapitre précédent. Nous allons donc travailler directement au sein du fichier `.css` que nous avons créé.

Ce chapitre va être l'occasion de découvrir de nombreuses propriétés CSS : nous allons voir comment modifier la taille du texte, changer la police, aligner le texte...

La taille

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

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

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 :

Voici un exemple d'utilisation (placez ce code dans votre fichier `style.css`) :

```
15 p
16 {
17     font-size: 14px; /*Paragraphes de 14 pixels*/
18 }
19
20 h1, em
21 {
22     font-size: 40px; /*Titres et textes affectés par une emphase en 40 pixels*/
23 }
24
```

Et en rappel voici votre fichier `Ex1 HTML-CSS.css`



```
style.css Ex1 HTML-CSS.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset=utf-8>
5     <Title>Premiers tests du CSS</Title>
6     <link rel="stylesheet" href="style.css" />
7 </head>
8
9 <body>
10 <h1>Mon super site</h1>
11
12 
13
14 <p class="introduction">Bonjour et bienvenue sur mon site !</p>
15
16 <p class="p">Pour le moment, <span class="nom">mon site</span> est un peu <em>vide</em>. Patientez encore un peu !</p>
17
18 </body>
19 </html>
```

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 7 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.

Tester les valeurs dans votre code CSS :

```
1  .introduction
2  {
3      color : Blue;
4      font-size: x-small;
5  }
```

Une taille peut aussi être introduite en `em`. Dans l'exemple que vous pouvez tester dans votre fichier CSS, le texte est 1.4 fois plus grand que la valeur normale.

```
12  .nom
13  {
14      color: red;
15      font-size: 1.4em;
16  }
```

Résultat :



Si la valeur est plus petite que 1 cela signifie que le texte sera plus petit que la valeur normale.

La police

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

La bonne nouvelle, c'est que depuis CSS 3, il est possible de faire télécharger automatiquement une police au navigateur. Explication futur.

Modifier la police utilisée

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

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

Il est préférable d'insérer plusieurs polices avec une priorité pour la première afin de permettre à l'internaute de quand-même apprécier votre site si la police principale utilisée n'est pas installée chez lui.

Voici un exemple que vous pouvez insérer dans votre fichier CSS :

```
23 h1, em
24 {
25     font-size: 40px; /*Titres et textes affectés par une emphase en 40 pixels*/
26     font-family: Impact, "Arial Black", Arial, Verdana, sans-serif; /*Polices*/
27 }
```

Si la police utilisée comporte plusieurs noms séparés d'un espace, tu dois l'insérer entre guillemets.

Italique, gras, souligné...

Il existe en CSS une série de propriétés classiques de mise en forme du texte. Nous allons découvrir ici comment afficher le texte en gras, italique, souligné... et au passage nous verrons qu'il est même possible d'aller jusqu'à le faire clignoter !

Mettre en italique

```
23 h1, em
24 {
25     font-size: 40px; /*Titres et textes affectés par une emphase en 40 pixels*/
26     font-family: Impact, "Arial Black", Arial, Verdana, sans-serif; /*Polices*/
27     font-style: italic;
```

Mettre en gras

```
18 p
19 {
20     font-size: 14px; /*Paragrapes de 14 pixels*/
21     font-weight: Bold;
22 }
```

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).

```
1 .introduction
2 {
3     color : Blue;
4     font-size: x-small;
5     text-decoration: line-through;
6 }
7
8 #logo
9 {
10     /* Indiquez les propriétés CSS ici */
11 }
12
13 .ncm
14 {
15     color: red;
16     font-size: 1.4em;
17     text-decoration: underline;
18 }
19
20 p
21 {
22     font-size: 14px; /*Paragrapes de 14 pixels*/
23     font-weight: Bold;
24     text-decoration: overline;
25 }
26
27 h1, em
28 {
29     font-size: 40px; /*Titres et textes affectés par une emphase en 40 pixels*/
30     font-family: Impact, "Arial Black", Arial, Verdana, sans-serif; /*Polices*/
31     font-style: italic;
32     text-decoration: blink;
33 }
```

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.

Appliquer les différents alignements mais avant tout tu vas séparer ces deux sélecteurs et modifier le contenu :

```
27 h1
28 {
29     font-size: 40px; /*Titres et textes affectés par une emphase en 40 pixels*/
30     font-family: Impact, "Arial Black", Arial, Verdana, sans-serif; /*Polices*/
31     font-style: italic;
32     text-decoration: blink;
33 }
34
35 em
36 {
37     font-size: 25px; /*Titres et textes affectés par une emphase en 40 pixels*/
38     font-family: "Comic Sans MS", "Arial Black", Arial, Verdana, sans-serif; /*Polices*/
39     font-style: italic;
40 }
```

Maintenant, tu vas ajouter un alignement au sélecteur `h1`.

```
27 h1
28 {
29     font-size: 40px; /*Titres et textes affectés par une emphase en 40 pixels*/
30     font-family: Impact, "Arial Black", Arial, Verdana, sans-serif; /*Polices*/
31     font-style: italic;
32     text-decoration: blink;
33     text-align: center;
34 }
```

Voici d'autres alignements : `center`, `justify`, `left`, `right`.

Les flottants

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

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vitae lorem imperdiet lacus molestie molestie. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec eu purus. Phasellus metus lorem, blandit et, posuere quis, tincidunt vitae, ante. Vivamus consequat mauris a diam. Vivamus nibh erat, hendrerit nec, aliquet ut, hendrerit quis, nunc. Vestibulum et turpis et elit tempor euismod.

Pour cela, recherche dans *GOOGLE* la lettre **f** avec le format image.

En voici un exemple : 

Enregistre cette lettre comme tu le désires sur ton disque dur de préférence là où ton travail se réalise.

Crée dans ton fichier CSS une nouvelle classe afin d'y intégrer certaines propriétés à ton texte. Nom de la classe **p1**.

```
27 .p1
28 {
29     font-size: 14px;
30     font-family: "Comic Sans MS";
31     color: Blue;
32     text-decoration: none;
33 }
```

Ajoute ces lignes à ton fichier HTML.

```
20 <p class="p1"> acile,
21 ceci est un texte normal de paragraphe, <br>écrit à la suite de l'image et qui <br>l'habillera car l'image est
22 flottante.</p>
```

Et voici le seul bout de code CSS qu'on ait besoin de taper, qui permet de faire flotter l'image à gauche :

```
51 .imageflottante
52 {
53     float: left;
54 }
```

Voici ce le résultat :



Pacile, ceci est un texte normal de paragraphe,
écrit à la suite de l'image et qui
l'habillera car l'image est flottante.

Stopper un flottant

Quand vous mettez en place un flottant, le texte autour l'habille. Mais comment faire si vous voulez qu'au bout d'un moment le texte continue *en dessous* du flottant ? On pourrait enchaîner plusieurs `
` à la suite mais cela ne serait ni élégant ni très propre...

En gros, on aimerait pouvoir obtenir le même résultat qu'à la figure suivante :



Pacile, ceci est un texte normal de paragraphe,
écrit à la suite de l'image et qui l'habillera
car l'image est flottante.

Il existe en fait une propriété CSS qui permet de dire : « Stop, ce texte doit être en-dessous du flottant et non plus à côté ». C'est la propriété `clear`, qui peut prendre ces trois valeurs :

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

Pour simplifier, on va utiliser tout le temps le `clear: both`, qui marche après un flottant à gauche et après un flottant à droite (cela fonctionne donc à tous les coups).

Ajoute ce code CSS en créant une nouvelle classe.

```
56 .dessous
57 {
58     clear: both;
59 }
```

Et voici le code HTML à modifier :

```
20 <p class="p1"> acile,  
21 ceci est un texte normal de paragraphe,<p class="p1","dessous">écrit à la suite de l'image et qui l'habillera </br> car l'image est  
22 flottante.</p>  
23
```

Regarde le résultat.

En résumé

- On modifie la taille du texte avec la propriété CSS `font-size`. On peut indiquer la taille en pixels (16px), en « em » (1.3em), en pourcentage (110%), etc.
- On change la police du texte avec `font-family`. Attention, seules quelques polices sont connues par tous les ordinateurs.
- De nombreuses propriétés de mise en forme du texte existent : `font-style` pour l'italique, `font-weight` pour la mise en gras, `text-decoration` pour le soulignement, etc.
- Le texte peut être aligné avec `text-align`.
- On peut faire en sorte qu'une image soit habillée (« entourée ») de texte avec `float`.

Chapitre 8 : La couleur et le fond

Continuons notre tour d'horizon des propriétés CSS existantes. Nous allons nous intéresser ici aux propriétés liées de près ou de loin à la couleur. Nous verrons entre autres :

- comment changer la couleur du texte ;
- comment mettre une couleur ou une image d'arrière-plan ;
- comment ajouter des ombres ;
- comment jouer avec les niveaux de transparence.

Le CSS n'a pas fini de nous étonner !

Indiquer le nom de la couleur

La méthode la plus simple et la plus pratique pour choisir une couleur consiste à taper son nom (*in english, of course*).

Avec cette méthode il n'existe que seize couleurs dites « standard » comme par exemple : white, silver, gray, black, red, maroon, lime, green, yellow, olive, blue navy, aqua, purple, fuchsia, teal.

Pour passer tous les titres en marron, tu peux écrire dans ton fichier CSS :

```
35  h1
36  {
37      font-size: 40px; /*Titres et textes affectés par une emphase en 40 pixels*/
38      font-family: Impact, "Arial Black", Arial, Verdana, sans-serif; /*Polices*/
39      font-style: italic;
40      text-decoration: blink;
41      text-align: center;
42      color: maroon; ←
43  }
```

Les autres méthodes ne vont pas être développées dans ce cours mais je vais quand-même te les citer : méthode RGB et méthode hexadécimal.

Voici un exemple pour chacun :

- Méthode RGB : `color: rgb(240,96,204);`
- Méthode hexadécimale : `color: #FFFFFF;`

À l'aide de ces deux méthodes vous pourrez utiliser beaucoup beaucoup de couleurs différentes.

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 tu peux 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>`. Cette balise 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.

Regardez très attentivement ce fichier CSS :

```
64  body
65  {
66      background-color: black; /* Le fond de la page sera noir */
67      color: white; /* Le texte de la page sera blanc sauf le texte ou une
68                  autre propriété est appliquée.
69                  On parle d'héritage*/
70  }
71
72  mark
73  {
74      /* La couleur de fond prend le pas sur celle de toute la page */
75      background-color: red;
76      color: black;
77  }
```

La balise `mark` va créer un surlignage. Applique donc cette balise dans ton fichier HTML.

```
20  <p class="p1">texte normal</mark> de paragraphe,<p
22  flottante.</p>
23
24  </body>
```

Admire maintenant ton travail.

Appliquer une image de fond

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.

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

À l'aide de ces mots, recherche un fichier sur internet correspondant : `neige.png`

Enregistre l'image de préférence là où est situé ton fichier CSS.

Modifie le code de ta balise `body` comme ci-dessous :

```
64  body
65  {
66      background-image: url("neige.png");
67  }
```

L'image peut être fixée au fond à l'aide de la propriété `background-attachment`. 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).

Modifie ton code :

```
64  body
65  {
66      background-image: url("neige.png");
67      background-attachment: fixed; /* Le fond restera fixe */
68  }
```

Par défaut, l'image de fond est répétée en mosaïque. Tu peux 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 d'utilisation à insérer dans ton code :

```
64 body
65 {
66     background-image: url("neige.png");
67     background-attachment: fixed; /* Le fond restera fixe */
68     background-repeat : no-repeat; /*Il ne sera pas répété */
69 }
```

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`;

Deux valeurs en pixels sont a donné pour indiquer la position du fond par rapport au coin supérieur gauche de la page (ou du paragraphe).

Introduis ce code :

```
64 body
65 {
66     background-image: url("neige.png");
67     background-attachment: fixed; /* Le fond restera fixe */
68     background-repeat : no-repeat; /*Il ne sera pas répété */
69     background-position: 30px 50px; /*position          */
70 }
```

Il est aussi possible d'utiliser ces valeurs en anglais :

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

Si beaucoup de propriétés en rapport avec le fond sont utilisées, tu peux faire en sorte de créer une « 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 :

```
64  body
65  {
66      background: url("neige.png") fixed no-repeat 30px 50px;
67  }
```

Regarde ton travail mais normalement rien ne change.

Plusieurs images peuvent-être introduite dans ton `background`, elles se positionneront dans m'ordre de l'écriture.

Essaie si tu as envie.

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` (pas vu dans ce cours).

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 :

```
69  mark
70  {
71      /* La couleur de fond prend le pas sur celle de toute la page */
72      background-color: red;
73      color: black;
74      opacity: 0.6;
75  }
```

Mettre une image à l'échelle, diminuer ou augmenter la grandeur d'un fond d'écran peut se réaliser à l'aide de cette propriété : `background-size`.

Essaie ce code :

```
64  body
65  {
66      background: url("neige.png") fixed no-repeat 30px 50px;
67      background-size: 130px 130px;
68  }
```

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` ou avec la notation `RGBA` (identique à la notation `RGB`, avec une quatrième valeur indiquant le niveau de transparence).

Chapitre 9 : Les bordures et les ombres

Nouveau chapitre, nouveau lot de propriétés CSS.

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.

Des couleurs vont être choisies pour nos bordures et de nos ombres, voir chapitre précédent.

On y va !

Bordures standard

Le CSS vous offre un large choix de bordures pour décorer votre page.

Nous utiliserons la super-propriété `border` qui regroupe l'ensemble de ces propriétés. Même fonctionnement que la super-propriété `background` ? Plusieurs valeurs peuvent être combinées.

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.

Ainsi, pour avoir une bordure bleue, en tirets, épaisse de 3 pixels autour de mes titres, je vais écrire dans le fichier CSS :

```
35 h1
36 {
37     font-size: 40px; /*Titres et textes affectés par une emphase en 40 pixels*/
38     font-family: Impact, "Arial Black", Arial, Verdana, sans-serif; /*Polices*/
39     font-style: italic;
40     text-decoration: blink;
41     text-align: center;
42     color: maroon;
43     border: 3px blue dashed;
44 }
```

Bordures à droite, à gauche, en bas et en haut

Des bordures différentes en fonction du côté (haut, bas, gauche ou droite). Voici les 4 propriétés :

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

Voici ce qui a été modifié pour les paragraphes :

```
20 p
21 {
22     font-size: 14px; /*Paragraphes de 14 pixels*/
23     font-weight: Bold;
24     text-decoration: overline;
25     border-left: 2px solid black;
26     border-right: 2px solid black;
27 }
```

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.

Des 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 !

Voici la propriété `border-radius`. Ne pas oublier d'introduire l'importance de l'arrondi. Voici un exemple que tu peux insérer dans ton travail.

Il te faudra modifier la classe `p1` :

```
29 .p1
30 {
31     font-size: 14px;
32     font-family: "Comic Sans MS";
33     color: Blue;
34     text-decoration: none;
35     border-radius: 10px;
36 }
```

Et ajouter un peu de texte dans ton fichier HTML :

```
20 <p class="p1"> acile,
21 ceci est un <mark>texte normal</mark> de paragraphe,<p class="p1", "dessous">écrit à la suite de l'image et qui
22 l'habillera </br> car l'image est flottante. Maintenant tu as introduit </br> une bordure à gauche et à droite
23 du texte et en plus </br> la bordure est arrondie sur les coins</p>
24
25 </body>
26 </html>
```

Les ombres

Nous allons ici découvrir deux types d'ombres :

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

Les ombres des boîtes : box-shadow

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

1. le décalage horizontal de l'ombre ;
2. le décalage vertical de l'ombre ;
3. l'adoucissement du dégradé ;
4. la couleur de l'ombre.

Avant de continuer je vais te demander de créer une nouvelle classe et d'en modifier une autre :

Crée celle-ci :

```
38 .p2
39 {
40     font-size: 14px;
41     font-family: "Comic Sans MS";
42     color: black;
43     border: 2px solid red;
44 }
```

Modifie celle-ci (mise en commentaire) :

```
20 p
21 {
22     font-size: 14px; /*Paragraphe de 14 pixels*/
23     font-weight: Bold;
24     /* text-decoration: overline;
25     border-left: 2px solid black;
26     border-right: 2px solid black;*/
27 }
```

Et ensuite tape tout ce texte :

```
25 <p class="p2">
26 JavaScript est un langage de programmation de scripts principalement employé dans les pages
27 web interactives mais aussi pour les serveurs2 avec l'utilisation (par exemple) de Node.JS.
28 C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales
29 interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun
30 équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de
31 prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions
32 sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel.
33 JavaScript est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances npm,
34 avec plus de 350 000 paquets3.
35 JavaScript a été créé en 1995 par Brendan Eich. Il a été standardisé sous le nom d'ECMAScript
36 en juin 1997 par Ecma International dans le standard ECMA-262. Le standard ECMA-262 en est
37 actuellement à sa 8e édition. JavaScript n'est depuis qu'une implémentation d'ECMAScript,
38 celle mise en œuvre par la fondation Mozilla. L'implémentation d'ECMAScript par Microsoft
39 se nomme JScript, tandis que celle d'Adobe Systems se nomme ActionScript.
40 </p>
41 </body>
42 </html>
```

Tu peux le trouver dans **Wikipédia**, cela sera plus rapide je crois.

Porter une ombre noire de 6 pixels, sans adoucissement, on écrira :

```
38 .p2
39 {
40     font-size: 14px;
41     font-family: "Comic Sans MS";
42     color: black;
43     border: 2px solid red;
44     box-shadow: 6px 6px 0px black;
45 }
```

Adoucissement :

```
38 .p2
39 {
40     font-size: 14px;
41     font-family: "Comic Sans MS";
42     color: black;
43     border: 2px solid red;
44     box-shadow: 6px 6px 6px black;
45 }
```

L'ombre du texte : text-shadow

La propriété **text-shadow** permet d'ajouter une ombre sur les lettres.

```
38 .p2
39 {
40     font-size: 14px;
41     font-family: "Comic Sans MS";
42     color: black;
43     border: 2px solid red;
44     box-shadow: 6px 6px 6px black;
45     text-shadow: 2px 2px 4px black;
46 }
```

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`.

Chapitre 10 : 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. Nous allons faire appel à une fonctionnalité puissante du CSS : les pseudo-formats.

Nous verrons dans ce chapitre comment changer l'apparence :

- au survol ;
- lors du clic ;
- lors du focus (élément sélectionné) ;
- lorsqu'un lien a été consulté.

Le langage CSS n'a pas encore fini de nous étonner.

Au survol

Nous allons découvrir dans ce chapitre plusieurs pseudo-formats CSS.

Le premier abordé 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 :

```
92 a: hover
93 {
94   |
95 }
```

Hover en anglais signifie : survoler. Réaction du texte quand la souris survole un ou des mots.

Voici un exemple que tu peux réaliser :

Dans le fichier CSS :

```
92 a:hover
93 {
94     text-decoration: underline;
95     color: green;
96 }
```

Et ensuite dans ton fichier HTML :

```
25 <p class="p2">
26 <a>JavaScript</a> est un langage de program
27 web interactives mais aussi pour les serveu
28 C'est un langage orienté objet à prototype,
```

On utilise souvent ce changement d'apparence pour les liens vers des sites Web.

Hover peut être aussi utilisé quand on pointe sur un paragraphe : `p:hover`

Au clic et lors de la sélection

Nous pouvons changer l'apparence des éléments lorsque l'on clique dessus et lorsqu'ils sont sélectionnés !

`:active` : au moment du clic

`:focus` : lors de la sélection

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, 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. Voici un exemple :

```
98     a:active /* Quand le visiteur clique sur le lien */
99     {
100         background-color: #FFCC66;
101     }
```

Tu appliques maintenant un autre changement d'apparence lors du clic sur le mot Javascript.

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 assez laid (de mon point de vue du moins !).

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. Voici un exemple que tu ne dois pas introduire dans ton travail.

```
109     a:visited /* Quand le visiteur a déjà vu la page concernée */
110     {
111         color: #AAA; /* Appliquer une couleur grise */
112     }
```

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.

Chapitre 11 : Exercices

Tu vas mettre en forme ta page avec le langage CSS. Trouve les bonnes propriétés CSS ?

À toi de jouer !

Il faut retrouver sur la page Web les points suivants. Ceux-ci montreront que la matière a bien été acquise :

1. Ajoutez une balise pour lier le fichier CSS au fichier HTML
2. Affichez le titre h1 en rouge
3. Agrandissez le titre h1 pour lui donner une taille de 2em
4. Centrez la figure (image) et sa description
5. Ajoutez une bordure continue de la taille et de la couleur de votre choix autour de la liste à puces
6. Ajoutez une ombre au cadre autour de la liste à puces
7. Affichez les liens en gras lorsqu'on les survole

Avant de commencer, tu dois créer un fichier HTML nommé : **Index Cupcakes**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Les cupcakes</title>
  </head>

  <body>
    <h1>Vive les cupcakes</h1>

    <p>J'ai découvert les cupcakes lors d'un voyage à <a href="https://fr.wikipedia.org/wiki/New_York">New York</a>
    ... et depuis c'est une véritable passion. Sur mon site, vous saurez <strong>tout</strong> sur les cupcakes !</p>

    <p>Un cupcake ? C'est ça :</p>

    <figure>
      
      <figcaption>Un cupcake</figcaption>
    </figure>

    <p>Délicieux non ? Sur mon site, je vous présenterai :</p>

    <ul>
      <li>Les différentes variétés de cupcakes</li>
      <li>Leur histoire</li>
      <li>Leurs recettes</li>
    </ul>

  </body>
</html>
```

N'oublie pas d'aller chercher une image d'un cupcake en jpg et de copier ce fichier dans le dossier de tes exercices.

Créer un fichier CSS nommé : **Style Cupcakes**

```
/* Feuille de styles CSS */
```

C'est à toi maintenant de travailler et de trouver la solution des 7 exercices présentés à la page précédente.

Chapitre 11 : Solutions

1. Ajoutez une balise pour lier le fichier CSS au fichier HTML

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Les cupcakes</title>
6     <link rel="stylesheet" href="Style Cupcakes.css" />
7   </head>
8
9   <body>
10    <h1>Vive les cupcakes</h1>
```

2. Affichez le titre h1 en rouge

```
1 /* Feuille de styles CSS */
2 h1
3 {
4   color: red;
5 }
6
```

3. Agrandissez le titre h1 pour lui donner une taille de 2em

```
1 /* Feuille de styles CSS */
2 h1
3 {
4   color: red;
5   font-size: 2em;
6 }
```

4. Centrez la figure (image) et sa description

Je ne veux pas voir ceci. Tu dois travailler à l'aide de ton fichier CSS

```
17 <figure style="text-align:center">
18   
19   <figcaption>Un cupcake</figcaption>
20 </figure>
21
22 <p>Délicieux non ? Sur mon site, je vous présenterai :</p>
```

Création d'une class

```
8 .figure
9 {
10     text-align: center;
11 }
```

Application de la classe à l'image et au texte)

```
<figure class="figure">
  
  <figcaption>Un cupcake</figcaption>
</figure>

<p>Délicieux non ? Sur mon site, je vous présenterai </p>
```

5. Ajoutez une bordure continue de la taille et de la couleur de votre choix autour de la liste à puces

```
13 .ul
14 {
15     border: 2px solid blue;
16 }
```

```
<ul class="ul">
  <li>Les différentes variétés de cupcakes</li>
  <li>Leur histoire</li>
  <li>Leurs recettes</li>
</ul>
```

6. Ajoutez une ombre au cadre autour de la liste à puces

```
13 .ul
14 {
15     border: 2px solid blue;
16     box-shadow: 6px 6px 6px grey;
17 }
```

7. Affichez les liens en gras lorsqu'on les survole

```
19 a:hover
20 {
21     font-weight: bold;
22 }
23 }
```

Bibliographie

- Site Open Class Room
- Apprendre à coder n° 4 - Vigot
- Réalisez votre site Web avec l'HTML 5 - Mathieu Nebra - Le Livre du zéro
- HTML5 et CSS3 : Maîtrisez les standards des applications Web 2^{ème} Edition - Luc Van Lancker - Eni Edition