

HTML



CSS



Mises en forme de textes en CSS



04/223.11.31 - www.isf.be

Rédigé par: Jean-Luc Colson

Date première rédaction: Jan 2020

SUIVI DES MODIFICATIONS A LA FICHE

Contents

La propriété CSS font-family et les Google Fonts.....	5
La propriété CSS font-family : définition et exemples d'utilisation	5
Les web safe fonts et les familles génériques	6
Définition, intégration et utilisation des Google Fonts	8
Les autres propriétés CSS liées à la police.....	15
La propriété font-size	15
Les valeurs « mot clef »	16
Les valeurs « longueur ».....	18
Les valeurs « pourcentage ».....	21
Les valeurs universelles	21
La propriété font-weight	22
La propriété font-style.....	22
La notation raccourcie font	23
Les propriétés CSS liées au texte.....	26
La propriété text-align.....	26
La propriété text-transform	27
La propriété text-decoration	28
La propriété text-indent	31
La propriété text-shadow	32
Gérer la taille des interlignes et des espaces dans les textes en CSS.....	34
La propriété line-height.....	34
La propriété letter-spacing.....	35
La propriété word-spacing	36
Couleur des textes HTML avec CSS color et opacité	38
Les valeurs de type « nom de couleur ».....	38
Les autres couleurs nommées.....	40
Avantages et limitations des valeurs de type « nom de couleur »	50
Les notations de type RGB	50
Les valeurs de type hexadécimales	52
Les notations de type HSL	55

L'opacité des éléments et des textes en CSS	58
Gérer l'opacité des éléments avec la propriété opacity	58
Gérer l'opacité des textes avec les notations RGBa ou HSLa	59

La propriété CSS font-family et les Google Fonts

Dans cette nouvelle partie, nous allons nous intéresser aux propriétés CSS permettant de mettre en forme des textes. Les propriétés CSS liées au texte peuvent être séparées en deux grandes catégories:

- Les propriétés de type **font-** qui vont définir l'aspect des caractères en soi en qui agissent directement sur la police d'écriture (choix de la police, de la taille des caractères ou de leur poids entre autres) ;
- Les propriétés de type **text-** qui ne vont pas impacter directement l'aspect des caractères du texte mais nous permettre d'ajouter des effets de style autour de celui-ci (alignement du texte, soulignement ou encore ajout d'ombres autour des textes par exemple).

Dans cette leçon, nous allons nous intéresser à une première propriété de type font- qui va nous permettre de définir la police de nos textes : la propriété font-family.

La propriété CSS font-family : définition et exemples d'utilisation

La propriété CSS font-family va nous permettre de définir la police de nos textes, c'est-à-dire le rendu graphique de chaque caractère.

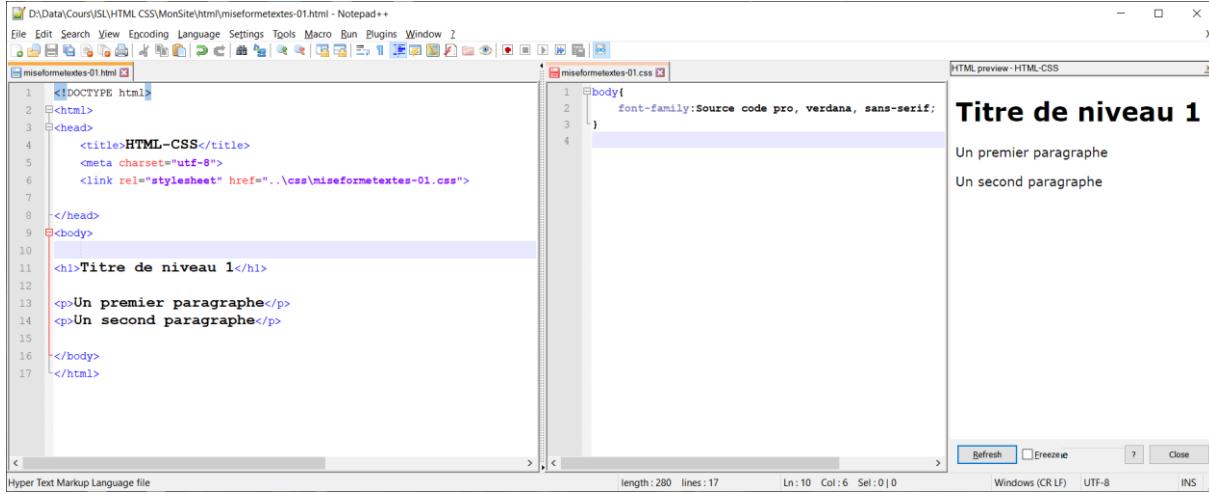
Il existe aujourd'hui un très grand nombre de polices d'écritures disponibles et parmi lesquelles on va pouvoir choisir mais il faut savoir que certaines versions de certains navigateurs que peuvent posséder vos visiteurs ne vont pas supporter certaines polices.

Pour cette raison, nous indiquerons toujours plusieurs noms de polices (une « famille de polices ») à utiliser en valeur de la propriété font-family en commençant par celle préférée et en séparant chaque valeur par une virgule.

Ainsi, le navigateur va lire les différentes polices renseignées dans font-family dans l'ordre et utiliser la première qu'il supporte.

Notez ici que si vous renseignez un nom de police qui contient des espaces, vous devrez le mettre entre guillemets ou apostrophes.

Voyons un premier exemple d'utilisation de font-family ensemble. On va définir une même police pour toute la partie visible de notre page HTML en appliquant notre propriété à l'élément body (les éléments dans le body hériteront par défaut de la valeur passée à la propriété font-family pour l'élément body).



The screenshot shows a Notepad++ interface with three windows:

- miseformetextes-01.html**: Contains the following HTML code:

```

<!DOCTYPE html>
<html>
<head>
    <title>HTML-CSS</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href=".\\css\\miseformetextes-01.css">
</head>
<body>
    <h1>Titre de niveau 1</h1>
    <p>Un premier paragraphe</p>
    <p>Un second paragraphe</p>
</body>
</html>

```
- miseformetextes-01.css**: Contains the following CSS code:

```

body {
    font-family: Source code pro, verdana, sans-serif;
}

```
- HTML preview - HTML-CSS**: Shows the rendered output with the title "Titre de niveau 1" in bold black font, and two paragraphs below it.

Dans cet exemple, j'indique trois polices d'écriture différentes en valeur de ma propriété font-family: Source code pro, Verdana et sans-serif.

Lors de l'affichage, mon navigateur va lire les valeurs dans leur ordre d'écriture et va appliquer la première valeur de police qu'il supporte / reconnaît.

Dans cet exemple, j'ai indiqué intentionnellement trois polices particulières : la police **Source code pro** est ce qu'on appelle une « **Google Font** » (nous reviendrons dessus plus tard) tandis que **Verdana** fait partie des « **web safe fonts** » (en français « famille de polices sûres ») et **sans-serif** est ce qu'on appelle une « **famille générique** ».

Les web safe fonts et les familles génériques

Les web safe fonts où « polices d'écriture sûres pour le web » sont un ensemble de polices qui sont lues de manière universelle par toutes les versions de tous les navigateurs dignes de ce nom.

En précisant une police qui appartient à cette catégorie en valeur de font-family, on s'assure donc que le navigateur de nos visiteurs sera à minima capable de lire et d'afficher cette police (à priori).

Une famille générique représente pas une police en soi mais va servir à définir la forme des caractères à afficher de manière sommaire. Les familles génériques doivent toujours être mentionnées en dernière valeur de la propriété font-family. Elles vont être utilisées en dernier recours si le navigateur n'arrive à lire aucune des polices indiquées dans la propriété font-family.

La famille générique serif par exemple est caractérisée l'empattement de ses caractères tandis que ce sera le contraire pour sans-serif. Notez bien que chaque police connue appartient automatiquement à une famille de polices par défaut.

Il existe 5 familles génériques disponibles en CSS aujourd'hui :

Famille générique	Description

serif	Les caractères possèdent un empattement
sans-serif	Les caractères ne possèdent pas d'empattement
monospace	Les caractères ont tous la même taille et sont espacés équitablement
cursive	Les caractères sont souvent fins et ressemblent à une écriture manuelle soignée
fantasy	Les caractères sont généralement difformes ou très stylisés, au contraire de la famille générique monospace

En général, on indiquera toujours au moins une police appartenant aux web safe fonts en valeur de notre propriété font-family ainsi qu'une famille générique associée afin d'être certain qu'au moins une valeur puisse être lue par le navigateur de vos visiteurs.

Comme je vous l'ai dit précédemment, chaque police connue appartient automatiquement à une famille de polices. Une bonne pratique va donc être de préciser une série de polices appartenant à la même famille générique (par exemple, que des polices sans-serif) puis de préciser en dernière valeur de font-family la famille générique associée afin de garder un maximum de cohérence.

Voici une liste des web safe fonts les plus utilisées et de leur famille générique associée :

Famille générique	Police
Serif	Times New Roman, Georgia
Sans-serif	Arial, Verdana
Monospace	Courier New, Lucida Console
Cursive	Comic sans MS

Définition, intégration et utilisation des Google Fonts

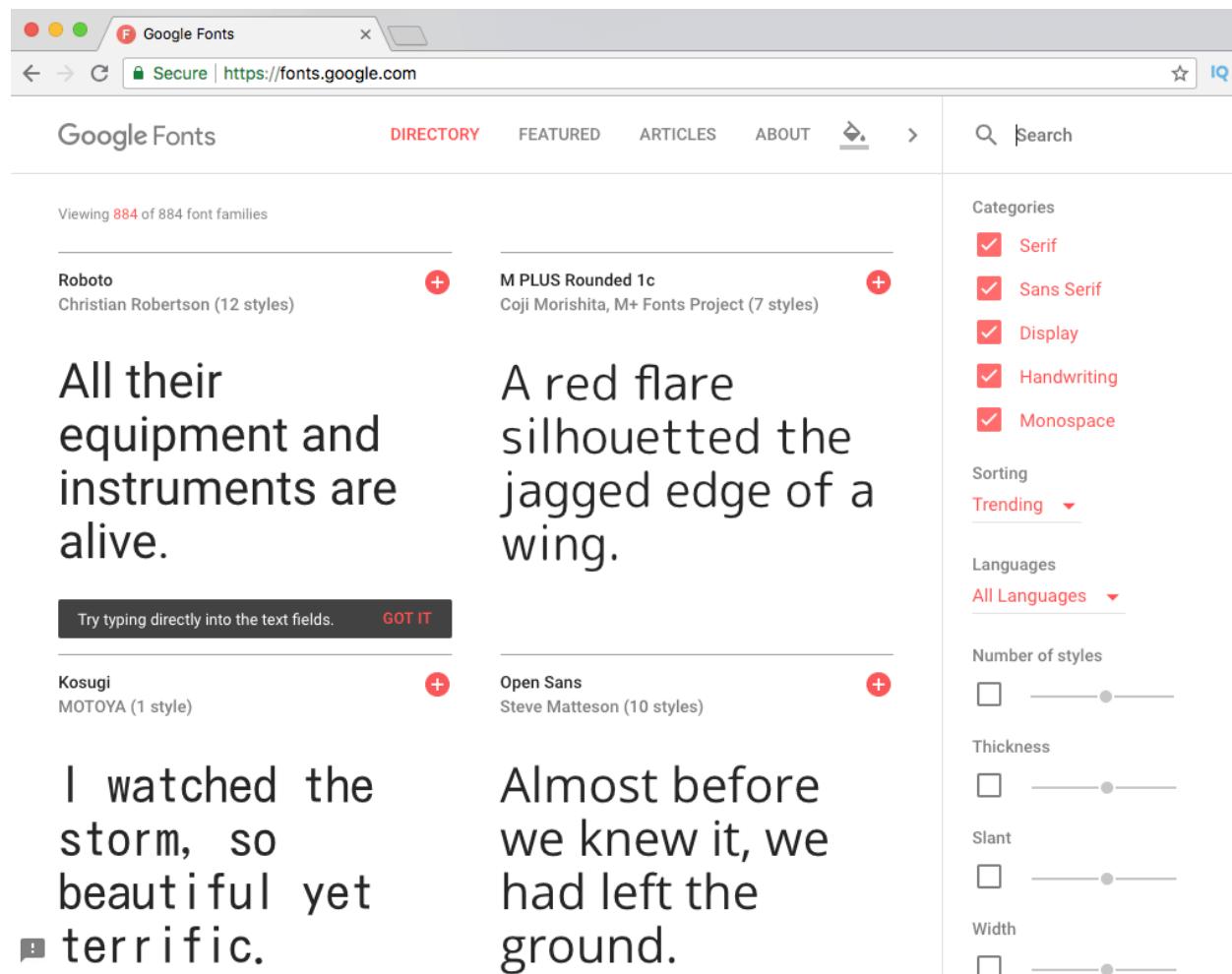
Google a développé sa propre liste de polices d'écriture appelées les **Google Fonts**.

Ces polices peuvent être utilisées sur n'importe quel site, le tout est de faire un lien en HTML vers la police Google désirée puis de l'utiliser comme valeur de notre propriété font-family en CSS.

Un des grands intérêts des Google Fonts est que les polices de Google ne dépendent pas d'un navigateur mais peuvent être lues par (quasiment) tous les navigateurs. En effet, l'idée va être de pré-charger le jeu complet de caractères de la police via Google dans le navigateur d'un utilisateur lorsque celui-ci tente d'accéder à une page de notre site. Ainsi, les risques de non comptabilité d'une police sont très sérieusement réduits.

L'autre grand avantage est que nous allons pouvoir choisir parmi une très longue liste de polices et ainsi totalement personnaliser l'écriture sur votre site.

Pour utiliser les Google Fonts, il va tout d'abord nous falloir choisir une police dans la liste proposée sur [le site de Google dédié](#).



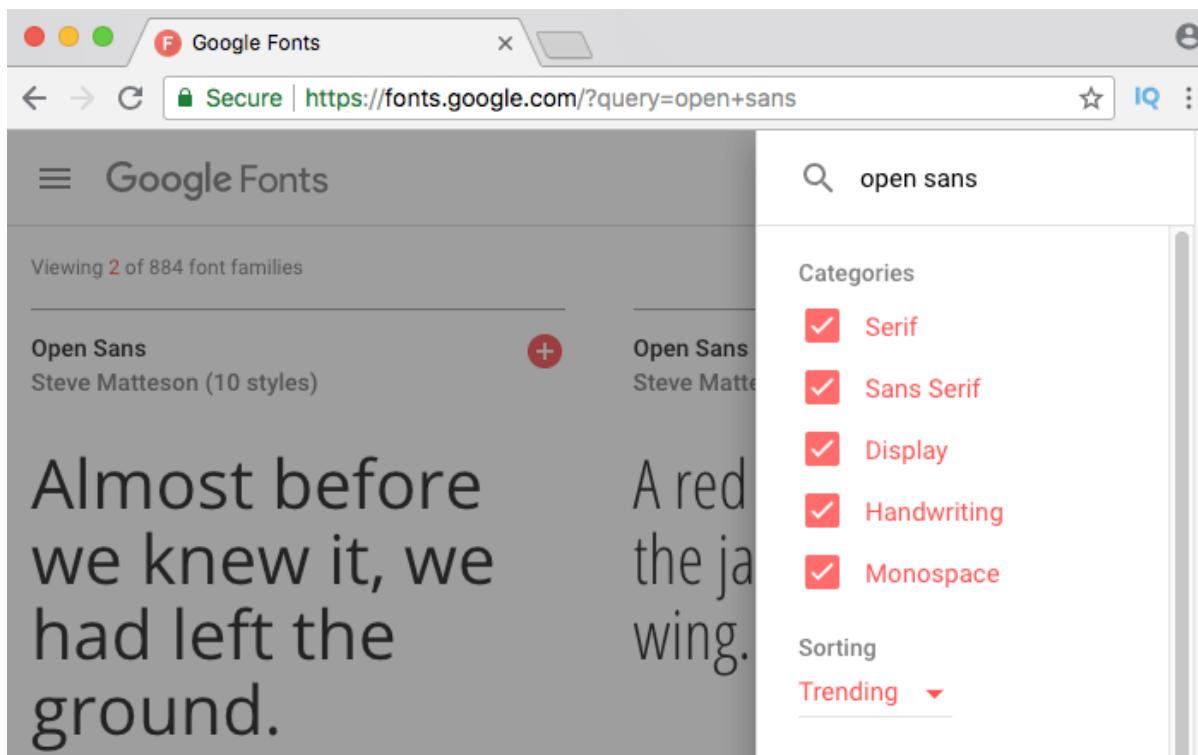
The screenshot shows the Google Fonts homepage with the following details:

- Header:** Google Fonts (with a red F icon), Secure connection (https://fonts.google.com).
- Navigation:** DIRECTORY, FEATURED, ARTICLES, ABOUT, Search bar.
- Left Sidebar:** Categories (Serif, Sans Serif, Display, Handwriting, Monospace) and Sorting (Trending).
- Right Sidebar:** Languages (All Languages), Number of styles (sliders for 0, 10, 20, 50, 100, 200, 500, 1000+), Thickness, Slant, and Width.
- Font Families:**
 - Roboto:** Christian Robertson (12 styles). Preview: "All their equipment and instruments are alive."
 - M PLUS Rounded 1c:** Coji Morishita, M+ Fonts Project (7 styles). Preview: "A red flare silhouetted the jagged edge of a wing."
 - Kosugi:** MOTOYA (1 style). Preview: "I watched the storm, so beautiful yet terrific."
 - Open Sans:** Steve Matteson (10 styles). Preview: "Almost before we knew it, we had left the ground."
- Bottom Bar:** Try typing directly into the text fields. GOT IT button.

Sur ce site, vous allez pouvoir soit rechercher une police en particulier en utilisant la recherche en haut à droite, soit pouvoir définir des critères de recherche pour trier les polices et n'afficher que certains types de polices.

Vous allez ainsi pouvoir filtrer les polices selon leur popularité pour voir quelles sont les polices les plus tendances ou encore pouvoir choisir de n'afficher les polices que d'un type de famille générique ce qui va pouvoir s'avérer très utile.

Pour ma part, je vais choisir la police « Open sans », l'une des polices les plus communément utilisées ces dernières années.



Nous allons ici soit pouvoir sélectionner directement la police en question avec les options par défaut en cliquant sur le « + », soit pouvoir cliquer sur la police en soi pour avoir davantage d'informations par rapport à cette Google Font.

Cela va également nous permettre de voir l'aspect des caractères de notre police selon l'épaisseur de celle-ci.



Characters

Styles

Light

Light Italic

Regular

Regular Italic

Semi-Bold

Semi-Bold Italic

Bold

Bold Italic

Extra-Bold

Extra-Bold Italic

On va également nous dire avec quelle autre police notre police est souvent couplée sur un site. Il est en effet reconnu comme une bonne pratique d'utiliser **deux polices différentes pour les titres et les autres contenus textuels** et la plupart des sites appliquent cette règle.

Popular Pairings with Open Sans

Roboto	+ ↕
Open Sans	Regular ↕
Roboto	Regular ↕
Lato	+ ↕
Oswald	+ ↕
Raleway	+ ↕
Montserrat	+ ↕

Ici, par exemple, on s'aperçoit que la police « Open Sans » est souvent utilisée avec une autre police bien connue qui est Roboto. Nous pouvons également à cet endroit choisir le style de notre ou de nos polices (Light, Regular, Bold, etc.).

Sélectionnons ici les polices Google « Open sans » et Roboto en version Regular. Pour cela, commencez par sélectionner la police « Open sans » en cliquant sur le « + » en haut à droite de la page puis sélectionnez de la même façon Roboto dans la liste des polices souvent utilisées ensemble. Vous devriez alors avoir un encadré en bas de page avec vos polices sélectionnées.

Popular Pairings with Open Sans

Roboto



Open Sans

Regular ▾



Roboto

Regular ▾



Lato



Oswald



Raleway



2 Families Selected



Cliquez sur cet encadré puis sur la flèche à l'intérieur.

2 Families Selected



Your Selection [Clear All](#)



Roboto

Open Sans

Vous êtes alors finalement ramené sur la page vous expliquant comment intégrer ces deux polices dans vos pages et vous donnant les codes d'intégration de ces deux polices.

Font Styles

Open Sans — Regular

Roboto — Regular

Share this selection

You are viewing a selection of fonts, including 2 typefaces in 2 styles from the Google Fonts directory. To share, copy and send the URL for this page. The URL does not expire and can be viewed by anyone.

1. Embed

To embed these fonts into a webpage, copy this code into the <head> of your HTML document.

STANDARD @IMPORT

```
<link href="https://fonts.googleapis.com/css?family=Open+Sans|Roboto" rel="stylesheet">
```

2. Specify in CSS

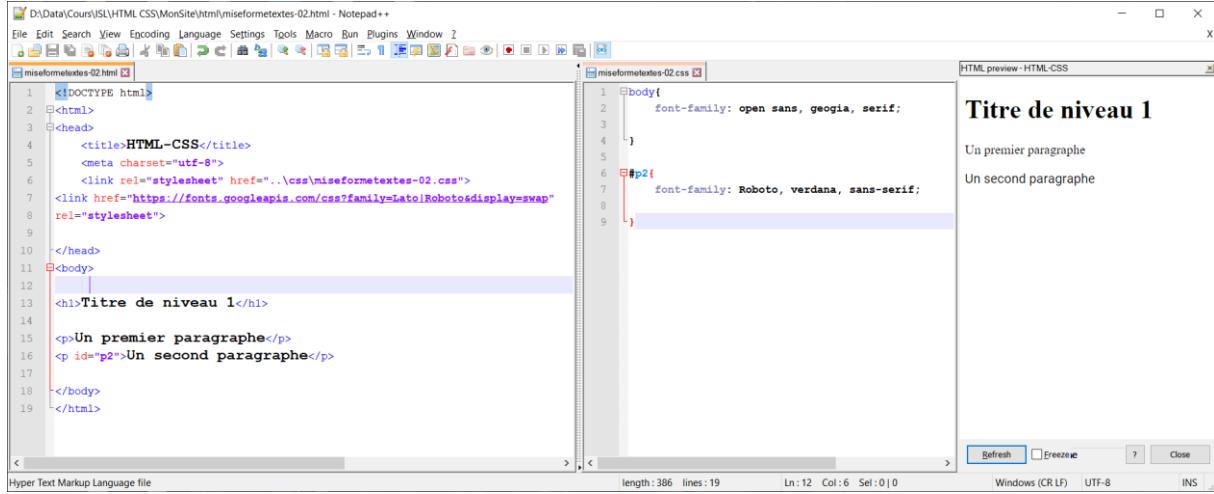
Use the following CSS properties to specify these families:

```
font-family: 'Open Sans', sans-serif;  
font-family: 'Roboto', sans-serif;
```

For examples of how fonts can be added to webpages, see the [getting started guide](#).

Comme vous pouvez le constater, la manipulation est très simple : il suffit d'ajouter dans l'élément head de notre page HTML un élément link et d'indiquer en valeur de href l'URL fournie (ici, par exemple, ce sera href="https://fonts.googleapis.com/css?family=Open+Sans|Roboto") puis de préciser finalement un attribut rel="stylesheet".

Ensuite, nous n'avons plus qu'à indiquer le nom de nos polices en valeur de notre propriété font-family dans notre feuillet CSS et le tour est joué ! Regardez plutôt l'exemple suivant dans lequel j'ai intégré mes deux Google Fonts :



The screenshot shows a Notepad++ interface with two tabs: "miseformetextes-02.html" and "miseformetextes-02.css".

HTML (miseformetextes-02.html):

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>HTML-CSS</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href=".\\css\\miseformetextes-02.css">
7   <link href="https://fonts.googleapis.com/css?family=Lato|Roboto&display=swap" rel="stylesheet">
8
9
10 </head>
11 <body>
12
13 <h1>Titre de niveau 1</h1>
14
15 <p>Un premier paragraphe</p>
16 <p id="p2">Un second paragraphe</p>
17
18 </body>
19 </html>

```

CSS (miseformetextes-02.css):

```

1 body{
2   font-family: open sans, georgia, serif;
3 }
4
5
6 #p2{
7   font-family: Roboto, verdana, sans-serif;
8 }
9

```

Preview: Titre de niveau 1
 Un premier paragraphe
 Un second paragraphe

Note : Si vous utilisez comme moi le navigateur Google Chrome il est possible que les Google Fonts soient disponibles pour vous sans que vous ayez besoin du lien d'intégration. Cependant, il faut ici penser à vos visiteurs : **tous ne vont pas utiliser Google Chrome et c'est la raison pour laquelle il est obligatoire d'insérer le lien de pré chargement des polices dans votre code via l'élément link et son attribut href.**

Les autres propriétés CSS liées à la police

Les propriétés CSS de type font- vont nous permettre de modifier l'apparence des caractères de notre police d'écriture.

Dans la leçon précédente, nous avons étudié la propriété font-family qui sert à définir une police d'écriture qui devra être utilisée pour nos textes.

Il existe d'autres propriétés CSS de type font- qu'il est bon de connaître. Dans cette nouvelle leçon, nous allons présenter les propriétés suivantes :

- La propriété **font-size** qui va nous permettre de modifier la taille de notre police ;
- La propriété **font-weight** qui va nous permettre de modifier le poids de notre police ;
- La propriété **font-style** qui va nous permettre de modifier l'inclinaison de notre police.

Notez déjà que les propriétés qui modifient le comportement des polices vont être soumises à la police qu'elles modifient : certaines polices ne supportent pas certaines modifications et les propriétés en question seront alors ignorées.

La propriété font-size

La propriété CSS font-size va nous permettre de modifier la taille de notre police d'écriture lors de l'affichage des différents textes. Cette première propriété va pouvoir s'appliquer à toutes les polices puisque toutes les polices supportent un changement de taille aujourd'hui.

Cette propriété va accepter deux grands types de valeurs : des valeurs absolues et des valeurs relatives. Chaque type de valeur va posséder des avantages et des inconvénients et on utilisera plutôt une valeur ou une autre selon la situation. Vous pouvez déjà trouver la liste des différentes valeurs acceptées ci-dessous :

```
/*Valeurs "mot clef" absolues*/
font-size: xx-small;
font-size: x-small;
font-size: small;
font-size: medium;
font-size: large;
font-size: x-large;
font-size: xx-large;

/*Valeurs "mot clef" relatives*/
font-size: larger;
font-size: smaller;

/*Exemple de valeur "longueur" absolue et fixe*/
font-size: 14px;

/*Exemple de valeurs "longueur" relatives (dynamiques)*/
font-size: 1.5ex;
font-size: 2em;
font-size: 2rem;

/*Exemple de valeur "pourcentage" (relative)*/
font-size: 80%;

/*Valeurs universelles ou globales*/
font-size: inherit;
font-size: initial;
font-size: unset;
```

Une valeur dite « absolue » est une valeur qui ne va pas dépendre d'autres éléments dans la page tandis qu'une valeur « relative » va au contraire être calculée en fonction de la valeur d'autres éléments dans la page.

Attention ici : une valeur absolue n'est pas forcément fixe ou statique puisque certaines valeurs absolues vont dépendre des réglages par défaut de l'utilisateur. « Absolu » ici veut simplement dire que la valeur ne va pas être calculée en fonction des autres éléments dans la page.

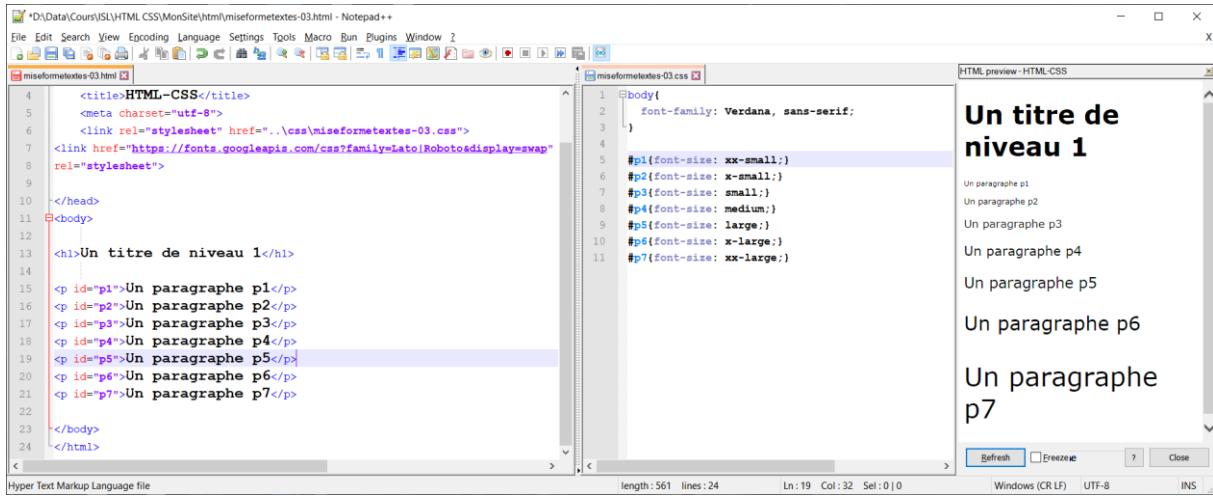
Les valeurs « mot clef »

On va donc déjà pouvoir passer un mot clef à la propriété font-size. Les mots clefs suivants correspondent à des valeurs absolues de police mais qui sont relatives aux réglages faits par l'utilisateur dans son navigateur.

Notez ici que la taille utilisée par défaut par les navigateurs (sans réglage spécifique de l'utilisateur donc) est généralement égale à 16px pour un paragraphe mais va être plus grande pour un titre.

- **xx-small** : la taille de la police sera égale à la moitié de celle définie dans le navigateur ;
- **x-small** : la taille de la police sera égale à 60% de celle définie dans le navigateur ;

- **small** : la taille de la police sera égale à 80% de celle définie dans le navigateur ;
- **medium** : la taille de la police sera égale à celle définie dans le navigateur ;
- **large** : la taille de la police sera 10% plus grande que celle définie dans le navigateur ;
- **x-large** : la taille de la police sera 50% plus grande que celle définie dans le navigateur ;
- **xx-large** : la taille de la police sera deux fois plus grande que celle définie dans le navigateur.
-



The screenshot shows a Notepad++ interface with two tabs: "miseformetextes-03.html" and "miseformetextes-03.css".

miseformetextes-03.html:

```

4 <title>HTML-CSS</title>
5 <meta charset="utf-8">
6 <link rel="stylesheet" href=".\\css\\miseformetextes-03.css">
7 <link href="https://fonts.googleapis.com/css?family=Lato|Roboto&display=swap" rel="stylesheet">
8
9 </head>
10 <body>
11 <h1>Un titre de niveau 1</h1>
12
13 <p id="p1">Un paragraphe p1</p>
14 <p id="p2">Un paragraphe p2</p>
15 <p id="p3">Un paragraphe p3</p>
16 <p id="p4">Un paragraphe p4</p>
17 <p id="p5">Un paragraphe p5</p>
18 <p id="p6">Un paragraphe p6</p>
19 <p id="p7">Un paragraphe p7</p>
20
21 </body>
22 </html>

```

miseformetextes-03.css:

```

1 body{
2   font-family: Verdana, sans-serif;
3 }
4
5 #p1{font-size: xx-small;}
6 #p2{font-size: x-small;}
7 #p3{font-size: small;}
8 #p4{font-size: medium;}
9 #p5{font-size: large;}
10 #p6{font-size: x-large;}
11 #p7{font-size: xx-large;}

```

HTML preview - HTML-CSS:

Un titre de niveau 1

Un paragraphe p1

Un paragraphe p2

Un paragraphe p3

Un paragraphe p4

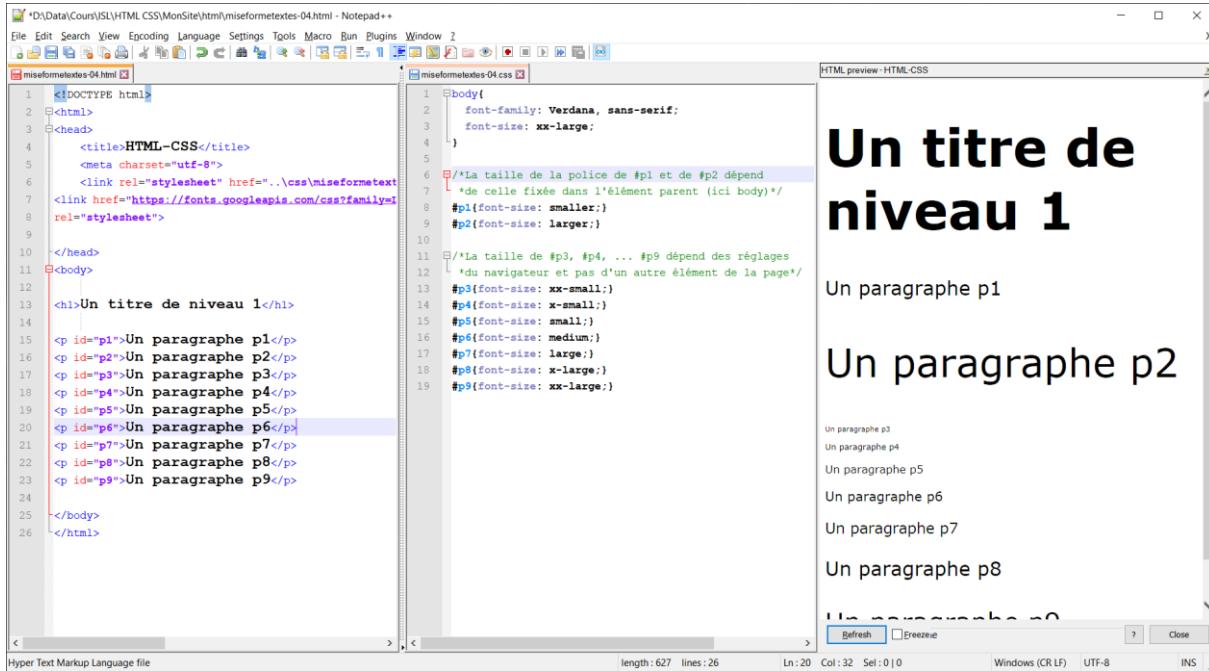
Un paragraphe p5

Un paragraphe p6

Un paragraphe p7

Les mots clefs suivants permettent de définir une taille de police relativement à celle de l'élément parent :

- **smaller** : la taille de la police de l'élément sera plus petite que celle de son élément parent ;
- **larger** : la taille de la police de l'élément sera plus grande que celle de son élément parent.



The screenshot shows a Notepad++ interface with two tabs: "miseformetextes-04.html" and "miseformetextes-04.css". The HTML tab contains a simple document structure with an h1 title and nine p paragraphs, each assigned an ID from p1 to p9. The CSS tab contains a single rule for the body element, setting its font-family to Verdana, sans-serif and its font-size to xx-large. To the right, a browser preview window displays the resulting page. The h1 title is rendered in a very large, bold font. Below it, the first two p paragraphs (p1 and p2) are also very large, while the subsequent six paragraphs (p3 through p8) are smaller, and the last one (p9) is the largest of all. This illustrates how the font size of child elements depends on the font size of their parent element.

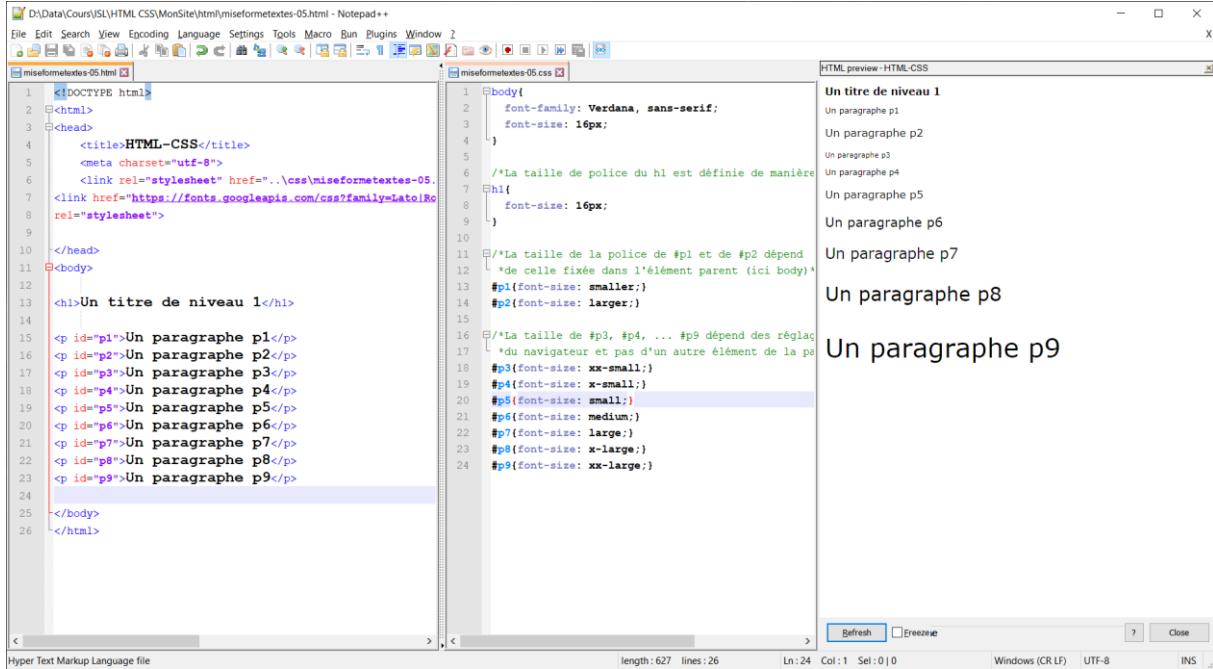
Ici, la taille de la police des paragraphes p1 et p2 dépend de la taille de la police de leur élément parent qui est dans ce cas l'élément body et qui a été réglée à font-size: xx-large.

Notez également qu'ici notre titre h1 hérite de la valeur de la propriété de son parent, c'est-à-dire font-size: xx-large et va donc s'afficher en très gros puisque le réglage par défaut des navigateurs pour un titre h1 définit généralement une taille de police deux fois plus grande que pour un paragraphe.

Les valeurs « longueur »

On va également pouvoir passer une longueur en valeur de la propriété font-size. Une longueur peut être exprimée en **px** (pixels). Dans ce cas, la taille de la police sera absolue et statiques c'est-à-dire indépendante de tout réglage de l'utilisateur ou des valeurs par défaut du navigateur.

Utiliser des valeurs en px permet donc d'avoir un résultat très prévisible mais empêche les utilisateurs d'adapter la taille du texte selon leurs réglages, ce qui va être très problématique pour les utilisateurs souffrant de déficiences visuelles.



The screenshot shows a Notepad++ interface with three tabs: 'miseformetextes-05.html', 'miseformetextes-05.css', and 'HTML preview - HTML-CSS'.

miseformetextes-05.html:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>HTML-CSS</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href=".\\css\\miseformetextes-05.css" />
7   <link href="https://fonts.googleapis.com/css?family=Lato|Raleway" rel="stylesheet">
8
9
10 </head>
11 <body>
12
13 <h1>Un titre de niveau 1</h1>
14
15 <p id="p1">Un paragraphe p1</p>
16 <p id="p2">Un paragraphe p2</p>
17 <p id="p3">Un paragraphe p3</p>
18 <p id="p4">Un paragraphe p4</p>
19 <p id="p5">Un paragraphe p5</p>
20 <p id="p6">Un paragraphe p6</p>
21 <p id="p7">Un paragraphe p7</p>
22 <p id="p8">Un paragraphe p8</p>
23 <p id="p9">Un paragraphe p9</p>
24
25 </body>
26 </html>

```

miseformetextes-05.css:

```

1 body{
2   font-family: Verdana, sans-serif;
3   font-size: 16px;
4 }
5
6 /*La taille de police du h1 est définie de manière
7 absolue*/
7 h1{
8   font-size: 16px;
9 }
10
11 /*La taille de la police de #p1 et de #p2 dépend
12 de celle fixée dans l'élément parent (ici body) */
13 #p1{font-size: smaller;}
14 #p2{font-size: larger;}
15
16 /*La taille de #p3, #p4, ... #p9 dépend des réglages
17 du navigateur et pas d'un autre élément de la page*/
18 #p3{font-size: xx-small;}
19 #p4{font-size: x-small;}
20 #p5{font-size: small;}
21 #p6{font-size: medium;}
22 #p7{font-size: large;}
23 #p8{font-size: x-large;}
24 #p9{font-size: xx-large;}

```

HTML preview - HTML-CSS:

Un titre de niveau 1
 Un paragraphe p1
 Un paragraphe p2
 Un paragraphe p3
 Un paragraphe p4
 Un paragraphe p5
 Un paragraphe p6
 Un paragraphe p7
 Un paragraphe p8
 Un paragraphe p9

Une longueur peut également être exprimée en **em** ou plus rarement en **ex**. Les valeurs en em (et en ex) vont être dynamiques et relatives à la valeur de l'élément parent pour la propriété concernée.

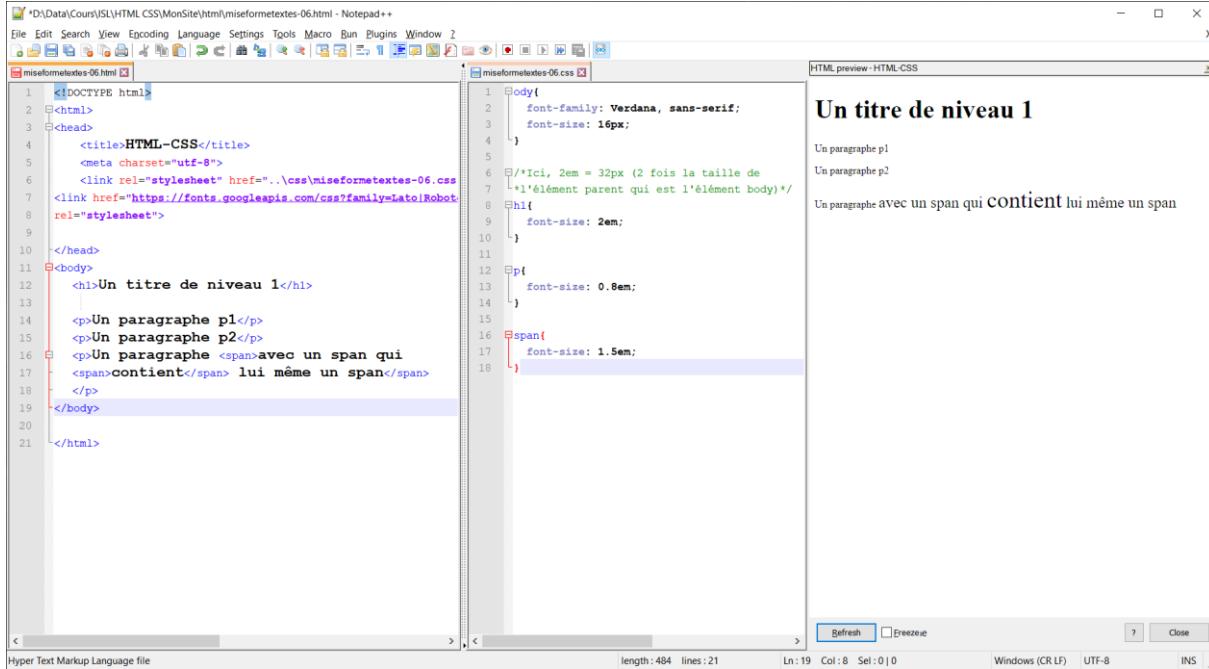
En déclarant **font-size : 1em** pour un élément on demande à ce que la valeur de la propriété soit la même que celle définie pour l'élément parent, en déclarant **font-size : 1.5em** on demande à ce que la valeur de la propriété soit égale à 1,5 fois celle définie pour l'élément parent et etc.

Note : Attention ici à bien utiliser des points à la place des virgules (notations anglo-saxonnes) lorsque vous préciser des valeurs non entières en CSS !

Si aucune taille n'a été définie pour l'élément parent alors la taille définie dans le navigateur sera utilisée (généralement 16px par défaut sauf réglage utilisateur spécifique).

Les valeurs en em possèdent donc l'avantage de pouvoir s'adapter mais peuvent entraîner des problèmes de composition ou de cohérence / d'homogénéisation. En effet, imaginons qu'un attribue un **font-size : 1.5em** pour un type d'éléments comme les éléments span par exemple.

Si dans ce cas un élément span contient lui-même un élément span, les tailles de police des différents éléments span vont être différentes puisque la police du span contenu dans l'autre span va être égale à 1,5 fois la taille du span parent.



The screenshot shows a Notepad++ interface with three panes. The left pane contains an HTML file named 'miseformetextes-06.html'. The middle pane contains a CSS file named 'miseformetextes-06.css'. The right pane is an 'HTML preview - HTML-CSS' window showing the rendered output.

```

HTML preview - HTML-CSS
Un titre de niveau 1
Un paragraphe p1
Un paragraphe p2
Un paragraphe avec un span qui
contient lui même un span

```

HTML code (miseformetextes-06.html):

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".\\css\\miseformetextes-06.css" />
7     <link href="https://fonts.googleapis.com/css?family=Lato|Roboto" rel="stylesheet">
8   </head>
9   <body>
10    <h1>Un titre de niveau 1</h1>
11    <p>Un paragraphe p1</p>
12    <p>Un paragraphe p2</p>
13    <p>Un paragraphe <span>avec un span qui
14      <span>contient</span> lui même un span</span></p>
15  </body>
16 </html>

```

CSS code (miseformetextes-06.css):

```

body {
  font-family: Verdana, sans-serif;
  font-size: 16px;
}

/* Ici, 2em = 32px (2 fois la taille de l'élément parent qui est l'élément body) */
h1 {
  font-size: 2em;
}

p {
  font-size: 0.8em;
}

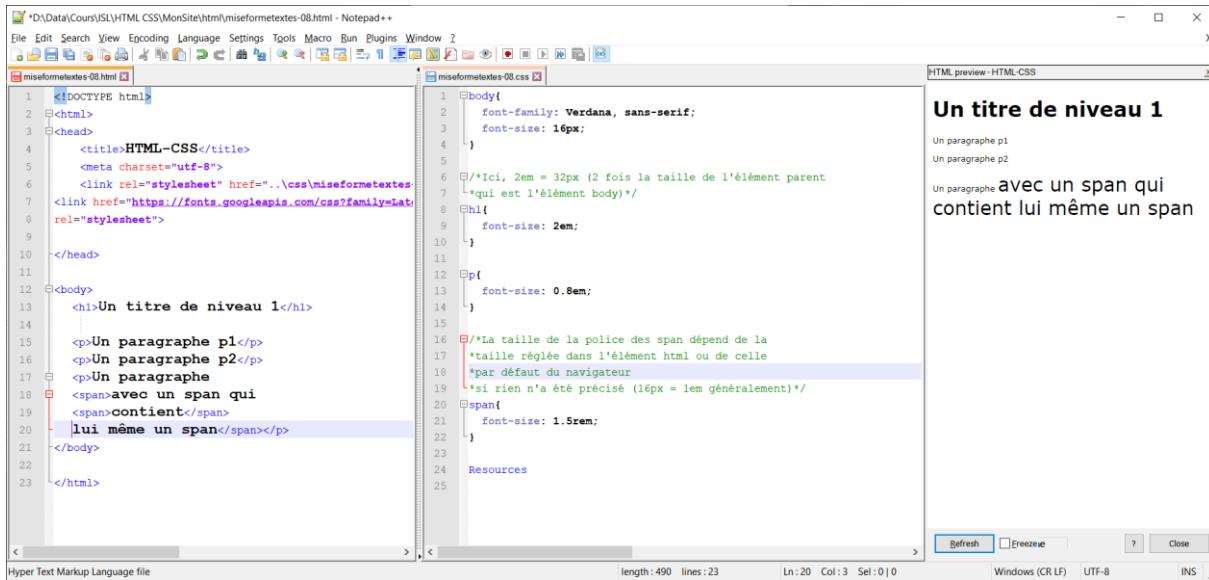
span {
  font-size: 1.5em;
}

```

Ici, la taille de la police du premier span dans notre paragraphe est égale à 1,5 fois celle du paragraphe (qui est son parent) tandis que la taille de la police du span enfant dans le span parent est égale à 1,5 fois celle du span parent.

Souvent, ce ne sera pas le comportement souhaité et ce genre de situations est ce qui rend ce type de valeurs (et les valeurs relatives en général) très difficile à manier en pratique.

Pour pallier ce problème, nous pouvons plutôt utiliser l'unité rem. Les valeurs fournies en rem vont être relatives à la taille définie pour la propriété de l'élément racine html et non pas à la taille définie pour l'élément parent.



The screenshot shows a Notepad++ interface with three panes. The left pane contains an HTML file named 'miseformetextes-08.html'. The middle pane contains a CSS file named 'miseformetextes-08.css'. The right pane is an 'HTML preview - HTML-CSS' window showing the rendered output.

```

HTML preview - HTML-CSS
Un titre de niveau 1
Un paragraphe p1
Un paragraphe p2
Un paragraphe avec un span qui
contient lui même un span

```

HTML code (miseformetextes-08.html):

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".\\css\\miseformetextes-08.css" />
7     <link href="https://fonts.googleapis.com/css?family=Lato|Roboto" rel="stylesheet">
8   </head>
9   <body>
10    <h1>Un titre de niveau 1</h1>
11    <p>Un paragraphe p1</p>
12    <p>Un paragraphe p2</p>
13    <p>Un paragraphe
14      <span>avec un span qui
15        <span>contient</span>
16        lui même un span</span></p>
17  </body>
18 </html>

```

CSS code (miseformetextes-08.css):

```

body {
  font-family: Verdana, sans-serif;
  font-size: 16px;
}

/* Ici, 2em = 32px (2 fois la taille de l'élément parent qui est l'élément body) */
h1 {
  font-size: 2em;
}

p {
  font-size: 0.8em;
}

span {
  /* La taille de la police des span dépend de la
  * taille réglée dans l'élément html ou de celle
  * par défaut du navigateur
  * si rien n'a été précisé (16px = 1em généralement) */
  font-size: 1.5rem;
}

Resources

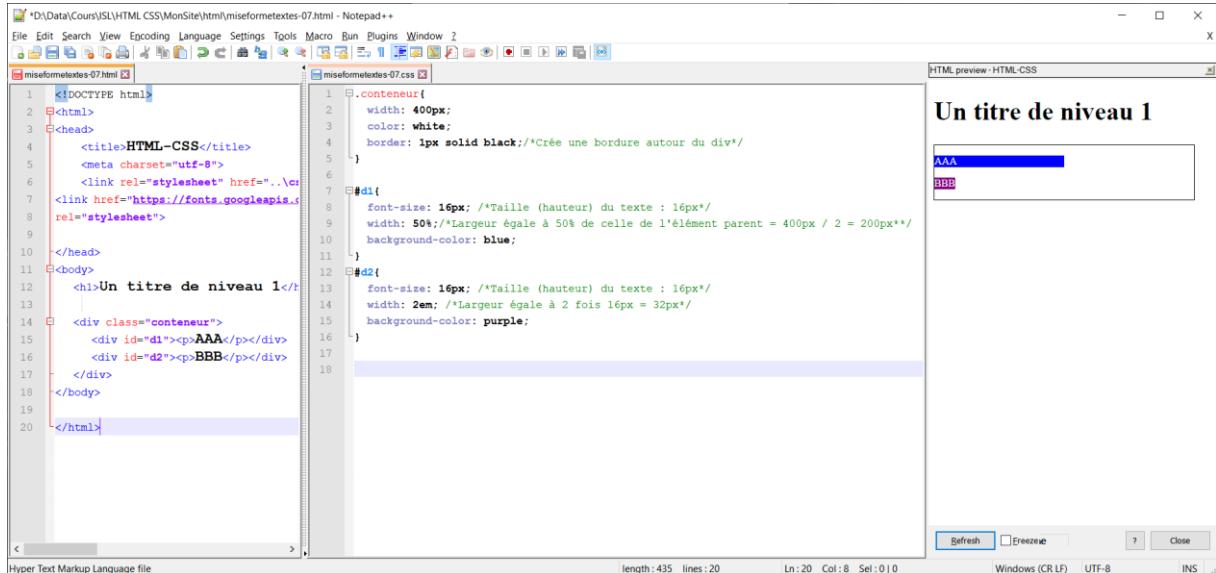
```

Les valeurs « pourcentage »

Nous allons enfin également pouvoir passer un pourcentage en valeur de la propriété font-size. Les valeurs exprimées en pourcentage vont être proportionnelles à la valeur renseignée pour la propriété dans l'élément parent.

Les unités en em et les % vont donc être équivalents pour définir la taille de la police d'un élément. Cependant, notez bien que ces unités vont pouvoir être réutilisées avec d'autres propriétés et dans ce cas elles ne vont plus l'être puisque les unités en em vont toujours être exprimées en fonction de la taille de la police.

Regardez plutôt l'exemple suivant. La propriété width sert ici à définir la largeur d'un élément et peut accepter des valeurs aussi bien en em qu'en % :



```

<!DOCTYPE html>
<html>
  <head>
    <title>HTML-CSS</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href=".><"/>
    <link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">
  </head>
  <body>
    <h1>Un titre de niveau 1</h1>
    <div class="conteneur">
      <div id="d1"><p>AAA</p></div>
      <div id="d2"><p>BBB</p></div>
    </div>
  </body>
</html>

```

```

.conteneur {
  width: 400px;
  color: white;
  border: 1px solid black; /*Crée une bordure autour du div*/
}

#d1 {
  font-size: 16px; /*Taille (hauteur) du texte : 16px*/
  width: 50%; /*Largeur égale à 50% de celle de l'élément parent = 400px / 2 = 200px*/
  background-color: blue;
}

#d2 {
  font-size: 16px; /*Taille (hauteur) du texte : 16px*/
  width: 2em; /*Largeur égale à 2 fois 16px = 32px*/
  background-color: purple;
}

```

Ici, la largeur d'un de nos éléments div est égale à deux fois la taille de la police définie dans l'élément (width : 2em) tandis que la largeur de l'autre div est égale à la moitié de la largeur de son élément parent (width : 50%).

Les valeurs universelles

On va également bien évidemment pouvoir passer une valeur universelle à la propriété font-size comme inherit ou initial par exemple pour que la propriété hérite du comportement défini dans un élément parent ou pour qu'elle soit réinitialisée à sa valeur d'origine définie dans la feuille de style par défaut du navigateur.

Note importante : les valeurs inherit, initial et unset sont des valeurs dites « globales » ou « universelles » car elles vont fonctionner avec toutes les propriétés CSS. Dans la suite de ce cours, je ne préciserai donc pas pour chaque propriété étudiée qu'elle peut utiliser ces valeurs car cela est évident.

La propriété font-weight

La propriété CSS **font-weight** va nous permettre de définir le poids d'une police, c'est-à-dire son épaisseur.

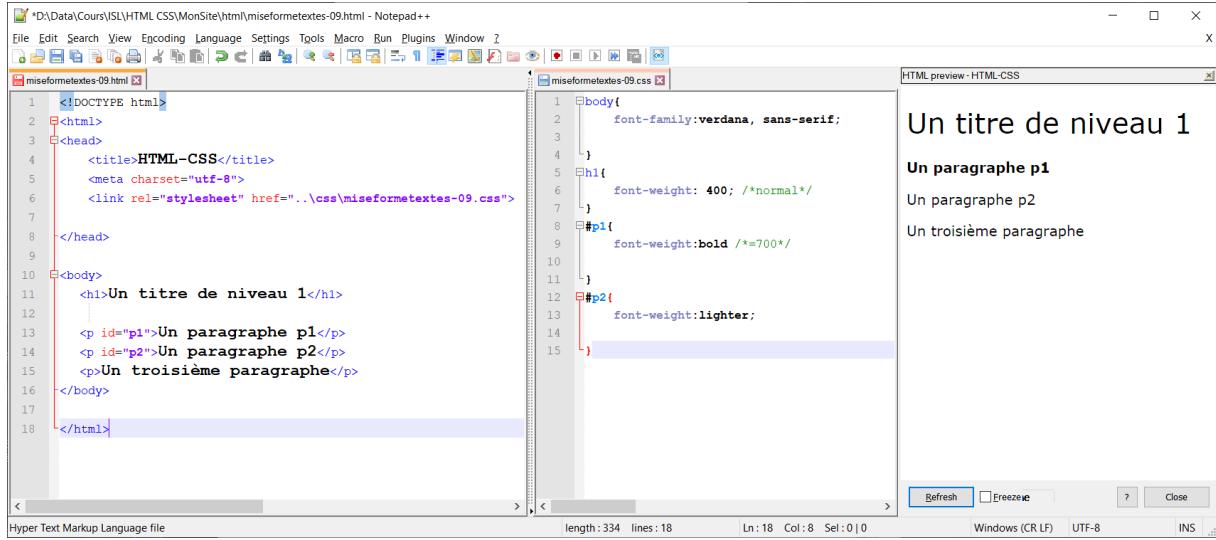
Cette propriété va pouvoir prendre un mot clef ou un chiffre exprimé en centaine(s) en valeur. Nous allons ainsi pouvoir choisir parmi les valeurs suivantes :

- Le mot clef **normal** : valeur par défaut qui correspond à un poids de police « normal » ;
- Le mot clef **lighter** qui va définir une police d'écriture plus fine que pour la valeur normal ;
- Le mot clef **bold** qui va définir une police d'écriture plus épaisse que pour la valeur normal ;
- Le mot clef **bolder** qui va définir une police d'écriture très épaisse ;
- Une centaine **entre 100 (qui correspond à une police très fine) et 900 (pour une police très épaisse)**. En termes d'équivalence avec les mots clefs, vous pouvez retenir que la valeur 400 est équivalente au mot clef normal et que la valeur 700 est équivalente au mot clef bold.

Notez bien ici que toutes les valeurs évoquées ci-dessus ne vont pas toujours pouvoir être appliquées : en effet, certaines polices ne supportent tout simplement pas de manière intrinsèque certains poids.

Dans le cas où l'on passe une valeur à la propriété font-weight qui ne peut pas être appliquée à une certaine police, alors elle sera tout simplement ignorée.

Les deux valeurs généralement bien supportées sont font-weight : normal ou font-weight : 400 et font-weight : bold ou font-weight : 700.



```

*D:\Data\Cours\ISL\HTML CSS\MonSite\Html\miseformetextes-09.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
miseformetextes-09.html
miseformetextes-09.css
HTML preview - HTML-CSS

<!DOCTYPE html>
<html>
<head>
<title>HTML-CSS</title>
<meta charset="utf-8">
<link rel="stylesheet" href=".\\css\\miseformetextes-09.css">
</head>
<body>
<h1>Un titre de niveau 1</h1>
<p id="p1">Un paragraphe p1</p>
<p id="p2">Un paragraphe p2</p>
<p>Un troisième paragraphe</p>
</body>
</html>

body{
    font-family:verdana, sans-serif;
}
h1{
    font-weight: 400; /*normal*/
}
#p1{
    font-weight:bold /*=700*/
}
#p2{
    font-weight:lighter;
}

Un titre de niveau 1
Un paragraphe p1
Un paragraphe p2
Un troisième paragraphe

```

La propriété font-style

La propriété **font-style** va nous permettre de gérer l'inclinaison de notre police.

On va pouvoir lui passer l'une des valeurs suivantes :

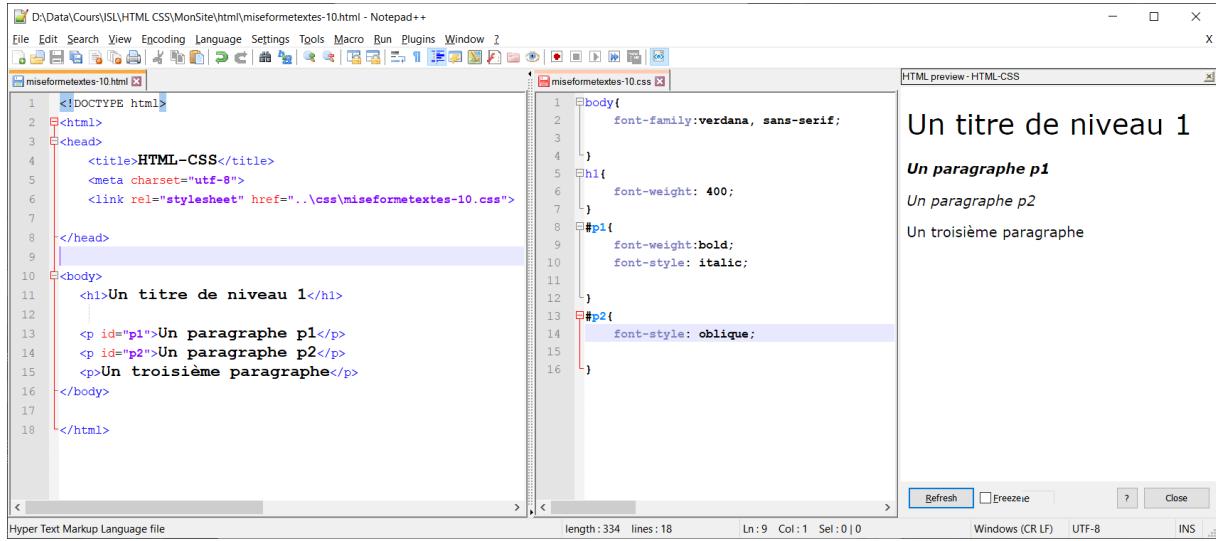
- **normal** : valeur par défaut, les caractères seront droits ;

- **italic** : la police va s'afficher en italique ;
- **oblique** : la police va être tordue pour être rendue de façon oblique.

Les valeurs italic et oblique produisent souvent un résultat très similaire. Sachez cependant qu'elles ne sont pas strictement équivalentes pour autant.

En effet, certaines polices ne supportent pas l'état italique (car elles n'ont pas été conçues pour pouvoir être rendues en italique). Dans ces cas-là, si on passe la valeur italic à la propriété font-style alors celle-ci sera simplement ignorée.

La valeur **oblique**, au contraire, va **forcer l'inclinaison** de n'importe quelles polices et même de celles qui n'ont pas été conçues pour pouvoir être rendues de façon oblique. Dans ce cas-là, cette valeur va « casser » la police afin de la rendre quand même oblique (ce qui peut amener à un rendu visuel non optimal).



The screenshot shows a Notepad++ interface with three panes. The left pane contains an HTML file named 'miseformetextes-10.html' with the following content:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".\\css\\miseformetextes-10.css">
7
8   </head>
9
10  <body>
11    <h1>Un titre de niveau 1</h1>
12
13    <p id="p1">Un paragraphe p1</p>
14    <p id="p2">Un paragraphe p2</p>
15    <p>Un troisième paragraphe</p>
16
17  </body>
18 </html>

```

The middle pane contains a CSS file named 'miseformetextes-10.css' with the following content:

```

1 body{
2   font-family:verdana, sans-serif;
3
4 }
5 h1{
6   font-weight: 400;
7 }
8 #p1{
9   font-weight:bold;
10  font-style: italic;
11 }
12 #p2{
13   font-style: oblique;
14 }
15
16

```

The right pane is a 'HTML preview - HTML-CSS' window showing the rendered output:

Un titre de niveau 1

Un paragraphe p1

Un paragraphe p2

Un troisième paragraphe

La notation raccourcie font

Les différentes propriétés de type **font-** peuvent être combinées en une seule qui représente leur notation raccourcie ou « short hand » en anglais et qui va être tout simplement la propriété **font**.

Nous reparlerons de l'utilisation des propriétés short-hand plus tard dans ce cours plus en détail.

Pour le moment, retenez simplement qu'une notation raccourcie ou « short-hand » va se définir par opposition à la version complète ou « long hand » des propriétés.

Les notations short-hand vont être un condensé de différentes propriétés et vont donc nous permettre d'écrire notre code CSS plus rapidement en déclarant d'un coup les valeurs relatives à plusieurs propriétés.

Cependant, ici, il faudra souvent respecter un ordre précis de déclarations des différentes valeurs relatives aux propriétés long-hand agrégées dans la version short-hand afin qu'il n'y ait pas d'ambiguïté sur ce à quoi va correspondre chaque valeur déclarée.

Nous allons pouvoir passer les valeurs suivantes à la notation short-hand font (dans l'ordre donné ci-après) :

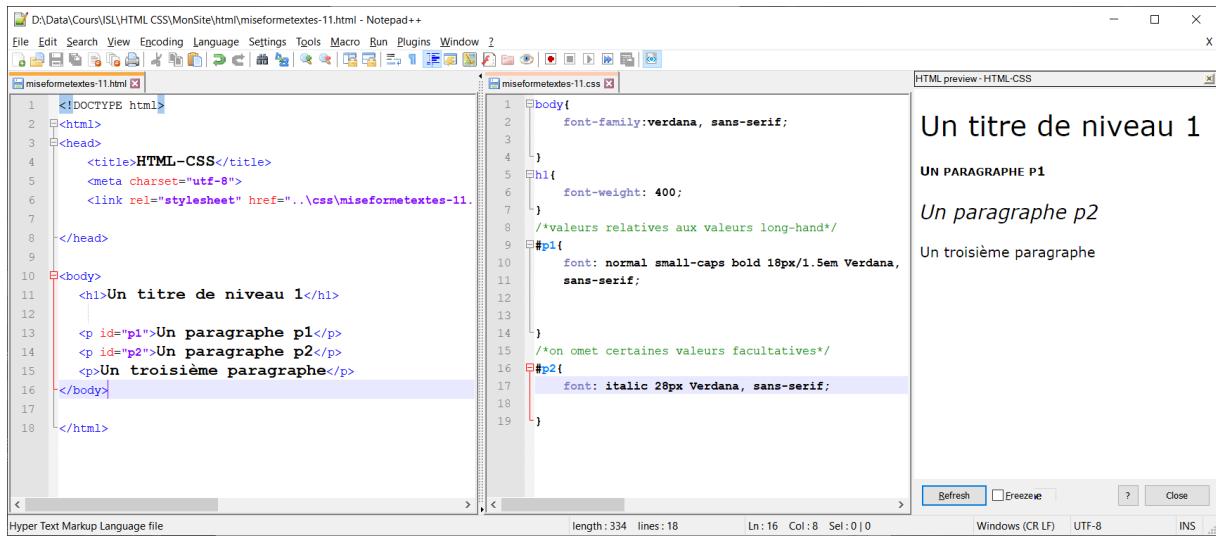
1. La valeur relative à la propriété **font-style** (facultatif) ;
2. La valeur relative à la propriété **font-variant** (facultatif) ;
3. La valeur relative à la propriété **font-weight** (facultatif) ;
4. La valeur relative à la propriété **font-size** (**obligatoire**)/**line-height** (facultatif) ;
5. La valeur relative à la propriété **font-family** (**obligatoire**).

La propriété **font-variant** n'est pas beaucoup utilisée et c'est la raison pour laquelle je n'en ai pas parlé pour le moment. Pour faire simple, elle est elle-même une notation raccourcie des sous propriétés **font-variant-caps**, **font-variant-numeric**, **font-variant-alternates**, **font-variant-ligatures** et **font-variant-east-asian** qui permettent d'utiliser des glyphes alternatifs pour les différents caractères d'une police.

La propriété **line-height** permet de définir la taille de l'espace entre les lignes (interligne). Nous étudierons cette propriété en détail plus tard dans cette partie.

Notez que pour que font fonctionne correctement, il faudra obligatoirement à minima préciser les valeurs liées aux propriétés **font-size et **font-family**.** Les autres valeurs sont facultatives et pourront être omises.

Voilà comment nous allons pouvoir utiliser cette propriété raccourcie en pratique :



The screenshot shows a Notepad++ interface with three windows:

- miseformetextes-11.html**: Contains the following HTML code:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".\\css\\miseformetextes-11.css" />
7   </head>
8   <body>
9     <h1>Un titre de niveau 1</h1>
10    <p id="p1">Un paragraphe p1</p>
11    <p id="p2">Un paragraphe p2</p>
12    <p>Un troisième paragraphe</p>
13  </body>
14 </html>

```
- miseformetextes-11.css**: Contains the following CSS code:

```

1 body {
2   font-family: verdana, sans-serif;
3 }
4 h1 {
5   font-weight: 400;
6   /*valeurs relatives aux valeurs long-hand*/
7 }
8 #p1 {
9   font: normal small-caps bold 18px/1.5em Verdana,
10   sans-serif;
11 }
12 /*on omet certaines valeurs facultatives*/
13 #p2 {
14   font: italic 28px Verdana, sans-serif;
15 }
16
17
18
19

```
- HTML preview - HTML-CSS**: Shows the rendered output:
 - Un titre de niveau 1**
 - UN PARAGRAPHE P1**
 - Un paragraphe p2**
 - Un troisième paragraphe**

Expliquons rapidement le code ci-dessus. Ma première définition de font liée au sélecteur **#p1** est complète : j'ai passé les valeurs relatives à toutes les propriétés décrites ci-dessus et celles-ci vont donc être appliquées à l'élément portant l'id="p1". Notez que les valeurs relatives aux propriétés **font-size** et **line-height** doivent être séparées par un slash.

Dans mon sélecteur `#p2`, cependant, j'ai omis certaines valeurs qui sont facultatives et n'ai passé que les valeurs relatives aux propriétés **font-style**, **font-size** et **font-family**.

Vous devriez avoir une question par rapport à cette deuxième déclaration : comment savoir à quelle propriété correspond une valeur lorsque celle-ci n'est pas unique à la propriété ?

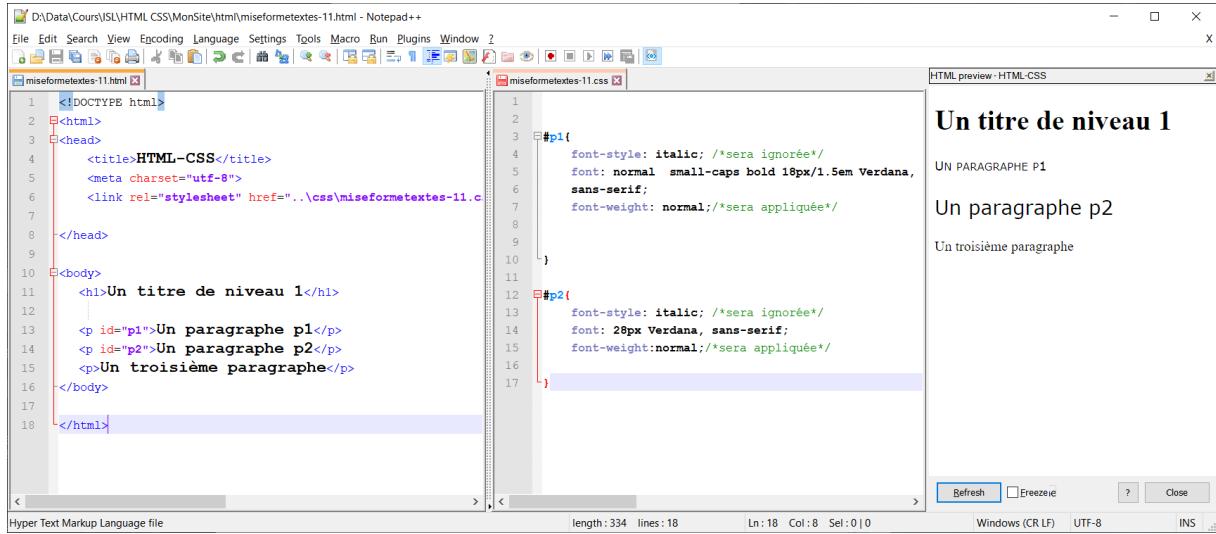
Imaginons ici que j'ai écrit `normal` à la place de `italic`. Comment savoir si cette valeur `normal` a été déclarée comme valeur pour ma propriété `font-style`, `font-variant` ou `font-weight` ?

Le CSS va en fait lire l'ensemble des valeurs fournies à la notation short hand font et, s'il y a ambiguïté, les valeurs fournies vont être liées à la première propriété qui l'accepte. C'est tout l'intérêt d'avoir défini un ordre des valeurs et de le respecter. Ainsi, dans le cas présent, comme font-style accepte la valeur normal, celle-ci lui sera attribuée.

Notez qu'ici ce n'est pas impactant puisque dans tous les cas la valeur `normal` est la valeur par défaut pour les trois propriétés `font-style`, `font-variant` ou `font-weight` mais ça pourra l'être dans d'autres cas où des propriétés partagent des valeurs en commun qui ne sont pas leurs valeurs par défaut.

Notez également **qu'il est généralement déconseillé d'utiliser à la fois les notations short-hand et long-hand** pour définir le comportement des mêmes propriétés car cela peut mener à de nombreux problèmes dans le code.

Si pour une raison ou une autre vous devez redéfinir vos propriétés de cette manière, retenez bien que pour toute notation long-hand déclarée avant la version short-hand associée sera ignorée même dans le cas où la version short-hand ne définit pas de comportement pour la propriété long-hand en question de manière explicite.



The screenshot shows the Notepad++ interface with three windows:

- HTML file (miseformetextes-11.html):**

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>HTML-CSS</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="..\css\miseformetextes-11.css" />
7 
8 </head>
9 
10 <body>
11   <h1>Un titre de niveau 1</h1>
12 
13   <p id="p1">Un paragraphe p1</p>
14   <p id="p2">Un paragraphe p2</p>
15   <p>Un troisième paragraphe</p>
16 
17 </body>
18 </html>
```
- CSS file (miseformetextes-11.css):**

```

1 
2 
3 #p1{
4   font-style: italic; /*sera ignorée*/
5   font: normal small-caps bold 18px/1.5em Verdana,
6   sans-serif;
7   font-weight: normal; /*sera appliquée*/
8 }
9 
10 
11 #p2{
12   font-style: italic; /*sera ignorée*/
13   font: 28px Verdana, sans-serif;
14   font-weight: normal; /*sera appliquée*/
15 }
16 
17 }
```
- HTML preview - HTML-CSS:**

Un titre de niveau 1

UN PARAGRAPHE P1

Un paragraphe p2

Un troisième paragraphe

Les propriétés CSS liées au texte

Les propriétés liées au texte ne vont pas nous permettre de modifier la forme des caractères de la police mais plutôt d'ajouter des effets autour de notre texte ou de le formater dans la page d'une manière ou d'une autre.

Dans cette nouvelle leçon, nous allons présenter et étudier quelques-unes des propriétés CSS de type **text-** les plus utiles et les plus intéressantes et notamment :

- La propriété **text-align** qui va nous permettre de gérer l'alignement du texte ;
- La propriété **text-transform** qui va nous permettre de gérer la casse du texte (le fait que le texte soit en majuscules ou en minuscules) ;
- La propriété **text-decoration** qui va nous permettre d'ajouter des éléments de décoration autour du texte comme un trait de soulignement par exemple ;
- La propriété **text-indent** qui va nous permettre de définir l'indentation d'un texte ;
- La propriété **text-shadow** qui va nous permettre d'ajouter des ombres autour d'un texte.

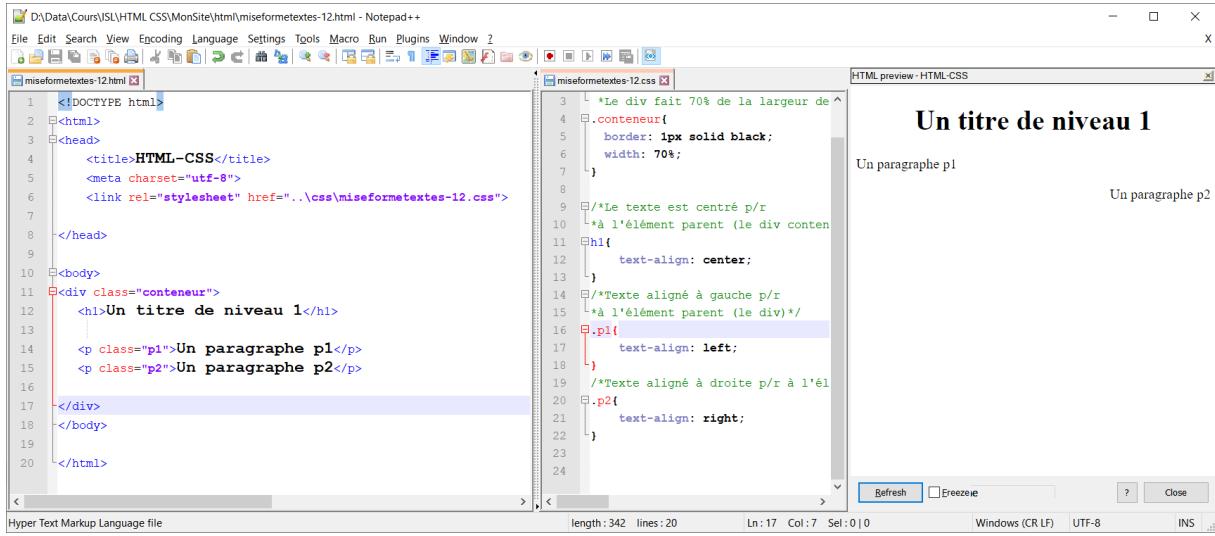
La propriété **text-align**

La propriété **text-align** va nous permettre de définir l'alignement d'un texte par rapport à son élément conteneur.

Nous allons pouvoir choisir parmi les valeurs d'alignement suivantes :

- **left** : valeur par défaut. Le texte sera aligné contre le bord gauche de l'élément qui le contient ;
- **center** : Le texte sera centré dans l'élément qui le contient ;
- **right** : Le texte sera aligné contre le bord droite de l'élément qui le contient ;
- **justify** : Le texte sera justifié (les écarts entre les mots vont être calculés de façon à ce que chaque ligne de texte occupe la même largeur).

*Notez bien que la propriété **text-align** va toujours aligner le texte dans / par rapport à l'élément qui le contient. Ainsi, appliquer cette propriété à un élément de type *inline* comme un *span* par exemple n'aurait aucun sens puisque ce type d'élément possède la même taille que son contenu.*



The screenshot shows a Notepad++ interface with two tabs: "miseformetextes-12.html" and "miseformetextes-12.css". The HTML tab contains the following code:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".\\css\\miseformetextes-12.css">
7   </head>
8   <body>
9     <div class="conteneur">
10       <h1>Un titre de niveau 1</h1>
11       <p class="p1">Un paragraphe p1</p>
12       <p class="p2">Un paragraphe p2</p>
13     </div>
14   </body>
15 </html>

```

The CSS tab contains the following code:

```

3   /*Le div fait 70% de la largeur de l'élement parent (le div conteneur) à l'élément parent (le div contenueur*/
4   .conteneur{
5     border: 1px solid black;
6     width: 70%;
7   }
8   /*Le texte est centré p/r à l'élément parent (le div conteneur)*/
9   h1{
10   text-align: center;
11 }
12 /*Texte aligné à gauche p/r à l'élément parent (le div)*
13 .p1{
14   text-align: left;
15 }
16 /*Texte aligné à droite p/r à l'élément parent (le div)*
17 .p2{
18   text-align: right;
19 }
20
21
22
23
24

```

A preview window titled "HTML preview - HTML-CSS" shows the rendered output:

Un titre de niveau 1

Un paragraphe p1

Un paragraphe p2

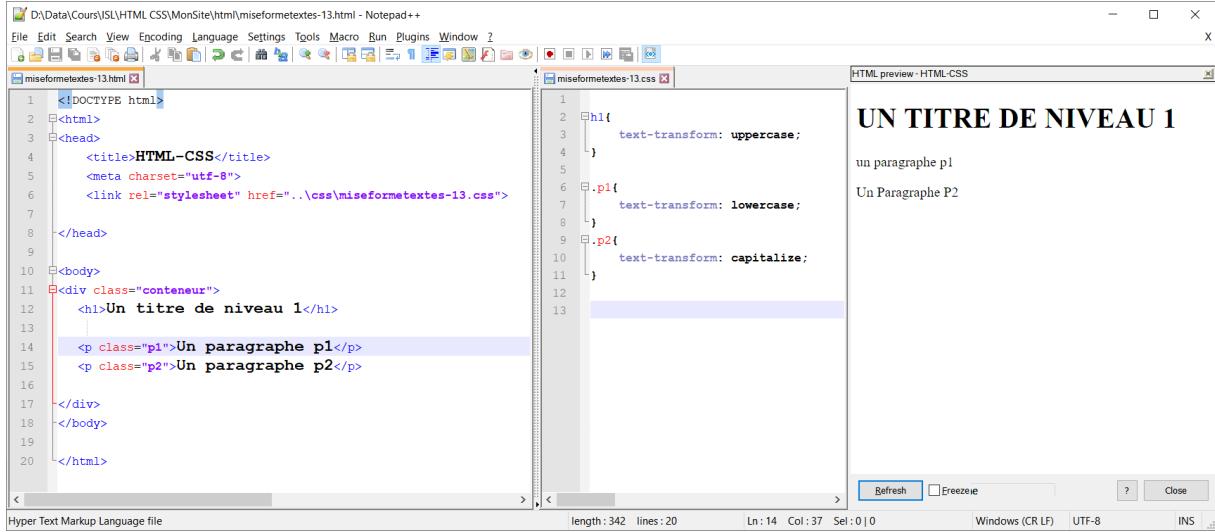
La propriété text-transform

La propriété **text-transform** va nous permettre de modifier la casse d'un texte ou de certaines parties d'un texte, c'est-à-dire de gérer le fait qu'un texte ou qu'une partie de texte s'affiche en majuscules ou en minuscules.

Cette propriété peut être utile dans le cas de textes générés automatiquement ou dans les cas où nous n'avons pas accès ou ne pouvons pas modifier le texte dans le code HTML directement.

Nous allons pouvoir passer l'une des valeurs suivantes à text-transform :

- **none** : Valeur par défaut. Pas de transformation du texte. Utile pour annuler une transformation transmise par héritage par exemple ;
- **lowercase** : Transforme tout le texte d'un élément en minuscules ;
- **uppercase** : Transforme tout le texte d'un élément en majuscules ;
- **capitalize** : Transforme la première lettre de chaque mot en majuscule ;



The screenshot shows a Notepad++ interface with two tabs: "miseformetextes-13.html" and "miseformetextes-13.css". The HTML tab contains the following code:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".\\css\\miseformetextes-13.css">
7
8   </head>
9
10  <body>
11    <div class="conteneur">
12      <h1>Un titre de niveau 1</h1>
13
14      <p class="p1">Un paragraphe p1</p>
15      <p class="p2">Un paragraphe p2</p>
16
17    </div>
18  </body>
19
20 </html>

```

The CSS tab contains the following code:

```

1 h1{
2   text-transform: uppercase;
3 }
4
5 .p1{
6   text-transform: lowercase;
7 }
8
9 .p2{
10   text-transform: capitalize;
11 }
12
13

```

The preview window on the right shows the rendered output:

UN TITRE DE NIVEAU 1

un paragraphe p1

Un Paragraphe P2

La propriété text-decoration

La propriété **text-decoration** va nous permettre d'ajouter des décos à un texte, comme un trait de soulignement ou de surlignement par exemple.

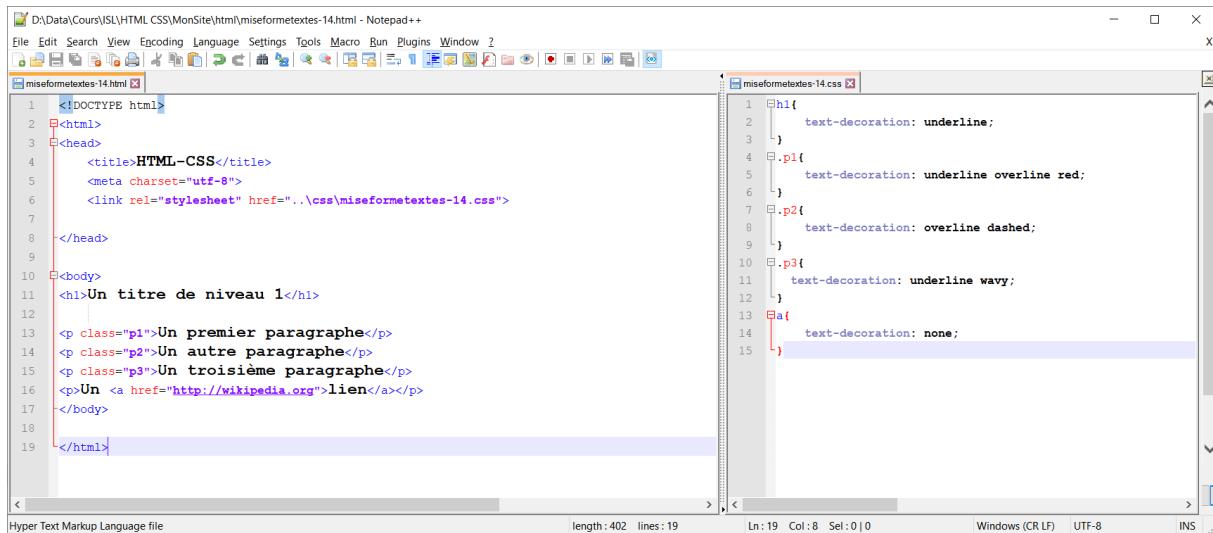
On va pouvoir lui passer jusqu'à trois valeurs qui vont correspondre au type de décoration (valeur obligatoire), à la couleur de la décoration (valeur facultative, couleur actuelle utilisée par défaut) et au style de la décoration (solide par défaut).

Concernant le type de décoration, nous allons pouvoir choisir parmi les valeurs :

- **underline** : ajoute un trait de soulignement au texte ;
- **overline** : ajoute un trait de surlignement au texte ;
- **line-through** : ajoute un trait qui va barrer le texte ;
- **underline overline** : ajoute un trait de soulignement et un trait de surlignement au texte.

Concernant le style de la décoration, nous allons pouvoir choisir parmi les valeurs :

- **solid** : valeur par défaut ; le trait de décoration sera solide ;
- **double** : le trait de décoration sera double ;
- **dotted** : le trait de décoration sera en pointillés ;
- **dashed** : le trait de décoration sera en tirets ;
- **wavy** : le trait de décoration sera courbé.



D:\Data\Cours\ISL\HTML CSS\MonSite\html\miseformetextes-14.html - Notepad++

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>HTML-CSS</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href=".\\css\\miseformetextes-14.css">
7
8 </head>
9
10 <body>
11   <h1>Un titre de niveau 1</h1>
12
13   <p class="p1">Un premier paragraphe</p>
14   <p class="p2">Un autre paragraphe</p>
15   <p class="p3">Un troisième paragraphe</p>
16   <p>Un lien <a href="http://wikipedia.org">lien</a></p>
17 </body>
18
19 </html>

```

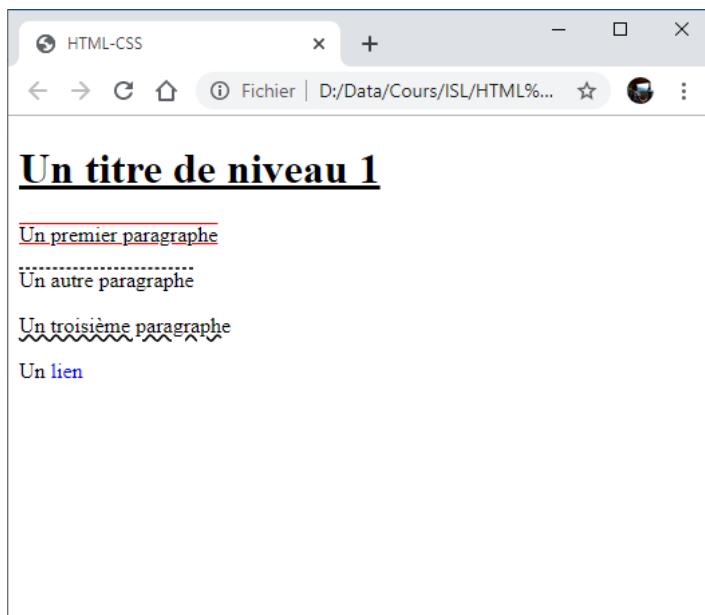
miseformetextes-14.css

```

1 h1{
2   text-decoration: underline;
3 }
4 .p1{
5   text-decoration: underline overline red;
6 }
7 .p2{
8   text-decoration: overline dashed;
9 }
10 .p3{
11   text-decoration: underline wavy;
12 }
13 a{
14   text-decoration: none;
15 }

```

Hyper Text Markup Language file length: 402 lines: 19 Ln: 19 Col: 8 Sel: 0 | 0 Windows (CR LF) UTF-8 INS



Notez également qu'à l'heure actuelle la définition de la propriété `text-decoration` est en train d'évoluer puisque de nouvelles sous propriétés CSS nous permettant de gérer un aspect de la décoration à la fois ont été dernièrement proposées. Si cette nouvelle définition est acceptée, alors la propriété `text-decoration` deviendra la notation raccourcie des propriétés suivantes :

- `text-decoration-line` qui permet de définir une décoration autour du texte ;
- `text-decoration-color` qui permet de choisir la couleur de la décoration (noire par défaut) ;
- `text-decoration-style` qui permet de choisir le style de la décoration (solide par défaut).

La propriété `text-decoration-line` va accepter les valeurs suivantes :

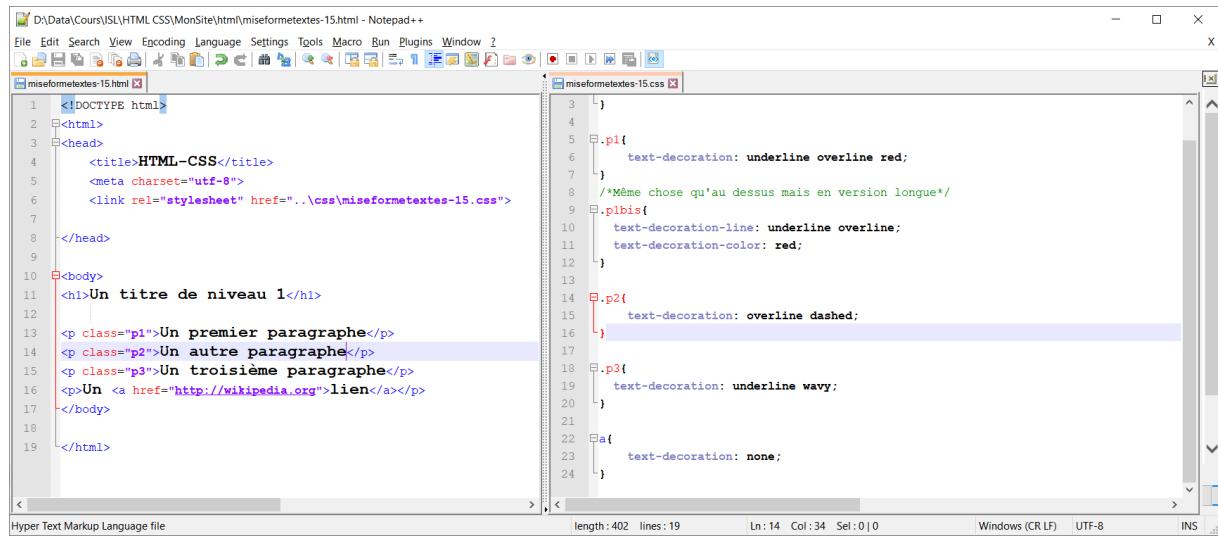
- `underline` : ajoute un trait de soulignement au texte ;
- `overline` : ajoute un trait de surlignement au texte ;

- **line-through** : ajoute un trait qui va barrer le texte ;
- **underline overline** : ajoute un trait de soulignement et un trait de surlignement au texte.

La propriété `text-decoration-style` va accepter les valeurs suivantes :

- **solid** : valeur par défaut ; le trait de décoration sera solide ;
- **double** : le trait de décoration sera double ;
- **dotted** : le trait de décoration sera en pointillés ;
- **dashed** : le trait de décoration sera en tirets ;
- **wavy** : le trait de décoration sera courbé.

A noter cependant que pour le moment ces propriétés ne sont pas encore officielles et vont pas encore forcément supportées par tous les navigateurs. Elles devraient cependant bientôt le devenir et c'est pourquoi je les mentionne dans ce cours.



The screenshot shows two tabs in Notepad++: "miseformetextes-15.html" and "miseformetextes-15.css".

miseformetextes-15.html:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".\\css\\miseformetextes-15.css">
7
8   </head>
9
10  <body>
11    <h1>Un titre de niveau 1</h1>
12
13    <p class="p1">Un premier paragraphe</p>
14    <p class="p2">Un autre paragraphe</p>
15    <p class="p3">Un troisième paragraphe</p>
16    <p>Un <a href="http://wikipedia.org">lien</a></p>
17
18  </body>
19
20 </html>

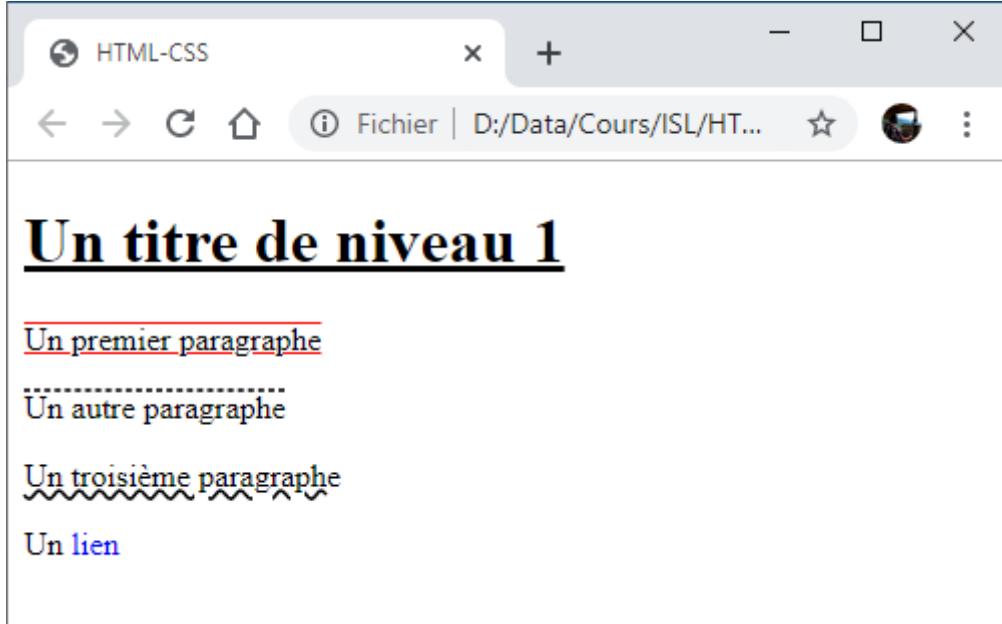
```

miseformetextes-15.css:

```

3   }
4
5   .p1{
6     text-decoration: underline overline red;
7   }
8   /*Même chose qu'au dessus mais en version longue*/
9   .p1bis{
10     text-decoration-line: underline overline;
11     text-decoration-color: red;
12   }
13
14   .p2{
15     text-decoration: overline dashed;
16   }
17
18   .p3{
19     text-decoration: underline wavy;
20   }
21
22   a{
23     text-decoration: none;
24   }

```



Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

Un troisième paragraphe

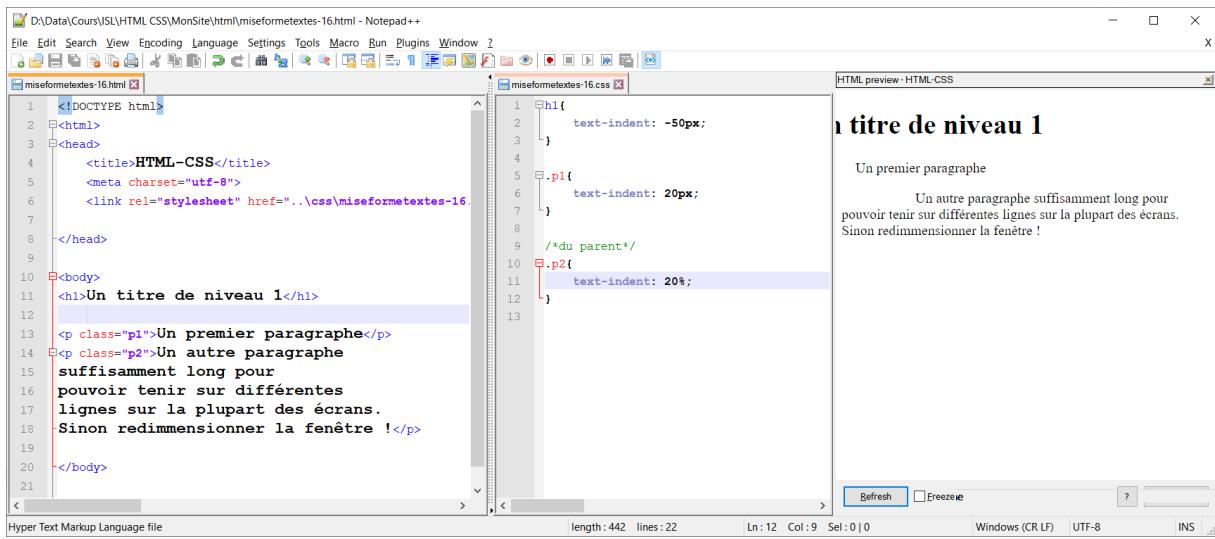
Un lien

La propriété text-indent

La propriété **text-indent** va nous permettre de préciser l'indentation c'est-à-dire le retrait de la première ligne d'un texte par rapport au bord de l'élément parent.

Nous allons ici pouvoir passer une valeur en **px**, **em**, **%**, etc. Les valeurs en **%** vont être exprimées en fonction de la largeur de l'élément parent.

Notez que les valeurs négatives sont acceptées.



```
miseformetextes-16.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="../../css/miseformetextes-16.css" />
7   </head>
8
9   <body>
10    <h1>Un titre de niveau 1</h1>
11
12    <p class="p1">Un premier paragraphe</p>
13    <p class="p2">Un autre paragraphe suffisamment long pour pouvoir tenir sur différentes lignes sur la plupart des écrans. Sinon redimensionner la fenêtre !</p>
14
15
16
17
18
19
20
21

miseformetextes-16.css
1 h1{
2   text-indent: -50px;
3 }
4
5 .p1{
6   text-indent: 20px;
7 }
8
9 /*du parent*/
10 .p2{
11   text-indent: 20%;
12 }
```

HTML preview - HTML-CSS

Un titre de niveau 1

Un premier paragraphe

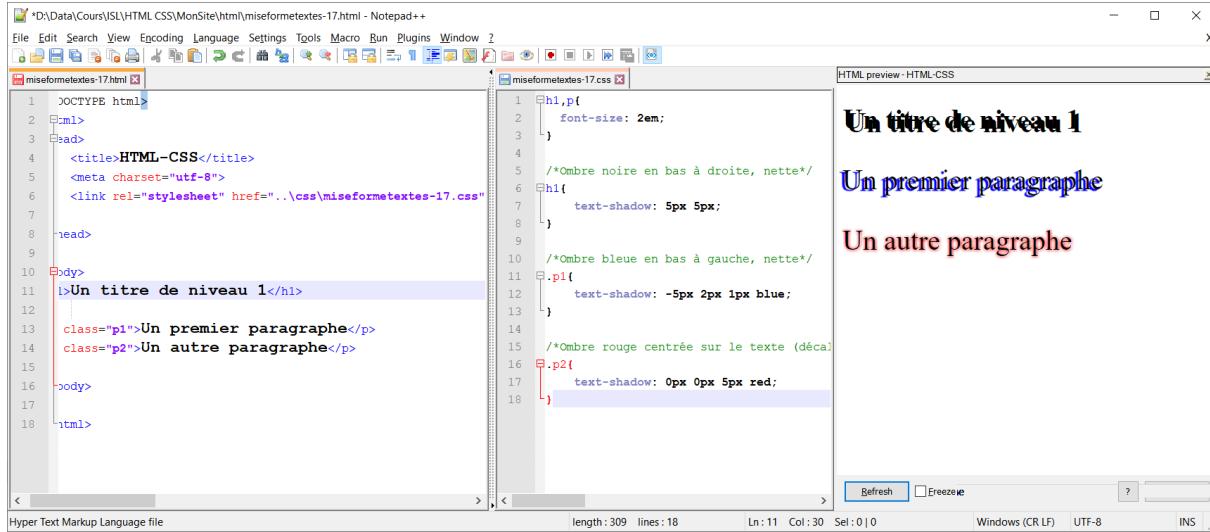
Un autre paragraphe suffisamment long pour pouvoir tenir sur différentes lignes sur la plupart des écrans. Sinon redimensionner la fenêtre !

La propriété text-shadow

La propriété **text-shadow** va nous permettre d'ajouter des ombres autour de nos textes. Cette propriété est relativement complexe à maîtriser puisqu'elle va pouvoir utiliser jusqu'à 4 valeurs pour définir précisément une ombre et surtout puisqu'on va pouvoir appliquer plusieurs ombres différentes à un même texte grâce à elle.

Les 4 valeurs de **text-shadow** vont correspondre :

1. Au **déplacement** (ou « projection ») **horizontal** de l'ombre par rapport au texte. En passant une **valeur positive**, l'ombre est **projeté à droite** du texte. En passant une valeur négative, l'ombre est projetée à gauche de celui-ci. Cette valeur doit obligatoirement être renseignée ;
2. Au **déplacement vertical** de l'ombre par rapport au texte. En passant une valeur positive, l'ombre est **projeté sous le texte**. En passant une valeur négative, l'ombre est projetée au-dessus de celui-ci. Cette valeur doit obligatoirement être renseignée ;
3. Au **rayon de flou** de l'ombre. Un flou Gaussien est utilisé ici : plus la valeur est grande, plus l'ombre sera étendue et floue. Cette valeur est facultative ;
4. A la **couleur de l'ombre**. On va ici pouvoir passer toutes les valeurs de couleurs (que nous étudierons plus tard) disponibles.

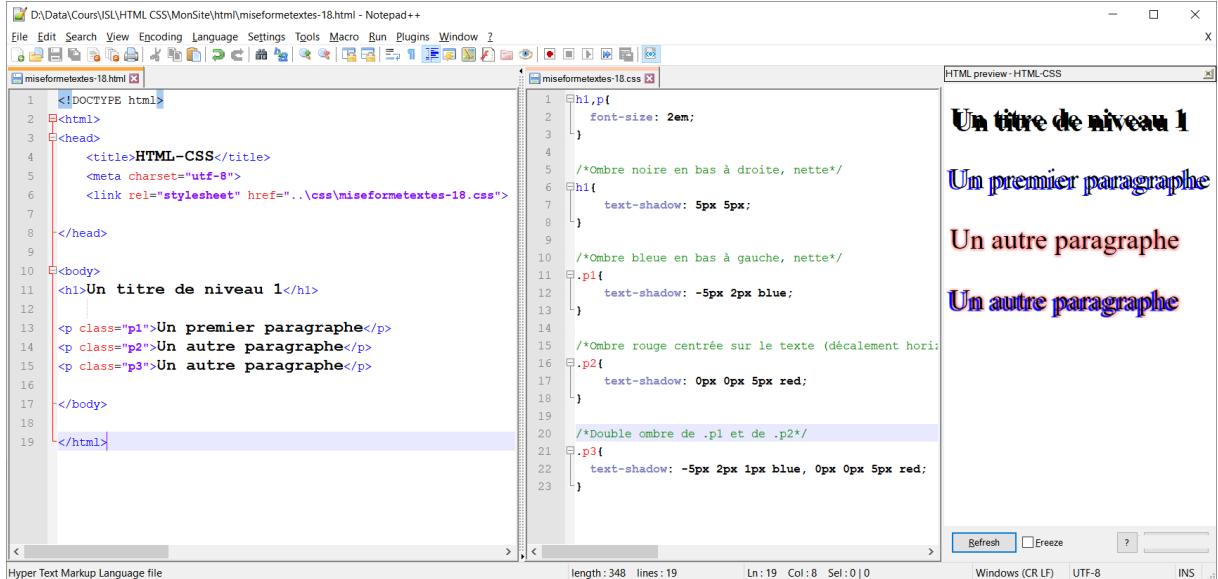


```
*DA\DATA\COURS\ISL\HTML CSS\MonSite\html\miseformetextes-17.html - Notepad+
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
miseformetextes-17.html
miseformetextes-17.css
HTML preview - HTML-CSS
Un titre de niveau 1
Un premier paragraphe
Un autre paragraphe

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".\\css\\miseformetextes-17.css">
7   </head>
8
9   <body>
10    <h1>Un titre de niveau 1</h1>
11
12    <p class="p1">Un premier paragraphe</p>
13    <p class="p2">Un autre paragraphe</p>
14
15  </body>
16
17 </html>
18

1 <h1,p{
2   font-size: 2em;
3 }
4
5 /*Ombre noire en bas à droite, nette*/
6 h1{
7   text-shadow: 5px 5px;
8 }
9
10 /*Ombre bleue en bas à gauche, nette*/
11 .p1{
12   text-shadow: -5px 2px 1px blue;
13 }
14
15 /*Ombre rouge centrée sur le texte (décalée)
16 .p2{
17   text-shadow: 0px 0px 5px red;
18 }
```

En plus de cela, nous allons pouvoir définir plusieurs ombres différentes pour un même texte. Pour faire cela, il suffit de séparer les différentes déclarations relatives à chaque ombre par une virgule dans la propriété **text-shadow**.



The screenshot shows a Notepad++ interface with two files open: "miseformetextes-18.html" and "miseformetextes-18.css".

miseformetextes-18.html:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>HTML-CSS</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href=".\\css\\miseformetextes-18.css">
7 
8 </head>
9 
10 <body>
11   <h1>Un titre de niveau 1</h1>
12 
13   <p class="p1">Un premier paragraphe</p>
14   <p class="p2">Un autre paragraphe</p>
15   <p class="p3">Un autre paragraphe</p>
16 
17 </body>
18 
19 </html>

```

miseformetextes-18.css:

```

1 h1,p{
2   font-size: 2em;
3 }
4 
5 /*Ombre noire en bas à droite, nette*/
6 h1{
7   text-shadow: 5px 5px;
8 }
9 
10 /*Ombre bleue en bas à gauche, nette*/
11 .p1{
12   text-shadow: -5px 2px blue;
13 }
14 
15 /*Ombre rouge centrée sur le texte (décalage hori*/
16 .p2{
17   text-shadow: 0px 0px 5px red;
18 }
19 
20 /*Double ombre de .p1 et de .p2*/
21 .p3{
22   text-shadow: -5px 2px 1px blue, 0px 0px 5px red;
23 }

```

HTML preview - HTML-CSS:

- Un titre de niveau 1
- Un premier paragraphe
- Un autre paragraphe
- Un autre paragraphe

Comme vous pouvez le voir ci-dessus, il faut faire bien attention avec cette propriété à ne pas rendre le texte illisible pour vos visiteurs ! Pour cela, nous allons pouvoir ajouter de la transparence à nos ombres en utilisant par exemple une notation RGBA pour la couleur. Nous reviendrons sur ces notations dans le chapitre de ce cours dédié à la couleur.

Gérer la taille des interlignes et des espaces dans les textes en CSS

Dans cette nouvelle leçon, nous allons étudier trois propriétés CSS qui vont nous permettre de gérer l'espace entre chaque lettre, entre chaque mot ainsi qu'entre chaque ligne de texte.

Ces propriétés sont les suivantes :

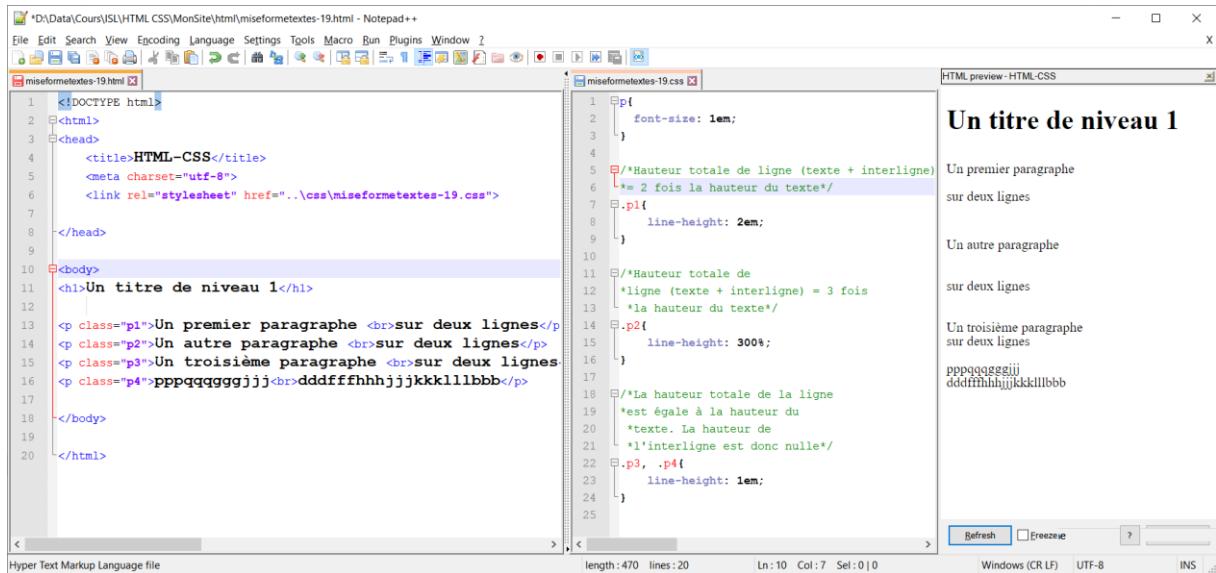
- La propriété **line-height** ;
- La propriété **letter-spacing** ;
- La propriété **word-spacing**.

La propriété line-height

La propriété **line-height** va nous permettre de définir la hauteur de chaque ligne d'un texte et donc de fait l'espace entre les lignes.

Cette propriété va pouvoir accepter en valeur un nombre simple, une unité de longueur en **px**, **em** etc. ou un pourcentage.

Les valeurs nombre simple et pourcentage vont nous permettre de définir la hauteur totale de la ligne par rapport à la taille de la police. Par exemple, indiquer **line-height : 2** ou **line-height : 200%** signifie que chaque ligne de notre élément aura une hauteur totale égale à deux fois la taille de la police, c'est-à-dire que l'interligne (espace entre deux lignes) sera égal à la hauteur de la police.



The screenshot shows a Notepad++ window with two tabs: "miseformetextes-19.html" and "miseformetextes-19.css".

miseformetextes-19.html:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".\\css\\miseformetextes-19.css">
7
8   </head>
9
10  <body>
11    <h1>Un titre de niveau 1</h1>
12
13    <p class="p1">Un premier paragraphe <br>sur deux lignes</p>
14    <p class="p2">Un autre paragraphe <br>sur deux lignes</p>
15    <p class="p3">Un troisième paragraphe <br>sur deux lignes</p>
16    <p class="p4">ppppqqqqgggjjj<br>dddffffhhhhjjkkkkllllbbb</p>
17
18  </body>
19
20 </html>

```

miseformetextes-19.css:

```

1 p{
2   font-size: 1em;
3 }
4
5 /*Hauteur totale de ligne (texte + interligne)
6 *= 2 fois la hauteur du texte*/
7 .p1{
8   line-height: 2em;
9 }
10
11 /*Hauteur totale de
12 *ligne (texte + interligne) = 3 fois
13 *la hauteur du texte*/
14 .p2{
15   line-height: 300%;
16 }
17
18 /*La hauteur totale de la ligne
19 *est égale à la hauteur du
20 *texte. La hauteur de
21 *l'interligne est donc nulle*/
22 .p3, .p4{
23   line-height: 1em;
24 }

```

HTML preview - HTML-CSS:

Un titre de niveau 1

Un premier paragraphe sur deux lignes

Un autre paragraphe sur deux lignes

Un troisième paragraphe sur deux lignes

ppppqqqqgggjjj
dddffffhhhhjjkkkkllllbbb

Ici, je donne un **line-height** égal à la taille de mon texte pour mon paragraphe p3. Chaque ligne va donc avoir exactement la même hauteur que le texte et il n'y aura pas d'espace entre les lignes.

Si on regarde attentivement, cependant, on observe qu'il reste tout de même un espace. Cela est dû à la définition même de la taille de la police. En effet, lorsqu'on définit une taille de police, on définit en fait la hauteur maximale que peut avoir un caractère.

Cependant, vous pouvez observer que tous les caractères ne font pas la même hauteur. Certains, comme le « e », le « a » ou le « r » occupent moins d'espace que le « l », le « p » ou que des caractères en majuscules par exemple. Notez que la hauteur réelle de chaque caractère va dépendre de la police utilisée.

C'est la raison pour laquelle il reste un espace visible entre les lignes lorsque les caractères ne sont pas des caractères occupant la hauteur maximale de police définie. Pour vous convaincre de cela, vous pouvez regarder le paragraphe p4 pour lequel nous avons défini la même taille de police et la même hauteur de ligne que pour p3 mais qui ne contient que des caractères occupant la taille maximale de police définie : il n'y a plus aucun espace visible entre les lignes dans ce cas.

La propriété letter-spacing

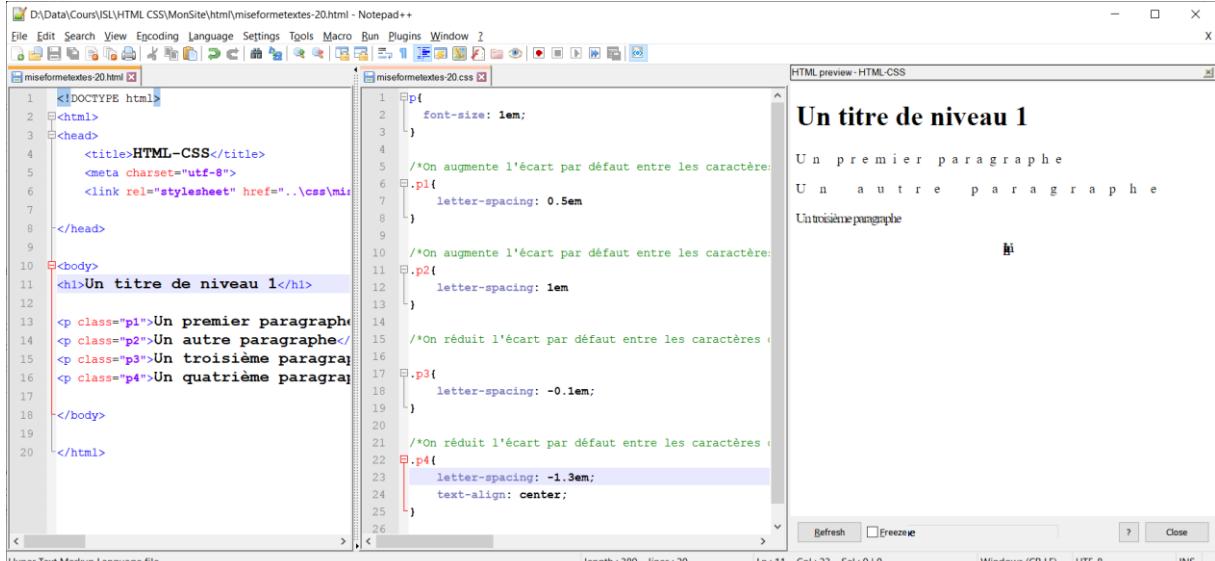
La propriété **letter-spacing** va nous permettre de définir l'espace entre les différents caractères de notre police.

Nous allons passer à cette propriété une valeur de type longueur en px ou en em par exemple. Les valeurs passées vont pouvoir être positives ou négatives.

La valeur passée à letter-spacing va venir s'ajouter à l'espace par défaut entre les caractères. Une valeur positive va donc augmenter l'espace entre les caractères tandis qu'une valeur négative va le réduire.

Attention toutefois : cette police est dépendante de la police utilisée et certaines valeurs passées vont possiblement ne pas être acceptées par la police et ainsi être modifiées automatiquement, notamment dans le cas de valeurs réduisant l'espace entre les caractères.

Notez également qu'en passant une valeur négative plus importante que la taille de votre police le texte s'affichera « à l'envers » c'est-à-dire que les lettres suivantes vont se retrouver avant les lettres précédentes.



The screenshot shows a Notepad++ interface with three tabs: "miseformetextes-20.html", "miseformetextes-20.css", and "HTML preview - HTML-CSS".

```

miseformetextes-20.html
<!DOCTYPE html>
<html>
<head>
<title>HTML-CSS</title>
<meta charset="utf-8">
<link rel="stylesheet" href="..\\css\\miseformetextes-20.css" type="text/css">
</head>
<body>
<h1>Un titre de niveau 1</h1>
<p class="p1">Un premier paragraphe</p>
<p class="p2">Un autre paragraphe</p>
<p class="p3">Un troisième paragraphe</p>
<p class="p4">Un quatrième paragraphe</p>
</body>
</html>

```

```

miseformetextes-20.css
p {
    font-size: 1em;
}
/*On augmente l'écart par défaut entre les caractères: */
.p1{
    letter-spacing: 0.5em;
}
/*On augmente l'écart par défaut entre les caractères: */
.p2{
    letter-spacing: 1em;
}
/*On réduit l'écart par défaut entre les caractères: */
.p3{
    letter-spacing: -0.1em;
}
/*On réduit l'écart par défaut entre les caractères: */
.p4{
    letter-spacing: -1.3em;
    text-align: center;
}

HTML preview - HTML-CSS
Un titre de niveau 1
Un premier paragraphe
Un autre paragraphe
Un troisième paragraphe
Un quatrième paragraphe

```

The "HTML preview" window shows the rendered HTML with applied letter-spacing styles.

miseformetextes-20.html

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

Un troisième paragraphe

Un quatrième paragraphe

eh paragraphe emè ir t auq nU

La propriété word-spacing

La propriété CSS **word-spacing** va fonctionner de manière similaire à letter-spacing mais va cette fois-ci nous permettre de définir l'espace entre les différents mots d'un texte.

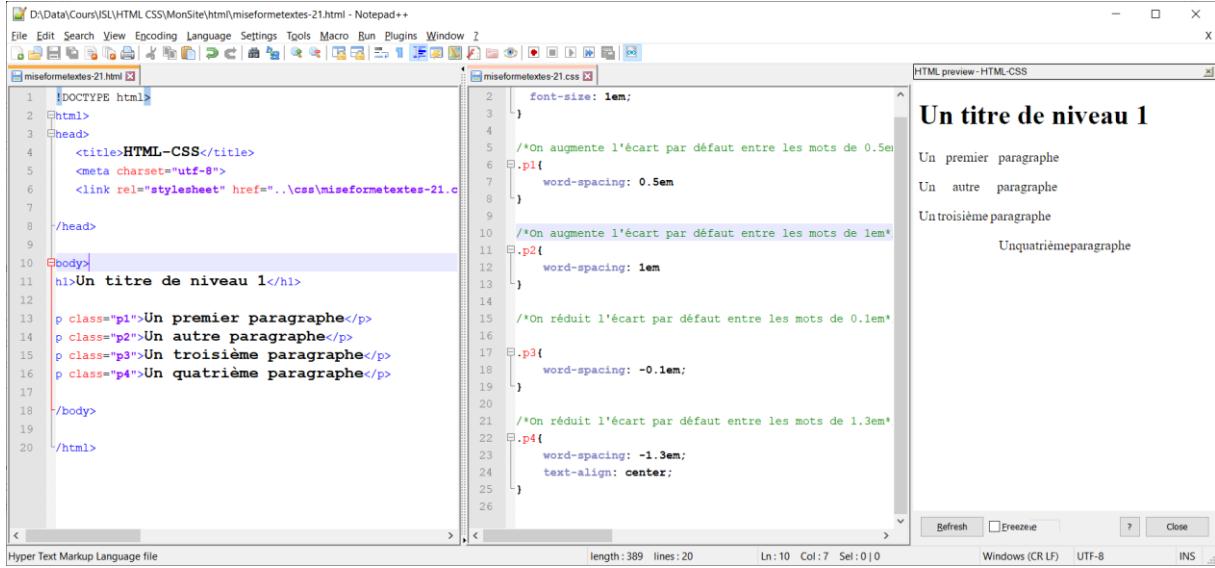
Une nouvelle fois, passer une valeur positive à word-spacing va augmenter l'espace défini par défaut par la police entre deux mots tandis que passer une valeur négative va le réduire.

De manière analogue à la propriété letter-spacing, il faudra faire très attention avec l'utilisation de valeurs négatives avec word-spacing car le texte peut très vite devenir illisible si plusieurs mots commencent à se chevaucher.

INSTITUT SAINT-LAURENT

ENSEIGNEMENT DE PROMOTION SOCIALE

Baccalauréat en informatique



The screenshot shows a Notepad++ interface with three windows:

- HTML file (miseformetextes-21.html):**

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".\\css\\miseformetextes-21.css">
7
8   </head>
9
10  <body>
11    <h1>Un titre de niveau 1</h1>
12
13    <p class="p1">Un premier paragraphe</p>
14    <p class="p2">Un autre paragraphe</p>
15    <p class="p3">Un troisième paragraphe</p>
16    <p class="p4">Un quatrième paragraphe</p>
17
18  </body>
19
20 </html>
```
- CSS file (miseformetextes-21.css):**

```
2   font-size: 1em;
3
4
5   /*On augmente l'écart par défaut entre les mots de 0.5em*/
6   .p1{
7     word-spacing: 0.5em
8   }
9
10  /*On augmente l'écart par défaut entre les mots de 1em*/
11  .p2{
12    word-spacing: 1em
13  }
14
15  /*On réduit l'écart par défaut entre les mots de 0.1em*/
16  .p3{
17    word-spacing: -0.1em;
18  }
19
20  /*On réduit l'écart par défaut entre les mots de 1.3em*/
21  .p4{
22    word-spacing: -1.3em;
23    text-align: center;
24  }
```
- Preview window (HTML preview - HTML-CSS):**

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

Un troisième paragraphe

Un quatrième paragraphe

Couleur des textes HTML avec CSS color et opacité

Dans cette nouvelle leçon, nous allons apprendre à modifier la couleur et l'opacité de nos textes grâce à la propriété **color** que nous avons déjà rencontré précédemment.

Cette propriété est à la fois très simple à utiliser et relativement complexe à parfaitement maîtriser car nous allons lui passer des valeurs de couleurs très différentes les unes des autres.

La propriété **color** va en effet pouvoir accepter des valeurs comme :

- Un **nom de couleur** (en anglais) ;
- Une **notation hexadécimale** ;
- Une notation **RGB** ou **RGBA** ;
- Une notation **HSL** ou **HSLa**.

Toutes ces notations vont nous permettre, in fine, d'attribuer une couleur particulière à notre texte. Le but de cette leçon est de comprendre comment chaque type de valeur fonctionne et les avantages et inconvénients de chacun.

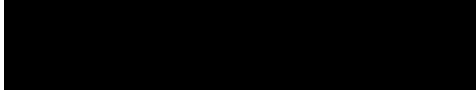
Les valeurs de type « nom de couleur »

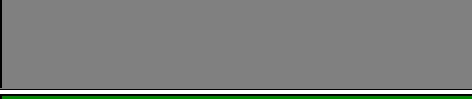
Les 16 premières couleurs normalisées

Il y a des années de cela, les langages de programmation ne disposaient pas de toutes les fonctionnalités d'aujourd'hui tout simplement car les infrastructures étaient beaucoup moins puissantes que de nos jours.

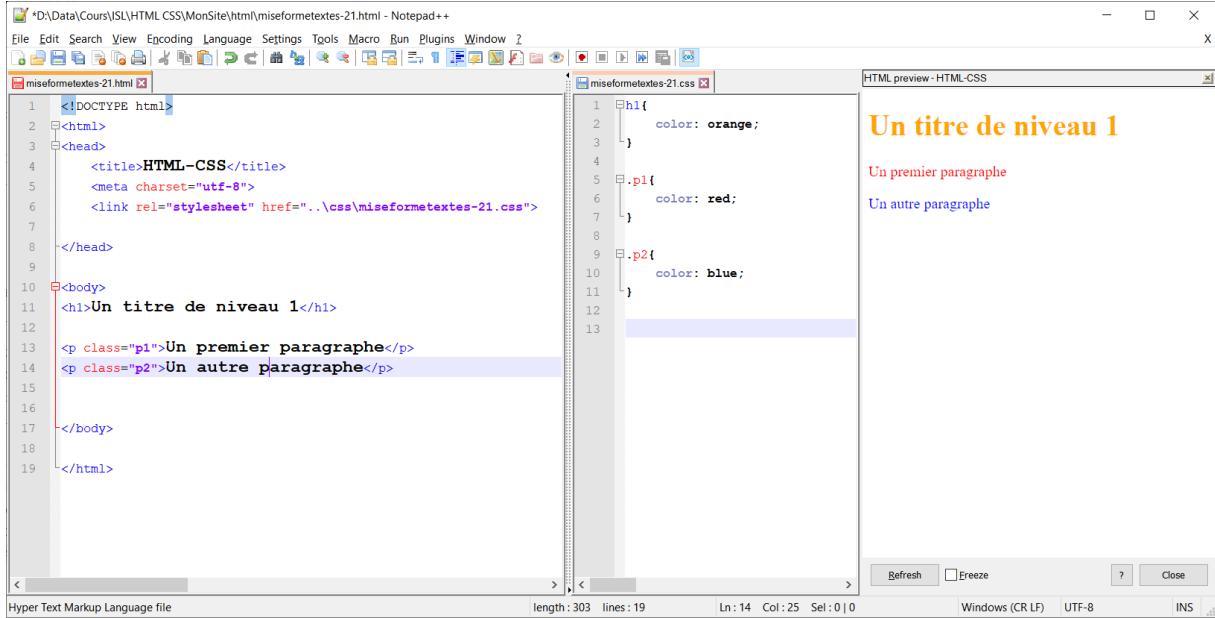
Ainsi, au départ, seules 16 couleurs ont été normalisées en CSS. C'était déjà un grand pas en avant pour les utilisateurs : ils n'avaient plus qu'à passer le nom (en anglais) de la couleur normalisée en valeur de la propriété CSS **color** afin de changer la couleur d'un élément.

Ces seize couleurs CSS sont les suivantes. Notez que j'ai déjà renseigné l'équivalent de chaque nom de couleur en **notation hexadécimale** dans le tableau ci-dessous. Nous reparlerons de ces notations plus tard dans cette leçon.

Nom de la couleur	Hexadecimal	Couleur
Aqua	#00FFFF	
Black	#000000	

Blue	#0000FF	
Fuschia	#FF00FF	
Gray	#808080	
Green	#008800	
Lime	#00FF00	
Maroon	#800000	
Navy	#000080	
Olive	#808000	
Purple	#800080	
Red	#FF0000	
Silver	#COCOCO	
Teal	#008080	
White	#FFFFFF	
Yellow	#00FFFF	

Voyons immédiatement en pratique comment utiliser ces noms de couleurs en CSS avec différents exemples utilisant color en CSS :



The screenshot shows a Notepad++ interface with two tabs: "miseformetextes-21.html" and "miseformetextes-21.css". The HTML tab contains the following code:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="..\css\miseformetextes-21.css">
7
8   </head>
9
10  <body>
11    <h1>Un titre de niveau 1</h1>
12
13    <p class="p1">Un premier paragraphe</p>
14    <p class="p2">Un autre paragraphe</p>
15
16
17  </body>
18
19 </html>

```

The CSS tab contains the following code:

```

1 h1 {
2   color: orange;
3 }
4
5 .p1{
6   color: red;
7 }
8
9 .p2{
10   color: blue;
11 }
12
13

```

The "HTML preview" window on the right shows the rendered output:

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

Dans l'exemple ci-dessus, vous pouvez voir que nous attribuons en CSS une couleur orange à notre titre de niveau 1 avec le code `h1{color : orange}`. Nous définissons également des couleurs rouge (red) et bleu (blue) pour les textes de nos éléments p1 et p2.

Les autres couleurs nommées

Avec l'évolution des performances et des langages, le support pour de nouvelles couleurs a progressivement été ajouté.

Ainsi, aujourd'hui, les navigateurs reconnaissent et supportent l'utilisation de plus de 140 noms de couleurs différents. Nous allons donc pouvoir passer chacune de ces valeurs à la propriété color pour définir une nouvelle couleur pour un texte.

Voici la liste de ces couleurs CSS (noms en anglais) ainsi que leur code hexadécimal :

Nom de la couleur	Hexadecimal	Couleur
AliceBlue	#F0F8FF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFDD	

Azure	#FOFFFF	
Beige	#F5F5DC	
Bisque	#FFE4C4	
Black	#000000	
BlanchedAlmond	#FFEBCD	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	
BurlyWood	#DEB887	
CadetBlue	#5F9EA0	
Chartreuse	#7FFF00	
Chocolate	#D2691E	
Coral	#FF7F50	
CornflowerBlue	#6495ED	
Cornsilk	#FFF8DC	
Crimson	#DC143C	
Cyan	#00FFFF	

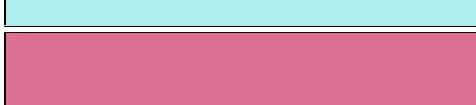
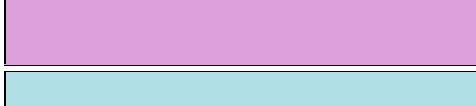
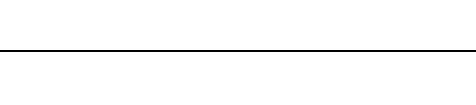
DarkBlue	#00008B	<div style="background-color: #00008B; width: 100px; height: 20px;"></div>
DarkCyan	#008B8B	<div style="background-color: #008B8B; width: 100px; height: 20px;"></div>
DarkGoldenRod	#B8860B	<div style="background-color: #B8860B; width: 100px; height: 20px;"></div>
DarkGray	#A9A9A9	<div style="background-color: #A9A9A9; width: 100px; height: 20px;"></div>
DarkGrey	#A9A9A9	<div style="background-color: #A9A9A9; width: 100px; height: 20px;"></div>
DarkGreen	#006400	<div style="background-color: #006400; width: 100px; height: 20px;"></div>
DarkKhaki	#BDB76B	<div style="background-color: #BDB76B; width: 100px; height: 20px;"></div>
DarkMagenta	#8B008B	<div style="background-color: #8B008B; width: 100px; height: 20px;"></div>
DarkOliveGreen	#556B2F	<div style="background-color: #556B2F; width: 100px; height: 20px;"></div>
DarkOrange	#FF8C00	<div style="background-color: #FF8C00; width: 100px; height: 20px;"></div>
DarkOrchid	#9932CC	<div style="background-color: #9932CC; width: 100px; height: 20px;"></div>
DarkRed	#8B0000	<div style="background-color: #8B0000; width: 100px; height: 20px;"></div>
DarkSalmon	#E9967A	<div style="background-color: #E9967A; width: 100px; height: 20px;"></div>
DarkSeaGreen	#8FBC8F	<div style="background-color: #8FBC8F; width: 100px; height: 20px;"></div>
DarkSlateBlue	#483D8B	<div style="background-color: #483D8B; width: 100px; height: 20px;"></div>
DarkSlateGray	#2F4F4F	<div style="background-color: #2F4F4F; width: 100px; height: 20px;"></div>
DarkSlateGrey	#2F4F4F	<div style="background-color: #2F4F4F; width: 100px; height: 20px;"></div>

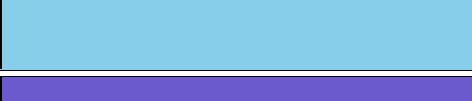
DarkTurquoise	#00CED1	<div style="background-color: #00CED1; width: 100px; height: 20px;"></div>
DarkViolet	#9400D3	<div style="background-color: #9400D3; width: 100px; height: 20px;"></div>
DeepPink	#FF1493	<div style="background-color: #FF1493; width: 100px; height: 20px;"></div>
DeepSkyBlue	#00BFFF	<div style="background-color: #00BFFF; width: 100px; height: 20px;"></div>
DimGray	#696969	<div style="background-color: #696969; width: 100px; height: 20px;"></div>
DimGrey	#696969	<div style="background-color: #696969; width: 100px; height: 20px;"></div>
DodgerBlue	#1E90FF	<div style="background-color: #1E90FF; width: 100px; height: 20px;"></div>
FireBrick	#B22222	<div style="background-color: #B22222; width: 100px; height: 20px;"></div>
FloralWhite	#FFFFA0	<div style="background-color: #FFFFA0; width: 100px; height: 20px;"></div>
ForestGreen	#228B22	<div style="background-color: #228B22; width: 100px; height: 20px;"></div>
Fuchsia	#FF00FF	<div style="background-color: #FF00FF; width: 100px; height: 20px;"></div>
Gainsboro	#DCDCDC	<div style="background-color: #DCDCDC; width: 100px; height: 20px;"></div>
GhostWhite	#F8F8FF	<div style="background-color: #F8F8FF; width: 100px; height: 20px;"></div>
Gold	#FFD700	<div style="background-color: #FFD700; width: 100px; height: 20px;"></div>
GoldenRod	#DAA520	<div style="background-color: #DAA520; width: 100px; height: 20px;"></div>
Gray	#808080	<div style="background-color: #808080; width: 100px; height: 20px;"></div>
Grey	#808080	<div style="background-color: #808080; width: 100px; height: 20px;"></div>

Green	#008000	<div style="background-color: #008000; width: 100px; height: 20px;"></div>
GreenYellow	#ADFF2F	<div style="background-color: #ADFF2F; width: 100px; height: 20px;"></div>
HoneyDew	#FOFFF0	<div style="background-color: #FOFFF0; width: 100px; height: 20px;"></div>
HotPink	#FF69B4	<div style="background-color: #FF69B4; width: 100px; height: 20px;"></div>
IndianRed	#CD5C5C	<div style="background-color: #CD5C5C; width: 100px; height: 20px;"></div>
Indigo	#4B0082	<div style="background-color: #4B0082; width: 100px; height: 20px;"></div>
Ivory	#FFFFFF	<div style="background-color: #FFFFFF; width: 100px; height: 20px;"></div>
Khaki	#F0E68C	<div style="background-color: #F0E68C; width: 100px; height: 20px;"></div>
Lavender	#E6E6FA	<div style="background-color: #E6E6FA; width: 100px; height: 20px;"></div>
LavenderBlush	#FFF0F5	<div style="background-color: #FFF0F5; width: 100px; height: 20px;"></div>
LawnGreen	#7CFC00	<div style="background-color: #7CFC00; width: 100px; height: 20px;"></div>
LemonChiffon	#FFFACD	<div style="background-color: #FFFACD; width: 100px; height: 20px;"></div>
LightBlue	#ADD8E6	<div style="background-color: #ADD8E6; width: 100px; height: 20px;"></div>
LightCoral	#F08080	<div style="background-color: #F08080; width: 100px; height: 20px;"></div>
LightCyan	#E0FFFF	<div style="background-color: #E0FFFF; width: 100px; height: 20px;"></div>
LightGoldenRodYellow	#FAFAD2	<div style="background-color: #FAFAD2; width: 100px; height: 20px;"></div>
LightGray	#D3D3D3	<div style="background-color: #D3D3D3; width: 100px; height: 20px;"></div>

LightGrey	#D3D3D3	<div style="background-color: #D3D3D3; width: 100px; height: 20px;"></div>
LightGreen	#90EE90	<div style="background-color: #90EE90; width: 100px; height: 20px;"></div>
LightPink	#FFB6C1	<div style="background-color: #FFB6C1; width: 100px; height: 20px;"></div>
LightSalmon	#FFA07A	<div style="background-color: #FFA07A; width: 100px; height: 20px;"></div>
LightSeaGreen	#20B2AA	<div style="background-color: #20B2AA; width: 100px; height: 20px;"></div>
LightSkyBlue	#87CEFA	<div style="background-color: #87CEFA; width: 100px; height: 20px;"></div>
LightSlateGray	#778899	<div style="background-color: #778899; width: 100px; height: 20px;"></div>
LightSlateGrey	#778899	<div style="background-color: #778899; width: 100px; height: 20px;"></div>
LightSteelBlue	#B0C4DE	<div style="background-color: #B0C4DE; width: 100px; height: 20px;"></div>
LightYellow	#FFFFE0	<div style="background-color: #FFFFE0; width: 100px; height: 20px;"></div>
Lime	#00FF00	<div style="background-color: #00FF00; width: 100px; height: 20px;"></div>
LimeGreen	#32CD32	<div style="background-color: #32CD32; width: 100px; height: 20px;"></div>
Linen	#FAF0E6	<div style="background-color: #FAF0E6; width: 100px; height: 20px;"></div>
Magenta	#FF00FF	<div style="background-color: #FF00FF; width: 100px; height: 20px;"></div>
Maroon	#800000	<div style="background-color: #800000; width: 100px; height: 20px;"></div>
MediumAquaMarine	#66CDAA	<div style="background-color: #66CDAA; width: 100px; height: 20px;"></div>
MediumBlue	#0000CD	<div style="background-color: #0000CD; width: 100px; height: 20px;"></div>

MediumOrchid	#BA55D3	<div style="background-color: #BA55D3; width: 100px; height: 20px;"></div>
MediumPurple	#9370DB	<div style="background-color: #9370DB; width: 100px; height: 20px;"></div>
MediumSeaGreen	#3CB371	<div style="background-color: #3CB371; width: 100px; height: 20px;"></div>
MediumSlateBlue	#7B68EE>	<div style="background-color: #7B68EE; width: 100px; height: 20px;"></div>
MediumSpringGreen	#00FA9A	<div style="background-color: #00FA9A; width: 100px; height: 20px;"></div>
MediumTurquoise	#48D1CC<	<div style="background-color: #48D1CC; width: 100px; height: 20px;"></div>
MediumVioletRed	#C71585	<div style="background-color: #C71585; width: 100px; height: 20px;"></div>
MidnightBlue	#191970	<div style="background-color: #191970; width: 100px; height: 20px;"></div>
MintCream	#F5FFFA	<div style="background-color: #F5FFFA; width: 100px; height: 20px;"></div>
MistyRose	#FFE4E1	<div style="background-color: #FFE4E1; width: 100px; height: 20px;"></div>
Moccasin	#FFE4B5	<div style="background-color: #FFE4B5; width: 100px; height: 20px;"></div>
NavajoWhite	#FFDEAD	<div style="background-color: #FFDEAD; width: 100px; height: 20px;"></div>
Navy	#000080	<div style="background-color: #000080; width: 100px; height: 20px;"></div>
OldLace	#FDF5E6	<div style="background-color: #FDF5E6; width: 100px; height: 20px;"></div>
Olive	#808000	<div style="background-color: #808000; width: 100px; height: 20px;"></div>
OliveDrab	#6B8E23	<div style="background-color: #6B8E23; width: 100px; height: 20px;"></div>
Orange	#FFA500	<div style="background-color: #FFA500; width: 100px; height: 20px;"></div>

OrangeRed	#FF4500	
Orchid	#DA70D6	
PaleGoldenRod	#EEE8AA<	
PaleGreen	#98FB98	
PaleTurquoise	#AFEEEE	
PaleVioletRed	#DB7093	
PapayaWhip	#FFEFD5	
PeachPuff	#FFDAB9	
Peru	#CD853F	
Pink	#FFC0CB	
Plum	#DDA0DD	
PowderBlue	#B0E0E6	
Purple	#800080	
RebeccaPurple	#663399	
Red	#FF0000	
RosyBrown	#BC8F8F	
RoyalBlue	#4169E1	

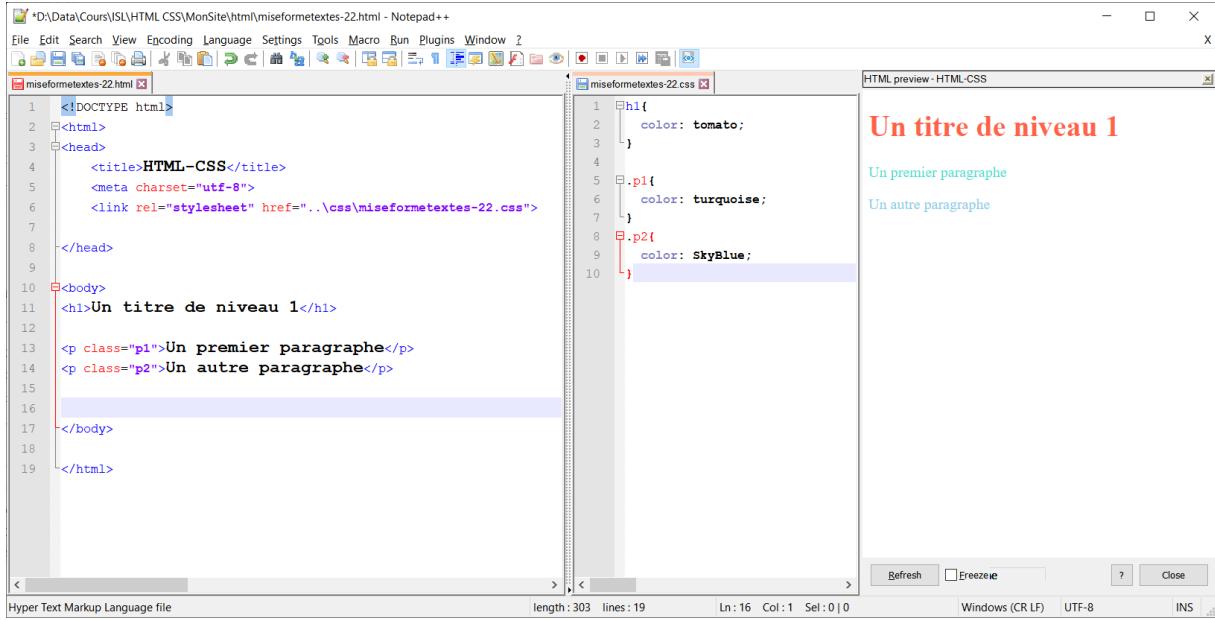
SaddleBrown	#8B4513	
Salmon	#FA8072	
SandyBrown	#F4A460	
SeaGreen	#2E8B57	
SeaShell	#FFF5EE	
Sienna	#A0522D	
Silver	#COCOCO	
SkyBlue	#87CEEB	
SlateBlue	#6A5ACD	
SlateGray	#708090	
SlateGrey	#708090	
Snow	#FFFFFA	
SpringGreen	#00FF7F	
SteelBlue	#4682B4	
Tan	#D2B48C	
Teal	#008080	
Thistle	#D8BFD8	

Tomato	#FF6347	<div style="background-color: #FF6347; width: 100px; height: 20px;"></div>
Turquoise	#40E0D0	<div style="background-color: #40E0D0; width: 100px; height: 20px;"></div>
Violet	#EE82EE	<div style="background-color: #EE82EE; width: 100px; height: 20px;"></div>
Wheat	#F5DEB3	<div style="background-color: #F5DEB3; width: 100px; height: 20px;"></div>
White	#FFFFFF	<div style="background-color: #FFFFFF; width: 100px; height: 20px;"></div>
WhiteSmoke	#F5F5F5	<div style="background-color: #F5F5F5; width: 100px; height: 20px;"></div>
Yellow	#FFFF00	<div style="background-color: #FFFF00; width: 100px; height: 20px;"></div>
YellowGreen	#9ACD32	<div style="background-color: #9ACD32; width: 100px; height: 20px;"></div>

Fait intéressant ici, vous pouvez noter que :

- Fuchsia possède le même code couleur que Magenta ;
- Aqua possède le même code couleur que Cyan.

Utilisons immédiatement ces noms de couleurs CSS avec la propriété color à travers de nouveaux exemples :



```

*D:\Data\Cours\ISL\HTML CSS\MonSite\html\miseformetextes-22.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
miseformetextes-22.html miseformetextes-22.css HTML preview - HTML-CSS
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>HTML-CSS</title>
5      <meta charset="utf-8">
6      <link rel="stylesheet" href=".\\css\\miseformetextes-22.css">
7
8  </head>
9
10 <body>
11     <h1>Un titre de niveau 1</h1>
12
13     <p class="p1">Un premier paragraphe</p>
14     <p class="p2">Un autre paragraphe</p>
15
16
17 </body>
18
19 </html>

```

```

1  h1{
2      color: tomato;
3  }
4
5  .p1{
6      color: turquoise;
7  }
8  .p2{
9      color: SkyBlue;
10 }

```

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

Avantages et limitations des valeurs de type « nom de couleur »

L'utilisation des valeurs de type « nom de couleur » avec la propriété CSS color est très pratique puisqu'il suffit d'indiquer le nom de la couleur souhaitée.

Cependant, ce type de valeurs possède une limitation majeure : **nous sommes limités en termes de couleurs à ces quelques 140 noms**. Or, parfois, nous voudrons utiliser une couleur ou une variation de couleur différente de ces 140 disponibles pour définir une identité visuelle précise.

Dans ces cas-là, nous utiliserons alors plutôt l'un des autres types de notation, que ce soit des notations RGB, HEX, ou HSL.

En effet, chacun de ces nouveaux types de valeurs va nous permettre de créer et d'utiliser jusqu'à **16 millions** de variations de couleurs afin de trouver la couleur exacte voulue en CSS. Ils vont tous reposer sur une logique similaire de mélange des couleurs rouge, vert et bleu.

Les notations de type RGB

Commençons déjà par expliquer comment fonctionnent les valeurs de type RGB et comment les utiliser avec color en CSS.

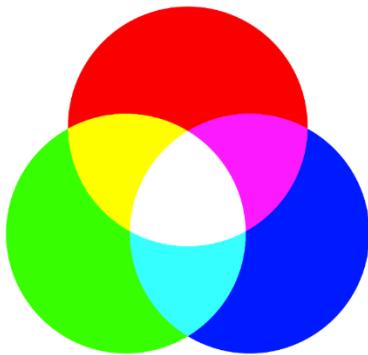
Avant tout, vous devez savoir que les lettres « **RGB** » sont les abréviations de « **Red Green Blue** » soit « Rouge Vert Bleu » en français. Effectivement, pour créer une couleur RGB, nous allons devoir préciser trois niveaux d'intensité de Rouge, de Vert, et de Bleu qui vont ensuite être mélangés pour créer la couleur finale.

Chaque niveau d'intensité qu'on va renseigner va être compris entre 0 (intensité nulle ou absence de la couleur en question) et 255 (intensité maximale ou couleur pure). En précisant une intensité de rouge de 0, par exemple, on signifie qu'on ne souhaite pas utiliser de rouge dans notre couleur finale.

En précisant une intensité de 255 de rouge, en revanche, on indique qu'on souhaite utiliser beaucoup de rouge pour créer notre couleur finale.

Vous vous rappelez à l'école quand vous mélangez différentes couleurs en peinture entre elles pour en créer une nouvelle ? Cela fonctionne également de la même manière !

Pour rappel, je vous donne une image avec les couleurs obtenues lorsqu'on mélange en quantité équivalente nos trois couleurs de base. Cela vous aidera pour comprendre la suite.



Illustrons immédiatement avec quelques exemples le fonctionnement des notations RGB en CSS.

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href=".css/miseformetextes-23.css">
7
8   </head>
9
10  <body>
11
12    <h1>Les notations RGB en CSS</h1>
13
14    <p>On va pouvoir <strong>créer une couleur RGB en CSS</strong> en mixant différentes intensités de Rouge, de Vert, et de Bleu.</p>
15
16    <p><em>L'intensité</em> de chacune de ces couleurs de base doit être comprise entre <em>0 (intensité minimale)</em> et <em>255 (intensité maximale)</em>.</p>
17
18
19
20
21
22
23
24
25
26

```

```

1 h1{
2   color : RGB(255, 180, 0);
3 }
4
5 p{
6   color : RGB(50, 50, 50);
7 }
8
9 strong{
10   color : RGB(0, 0, 200);
11 }
12
13 em{
14   color : RGB(0, 200, 50);
15 }

```

Les notations RGB en CSS

On va pouvoir **créer une couleur RGB en CSS** en mixant différentes intensités de Rouge, de Vert, et de Bleu.

L'intensité de chacune de ces couleurs de base doit être comprise entre **0 (intensité minimale)** et **255 (intensité maximale)**.

Regardons ces exemples de plus près, en commençant déjà par nous intéresser à la syntaxe des notations RGB en CSS. Comme vous pouvez le voir, lorsqu'on renseigne une valeur de type RGB en valeur de la propriété CSS color, il faut le déclarer en écrivant RGB().

Ensuite, vous pouvez observer dans chaque cas 3 chiffres séparés par des virgules à l'intérieur des parenthèses. Ces trois chiffres correspondent respectivement aux niveaux d'intensité de Rouge, de Vert et de Bleu qui seront utilisés pour créer notre couleur finale.

A ce niveau, vous pouvez noter que plus un niveau d'intensité va de rapprocher de 0, plus cela va correspondre à une couleur foncée. A l'inverse, plus un niveau d'intensité va se rapprocher de 255, plus la couleur va être claire. Pour retenir cela, retenez que **color : RGB(0, 0, 0)** correspond à du noir tandis que **color : RGB(255, 255, 255)** correspond à du blanc en CSS.

Intéressons-nous maintenant de plus près aux différents cas ci-dessus. On commence avec le code CSS `h1{color : RGB(255, 180, 0);}`. Vous remarquez que ce code me donne une sorte de orange en résultat. Essayons de déterminer pourquoi. Pour créer ce orange, j'ai précisé une intensité maximale de rouge (255) que j'ai mélangé avec une intensité forte de vert (180) et une intensité nulle de bleu (0).

Vous devriez normalement savoir que lorsqu'on mélange du rouge et du vert en quantité égale, on obtient du jaune. Ici, nous avons mélangé le maximum de rouge avec beaucoup de vert (mais en mettant moins de vert que de rouge). Notre couleur finale va donc se trouver entre du jaune et du rouge... C'est-à-dire du orange !

Pour nos paragraphes, nous précisons la même intensité de rouge, de vert et de bleu et nos trois intensités sont relativement basses. Je vous ai dit plus haut que `RGB(0, 0, 0)` donnait du noir tandis que `RGB(255, 255, 255)` donnait du blanc. Nous allons donc ici obtenir une sorte de gris foncé.

Essayez de comprendre par vous-même les couleurs obtenues pour les textes contenus dans nos éléments `strong` et `em`, ça vous fera un bon exercice.

Bon à savoir : Vous vous rappelez lorsque je vous ai dit que le type de valeurs RGB permettait de créer plus de 16 millions de couleurs en CSS ? Comprenez-vous maintenant d'où vient ce chiffre ? Explication : nous pouvons choisir parmi 256 niveaux d'intensité de rouge, vert et bleu pour créer notre couleur finale. Oui, j'ai bien dit 256 et pas 255 car le 0 compte comme un niveau d'intensité : l'intensité nulle. Ainsi, on va pouvoir créer $256 * 256 * 256 = 16\ 777\ 216$ couleurs différentes en utilisant les notations RGB avec `color` en CSS ! Les valeurs hexadécimales vont fonctionner sur le même principe.

Les valeurs de type hexadécimales

Les valeurs de type hexadécimal vont reposer sur le même principe que les valeurs de type RGB : nous allons à nouveau pouvoir préciser trois niveaux d'intensité de rouge, de vert et de bleu pour créer une couleur personnalisée.

La seule différence entre ces deux types de notation va être la façon dont on va compter : le mot « hexadécimal » signifie « qui fonctionne en base 16 ». Cela veut dire que le système hexadécimal utilise **16** symboles différents pour compter et représenter les chiffres.

Dans la vie de tous les jours, nous utilisons le système décimal : nous comptons en base 10. Cela signifie que nous utilisons 10 symboles pour compter : ce sont le 0, le 1, le 2, 3, 4, 5, 6, 7, 8, et le 9.

Le système hexadécimal, comme je vous l'ai dit, va utiliser non pas 10 mais 16 symboles pour compter. Ces symboles vont être : **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E et F**. Comme vous pouvez le remarquer, nous comptons de la même manière en hexadécimal qu'en décimal jusqu'à 9 puis nous utilisons les lettres A B C D E F qui vont être l'équivalent des unités « 10 », « 11 », « 12 », « 13 », « 14 » et « 15 » en système décimal.

Comment représente-t-on le 16 en hexadécimal me direz-vous ? Nous allons utiliser le même principe qu'avec le système décimal, excepté que cette fois-ci nous utilisons des dizaines et non pas des dizaines. Ainsi, pour représenter le « 16 » en hexadécimal, nous utiliserons le symbole « 10 » (comprendre 1 dizaine + 0 unité).

Regardez plutôt le tableau ci-dessous pour mieux comprendre :

Décimal	Equivalent hexadécimal
0 (ou 00)	0 (ou 00)
1 (ou 01)	1 (ou 01)
2 (ou 02)	2 (ou 02)
9 (ou 09)	9 (ou 09)
10	A (ou 0A)
11	B (ou 0B)
16	10
17	11
25	19
26	1A
32	20

Au final, le système hexadécimal n'est qu'une façon différente de compter en utilisant 16 unités de base et non pas 10 comme on en a l'habitude. Cela peut désorienter au premier abord mais c'est en

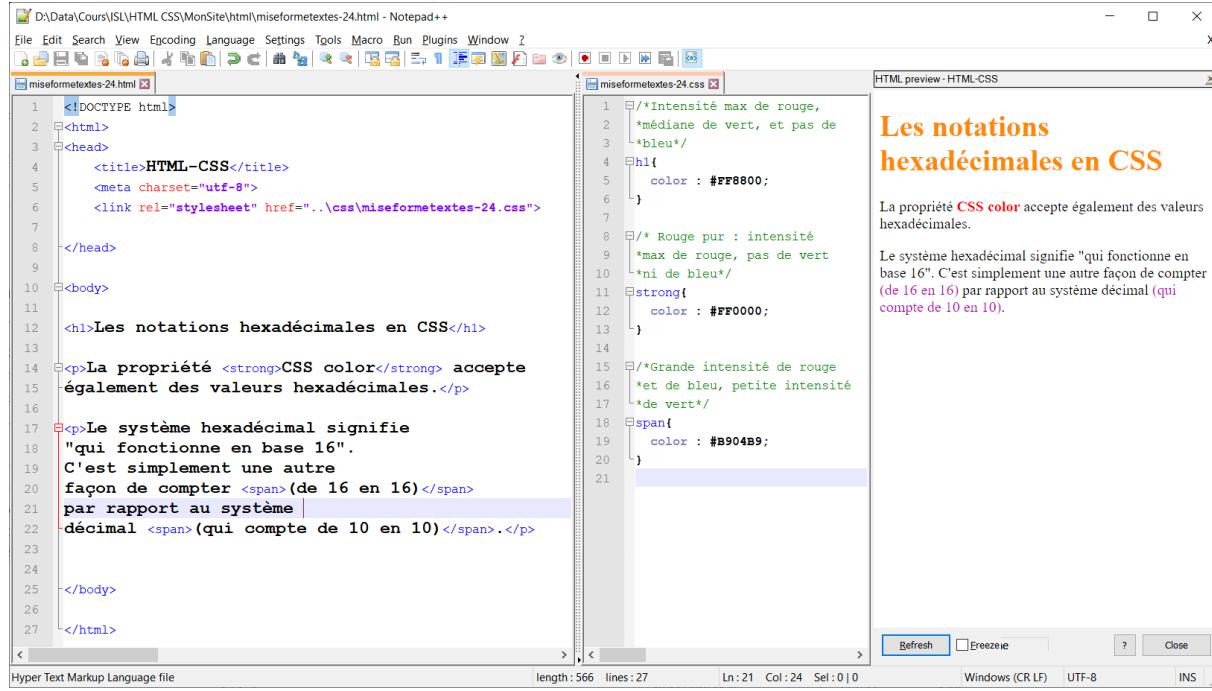
fait très simple. Faites l'effort de comprendre cela pour bien comprendre l'utilisation des valeurs hexadécimales avec les couleurs en CSS !

Nous allons en effet pouvoir utiliser les notations hexadécimales pour créer une couleur avec la propriété CSS color. Les valeurs hexadécimales sont même les plus utilisées en CSS pour créer des couleurs !

Nous allons ici utiliser exactement le même principe qu'avec les valeurs RGB en précisant trois intensités de Rouge, Vert et Bleu pour créer une couleur. Là encore, chacune des trois intensités des couleurs de base va pouvoir être comprise entre 0 et 255 ou plus exactement entre 00 et FF en notation hexadécimale (FF en hexadécimal est équivalent à 255 en décimal).

Pour créer une couleur en utilisant les notations hexadécimales en CSS, nous allons donc devoir préciser trois intensités de Rouge, Vert et de Bleu entre 00 et FF à la suite. De plus, nous devrons faire précéder cette notation par un dièse (symbole #).

Prenons immédiatement quelques exemples pour illustrer cela :



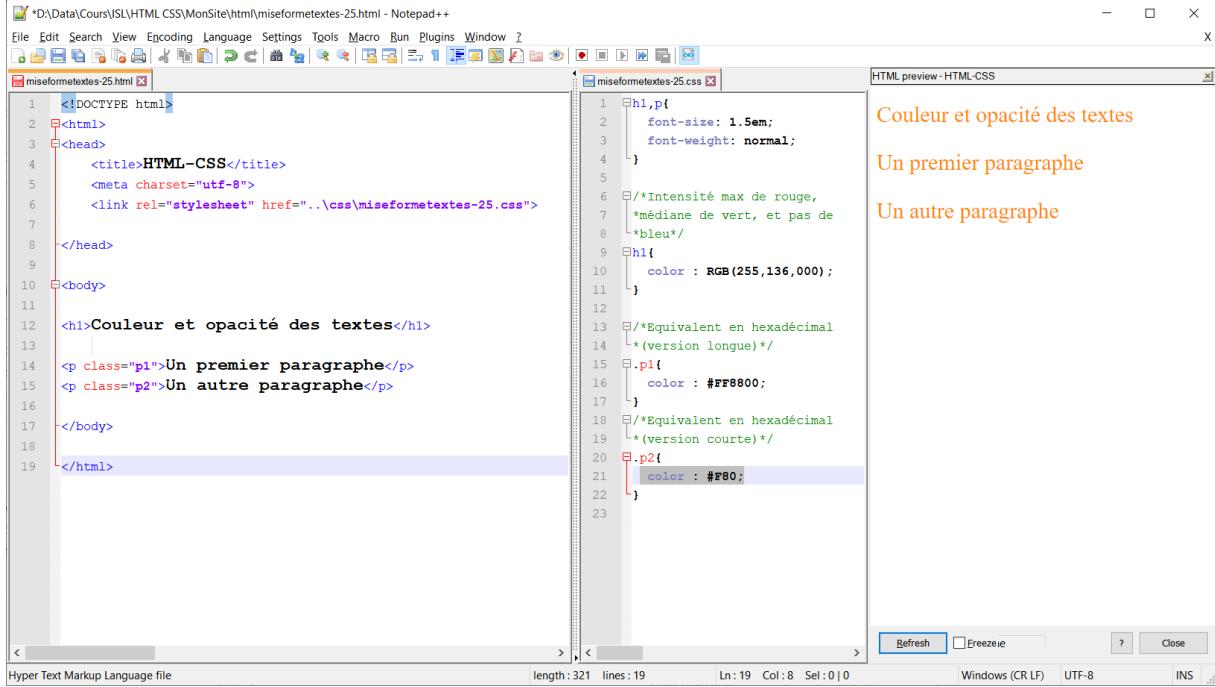
```

D:\Data\Cours\ISL\HTML CSS\MonSite\html\miseformetextes-24.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
miseformetextes-24.html [1]
miseformetextes-24.css [2]
HTML preview - HTML-CSS
Les notations hexadécimales en CSS
La propriété CSS color accepte également des valeurs hexadécimales.
Le système hexadécimal signifie "qui fonctionne en base 16". C'est simplement une autre façon de compter (de 16 en 16) par rapport au système décimal (qui compte de 10 en 10).
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="../css/miseformetextes-24.css">
7
8   </head>
9
10  <body>
11    <h1>Les notations hexadécimales en CSS</h1>
12
13    <p>La propriété CSS color accepte également des valeurs hexadécimales.</p>
14
15    <p>Le système hexadécimal signifie "qui fonctionne en base 16". C'est simplement une autre façon de compter (de 16 en 16) par rapport au système décimal (qui compte de 10 en 10).</p>
16
17
18
19
20
21
22
23
24
25
26
27
</body>
</html>

```

On applique une `color : #FF8800` à notre titre h1 en CSS. On utilise ici une intensité maximale de rouge (FF = 255 en décimal), moyenne de vert (88 = 136 en décimal) et minimale (00) de bleu pour créer notre couleur finale. Comme on mélange beaucoup de rouge avec un peu de vert, on obtient naturellement du orange.

Notez ici quelque chose d'intéressant : lorsqu'on souhaite utiliser deux fois la même unité pour préciser une intensité (par exemple « 88 », « BB » ou « EE », on peut utiliser une notation raccourcie en se contentant de ne préciser l'unité qu'une seule fois.



The screenshot shows a Notepad++ interface with two tabs open. The left tab contains an HTML file named 'miseformetextes-25.html' with the following content:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="..\\css\\miseformetextes-25.css">
7
8   </head>
9
10  <body>
11
12    <h1>Couleur et opacité des textes</h1>
13
14    <p class="p1">Un premier paragraphe</p>
15    <p class="p2">Un autre paragraphe</p>
16
17  </body>
18
19 </html>

```

The right tab contains a CSS file named 'miseformetextes-25.css' with the following content:

```

1 h1,p{
2   font-size: 1.5em;
3   font-weight: normal;
4 }
5
6 /*Intensité max de rouge,
7 *médiane de vert, et pas de
8 *bleu*/
9 h1{
10   color : RGB(255,136,000);
11 }
12
13 /*Équivalent en hexadécimal
14 *(version longue)*/
15 .p1{
16   color : #FF8800;
17 }
18 /*Équivalent en hexadécimal
19 *(version courte)*/
20 .p2{
21   color : #F80;
22 }

```

An 'HTML preview - HTML-CSS' window is also visible on the right, showing the rendered output of the HTML and CSS files.

Ici, vous pouvez remarquer qu'il est strictement équivalent d'écrire `color : FF8800` et `color : #F80`.

Notez par ailleurs qu'il est strictement équivalent d'utiliser des majuscules ou des minuscules pour les lettres des valeurs hexadécimales.

Les notations de type HSL

« HSL » est l'abréviation de « **Hue-Saturation-Lightning** », c'est-à-dire « teinte-saturation-luminosité » en français.

Pour créer une couleur en utilisant les notations HSL, nous allons devoir renseigner trois valeurs :

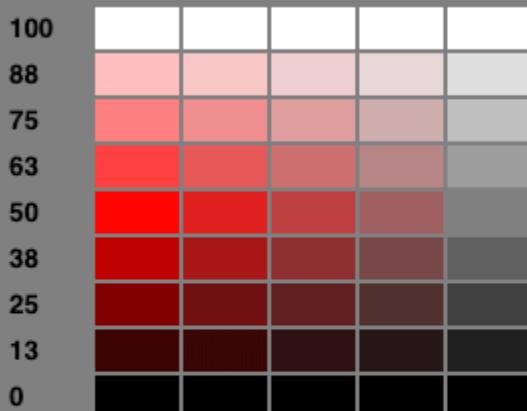
- La **teinte**. On va devoir ici renseigner un nombre entre 0 et 360 qui représente un angle du cercle chromatique (c'est-à-dire l'arc en ciel représenté dans un cercle). Ici, vous pouvez retenir que la couleur rouge correspond à un angle de 0 (degré) ou de 360 (degrés), tandis que le vert va se situer à 120 degrés et le bleu à 240 degrés. Nous allons pouvoir préciser des valeurs décimales ici pour choisir précisément une couleur ;
- La **saturation**. Celle-ci va être représentée sous forme de pourcentage. 100% correspond à une saturation maximale tandis que 0% correspond à une saturation minimale ;
- La **luminosité**. Celle-ci va également être représentée par un pourcentage. 100% de luminosité correspond à du blanc tandis que 0% correspond à du noir.

Vous pouvez vous aider des repères suivants fournis par le W3C pour bien comprendre comment fonctionnent les couleurs HSL. Ici, le pourcentage de saturation est indiqué sur l'axe horizontal tandis que le pourcentage de luminosité est indiqué sur l'axe vertical :

0° Reds

Saturation

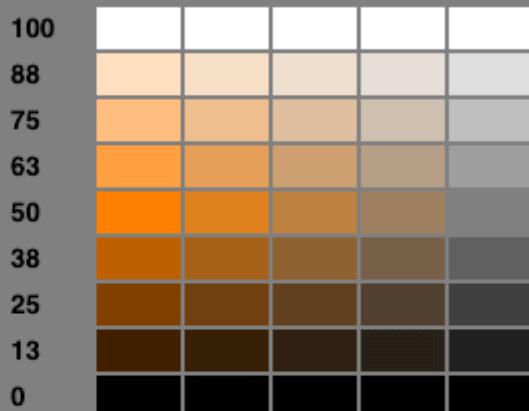
100% 75% 50% 25% 0%



30° Red-Yellows (=Oranges)

Saturation

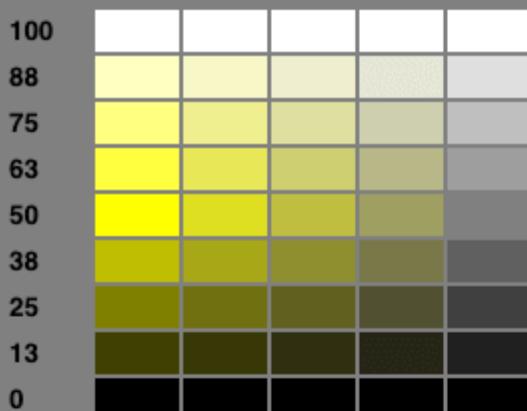
100% 75% 50% 25% 0%



60° Yellows

Saturation

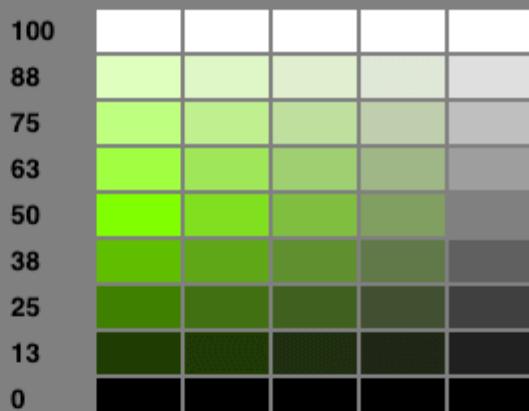
100% 75% 50% 25% 0%



90° Yellow-Greens

Saturation

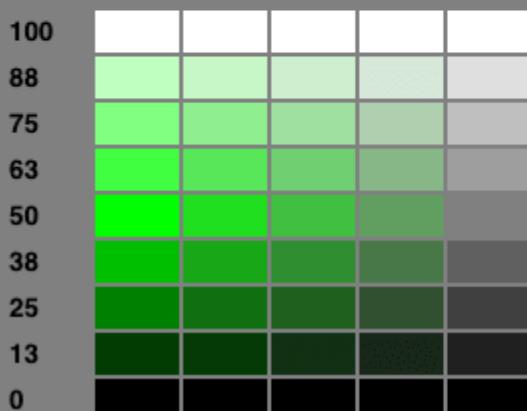
100% 75% 50% 25% 0%



120° Greens

Saturation

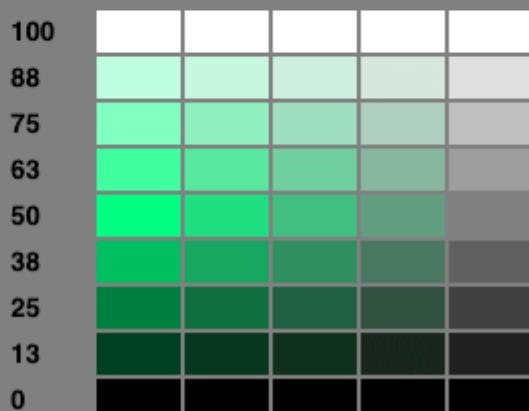
100% 75% 50% 25% 0%

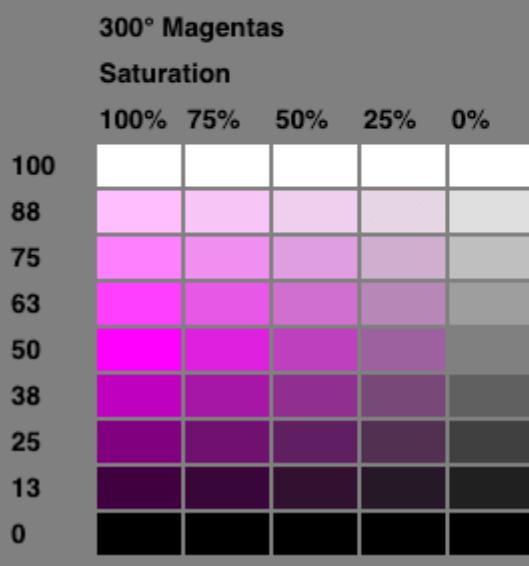
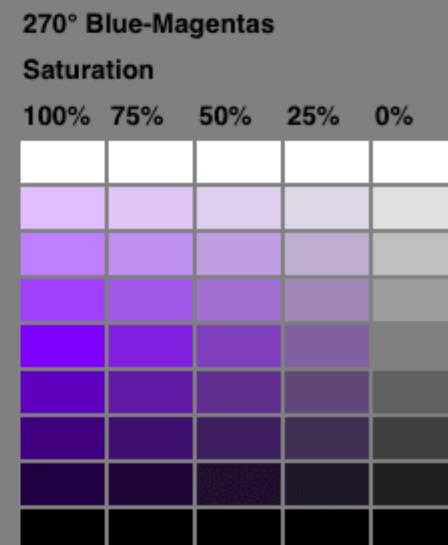
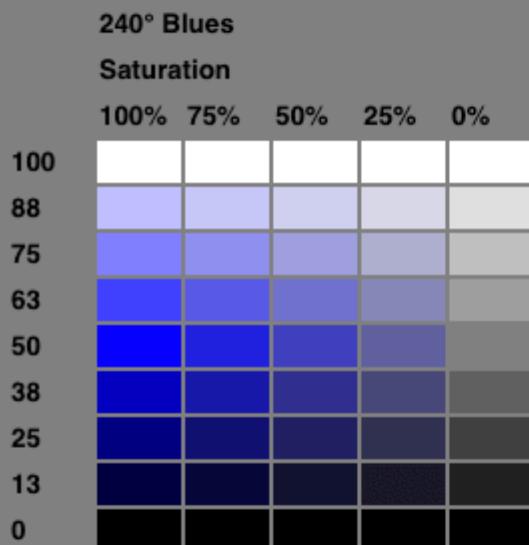
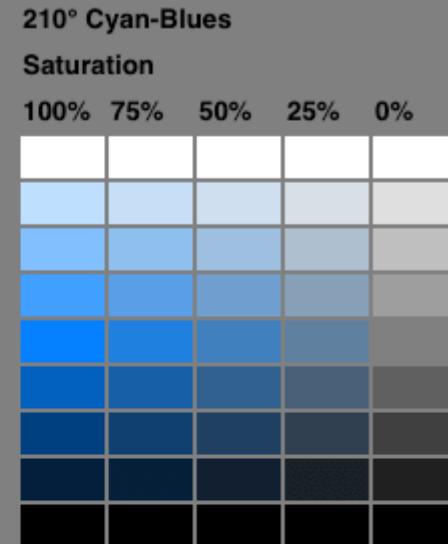
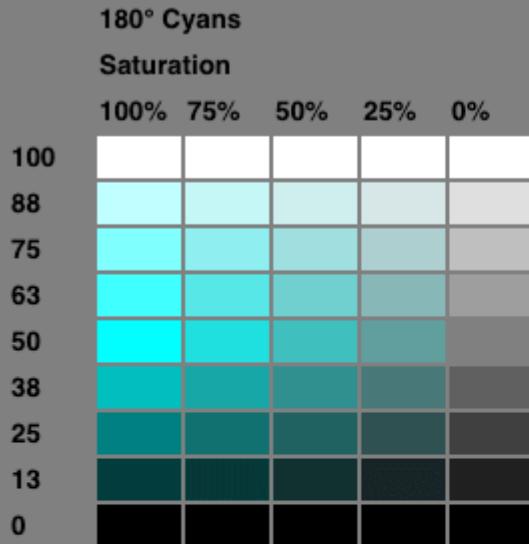


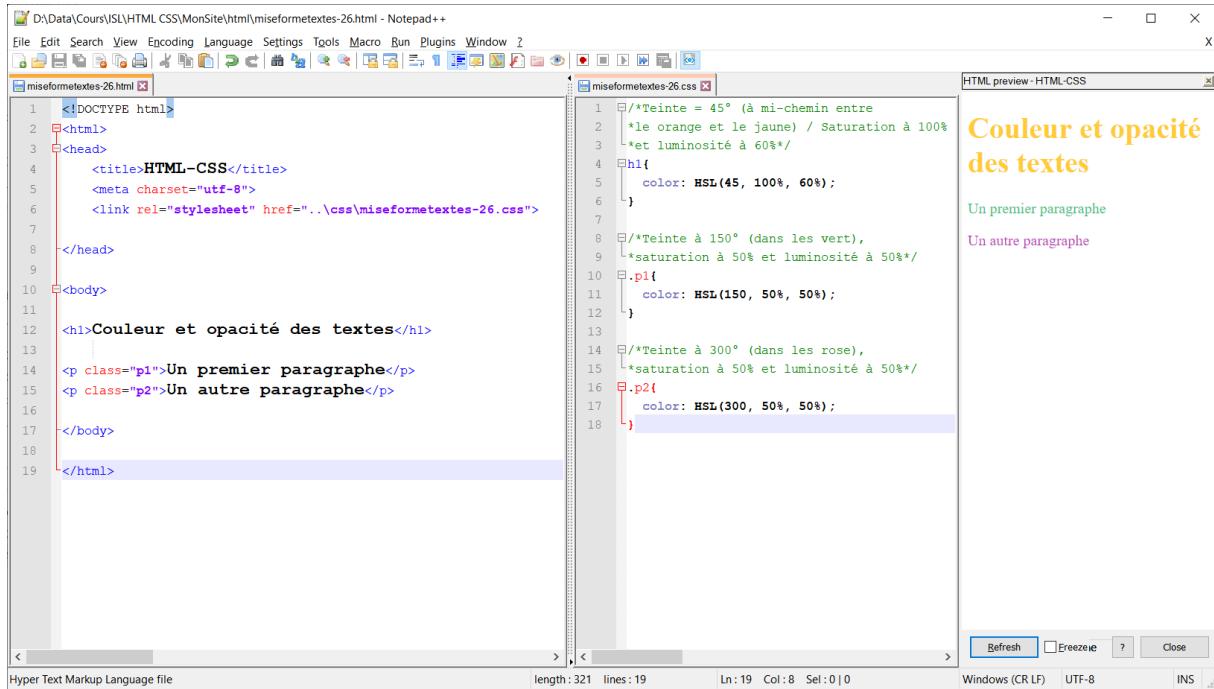
150° Green-Cyans

Saturation

100% 75% 50% 25% 0%







The screenshot shows a dual-pane Notepad++ interface. The left pane displays an HTML file named 'miseformetextes-26.html' containing the following code:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>HTML-CSS</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href=".css/miseformetextes-26.css">
7 
8 </head>
9 
10 <body>
11   <h1>Couleur et opacité des textes</h1>
12   <p class="p1">Un premier paragraphe</p>
13   <p class="p2">Un autre paragraphe</p>
14 
15 </body>
16 
17 </html>

```

The right pane displays a corresponding CSS file named 'miseformetextes-26.css' containing the following code:

```

1 /*Teinte = 45° (à mi-chemin entre
2 *le orange et le jaune) / Saturation à 100%
3 et luminosité à 60%*/
4 h1{
5   color: HSL(45, 100%, 60%);
6 }
7 
8 /*Teinte à 150° (dans les vert),
9 *saturation à 50% et luminosité à 50%*/
10 .p1{
11   color: HSL(150, 50%, 50%);
12 }
13 
14 /*Teinte à 300° (dans les rose),
15 *saturation à 50% et luminosité à 50%*/
16 .p2{
17   color: HSL(300, 50%, 50%);
18 }

```

A preview window titled 'HTML preview - HTML-CSS' shows the rendered output with the text 'Couleur et opacité des textes' in orange, 'Un premier paragraphe' in green, and 'Un autre paragraphe' in blue.

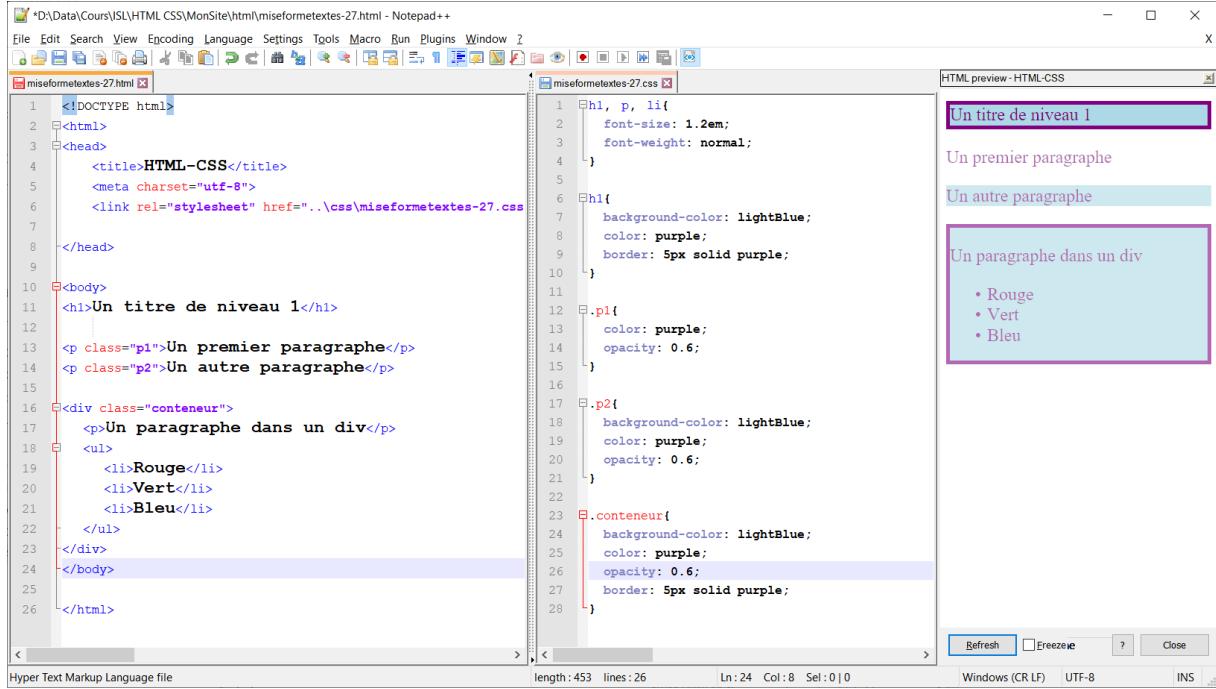
L'opacité des éléments et des textes en CSS

Jusqu'à présent, nous n'avons pas parlé d'opacité des couleurs en CSS. Nos couleurs étaient donc par défaut complètement opaques. Le CSS nous permet cependant de préciser un niveau d'opacité (ou de transparence, comme vous préférez) pour nos différents textes ou pour nos éléments de différentes façons.

Gérer l'opacité des éléments avec la propriété opacity

Tout d'abord, on va pouvoir rendre un élément HTML plus ou moins transparent grâce à la propriété CSS opacity.

Cette propriété va accepter **une valeur comprise entre 0 et 1** et qui va déterminer le niveau d'opacité d'un élément : la valeur 0 va rendre l'élément totalement transparent tandis que la valeur 1 (valeur par défaut) le rend totalement opaque.



The screenshot shows a Notepad++ interface with three panes. The left pane contains the HTML file 'miseformetextes-27.html' with the following content:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML-CSS</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="..\css\miseformetextes-27.css" />
7 
8   </head>
9 
10  <body>
11    <h1>Un titre de niveau 1</h1>
12 
13    <p class="p1">Un premier paragraphe</p>
14    <p class="p2">Un autre paragraphe</p>
15 
16   <div class="conteneur">
17     <p>Un paragraphe dans un div</p>
18     <ul>
19       <li>Rouge</li>
20       <li>Vert</li>
21       <li>Bleu</li>
22     </ul>
23   </div>
24 </body>
25 
26 </html>

```

The right pane contains the CSS file 'miseformetextes-27.css' with the following content:

```

1 h1, p, li{
2   font-size: 1.2em;
3   font-weight: normal;
4 }
5 
6 h1{
7   background-color: lightBlue;
8   color: purple;
9   border: 5px solid purple;
10 }
11 
12 .p1{
13   color: purple;
14   opacity: 0.6;
15 }
16 
17 .p2{
18   background-color: lightBlue;
19   color: purple;
20   opacity: 0.6;
21 }
22 
23 .conteneur{
24   background-color: lightBlue;
25   color: purple;
26   opacity: 0.6;
27   border: 5px solid purple;
28 }

```

The bottom-right pane shows the HTML preview with the following structure:

- Un titre de niveau 1
- Un premier paragraphe
- Un autre paragraphe
- Un paragraphe dans un div
 - * Rouge
 - * Vert
 - * Bleu

Ici, j'attire cependant votre attention sur un point important : la propriété opacity ne va pas gérer la transparence d'un texte mais bien définir le niveau d'opacité d'un élément en soi.

Ainsi, si l'élément possède une couleur de fond ou des bordures, celles-ci vont également être impactées par opacity et vont donc possiblement être semi transparentes. De même, en appliquant cette propriété à un élément conteneur div, nous allons rendre le div en soi et tous les éléments qu'il contient semi-transparents (sauf si une règle contraire est précisée pour chaque élément du div bien évidemment).

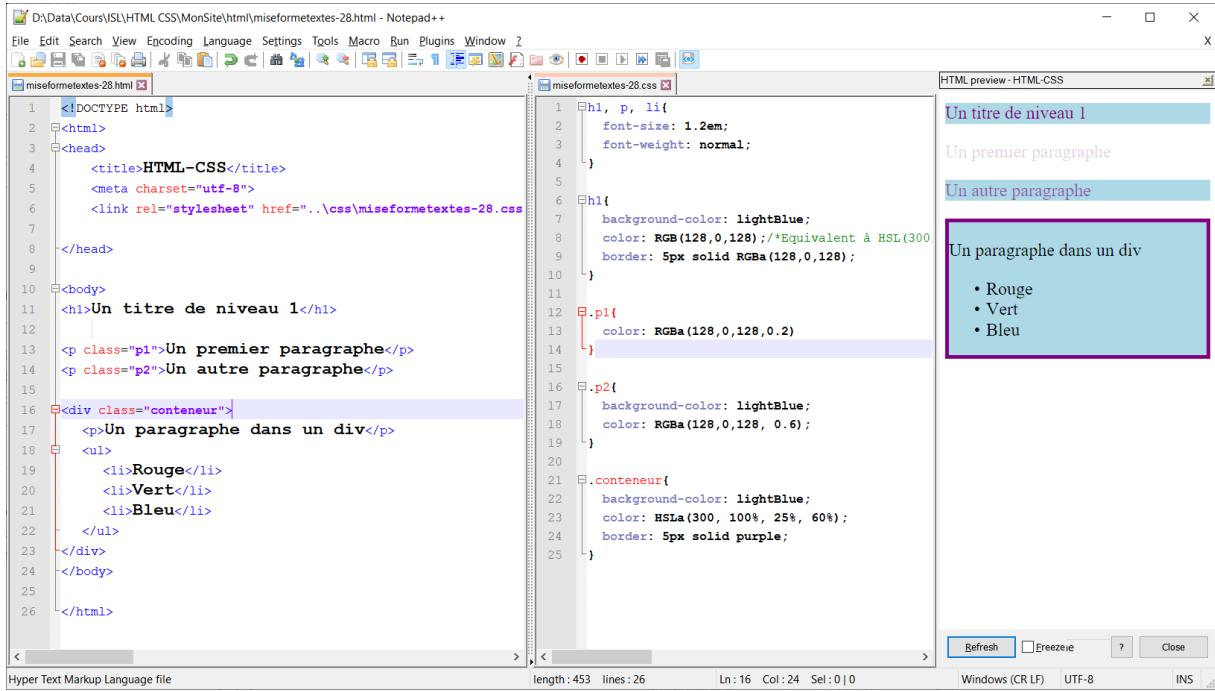
Ce comportement va parfois être le comportement voulu mais ce n'est pas celui attendu dans le cas où l'on souhaite simplement gérer l'opacité des textes (et uniquement des textes) de nos éléments.

Pour faire cela, nous allons plutôt devoir utiliser des variantes des notations RGB et HSL : les notations **RGBa** et **HSLa**.

Gérer l'opacité des textes avec les notations RGBa ou HSLa

Les notations RGBa et HSLa vont accepter une valeur de plus qui va correspondre au niveau d'opacité de la couleur (le « a » est l'abréviation de « **alpha channel** »).

Nous allons ici à nouveau devoir préciser un chiffre compris entre 0 et 1 et indiquant le niveau d'opacité de nos textes (0 = texte transparent / 1 = texte opaque).



The screenshot shows a Notepad++ interface with three panes:

- Left pane:** HTML code (miseformetextes-28.html) containing an

title, two elements, and a element with a nested .
- Middle pane:** CSS code (miseformetextes-28.css) defining styles for

, , and elements.
- Right pane:** Preview of the rendered HTML. It shows the

with a light blue background and white text. The first has a light blue background and white text. The inside the has a light blue background and white text. The list items (-) have a light blue background and white text.

Comme vous pouvez le voir, cette fois-ci seul le niveau d'opacité de nos textes change.

Notez que les notations hexadécimales supportent également la notion de transparence. Pour indiquer un niveau d'opacité, nous allons devoir ici préciser un quatrième jeu de caractères compris entre 00 (couleur totalement transparente) et FF (couleur totalement opaque).

Cependant, je vous déconseille d'utiliser les notations hexadécimales de cette manière pour le moment car la gestion de la transparence pour ce type de valeur n'est pas encore une recommandation officielle et donc le support par les différents navigateurs n'est pas forcément assuré. Utiliser plutôt les notations RGBa ou HSLa si vous souhaitez définir des couleurs avec un niveau de transparence.



04/223.11.31 - www.isl.be

INSTITUT SAINT-LAURENT

ENSEIGNEMENT DE PROMOTION SOCIALE

Baccalauréat en informatique