

HTML



CSS



Mises en forme de textes en CSS

Rédigé par: Jean-Luc Colson

Date première rédaction: Jan 2020

SUIVI DES MODIFICATIONS A LA FICHE

Série	Date	Page(s) modifiée(s)	Raison

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.

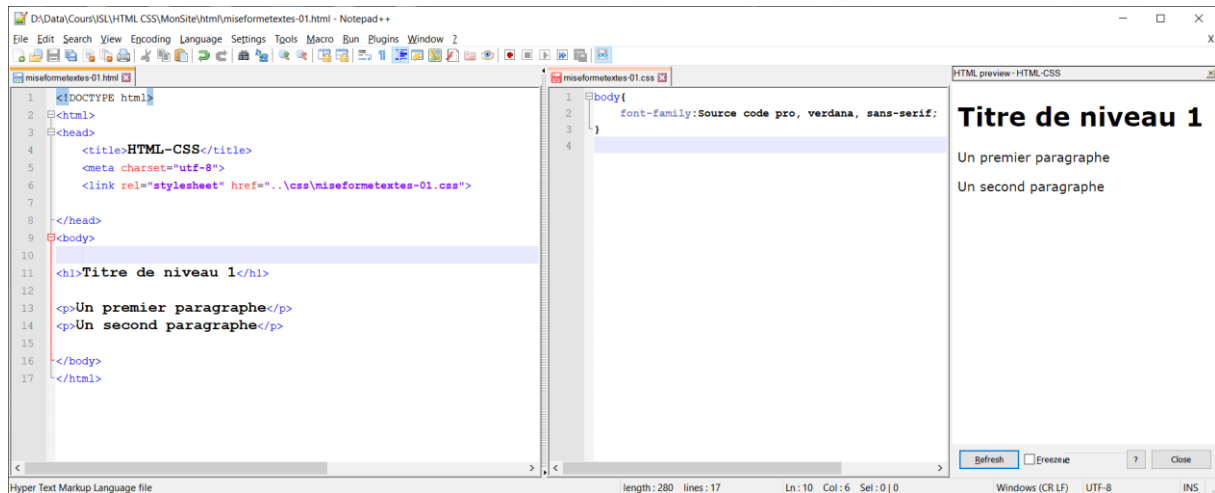
Il existe aujourd'hui un très grand nombre de polices d'écritures disponibles et parmi lesquelles on va pouvoir choisir mais il faut savoir que certaines versions de certains navigateurs que peuvent posséder vos visiteurs ne vont pas supporter certaines polices.

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).



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 ou « 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

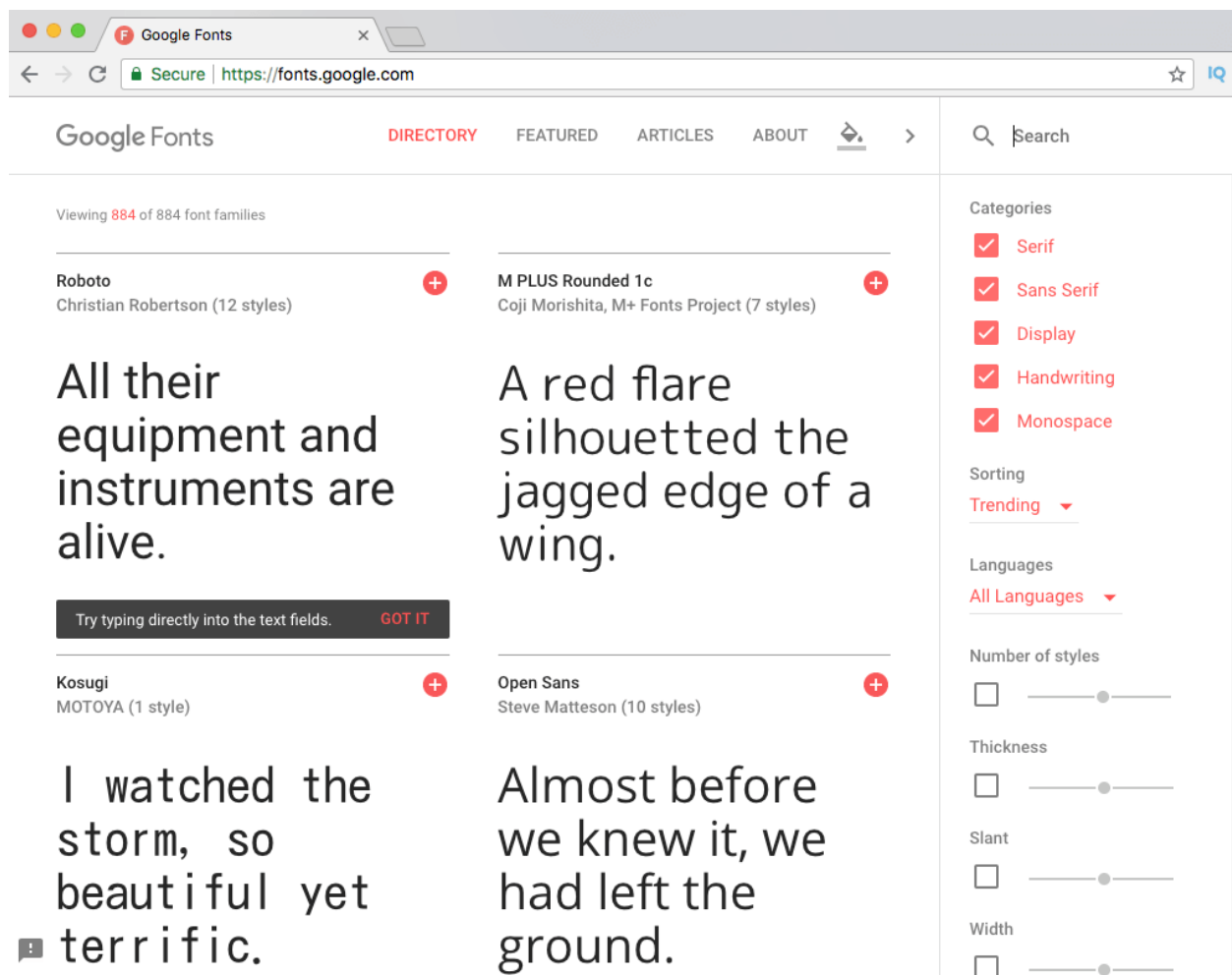
Google a développé sa propre liste de polices d'écriture appelées les **Google Fonts**.

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.

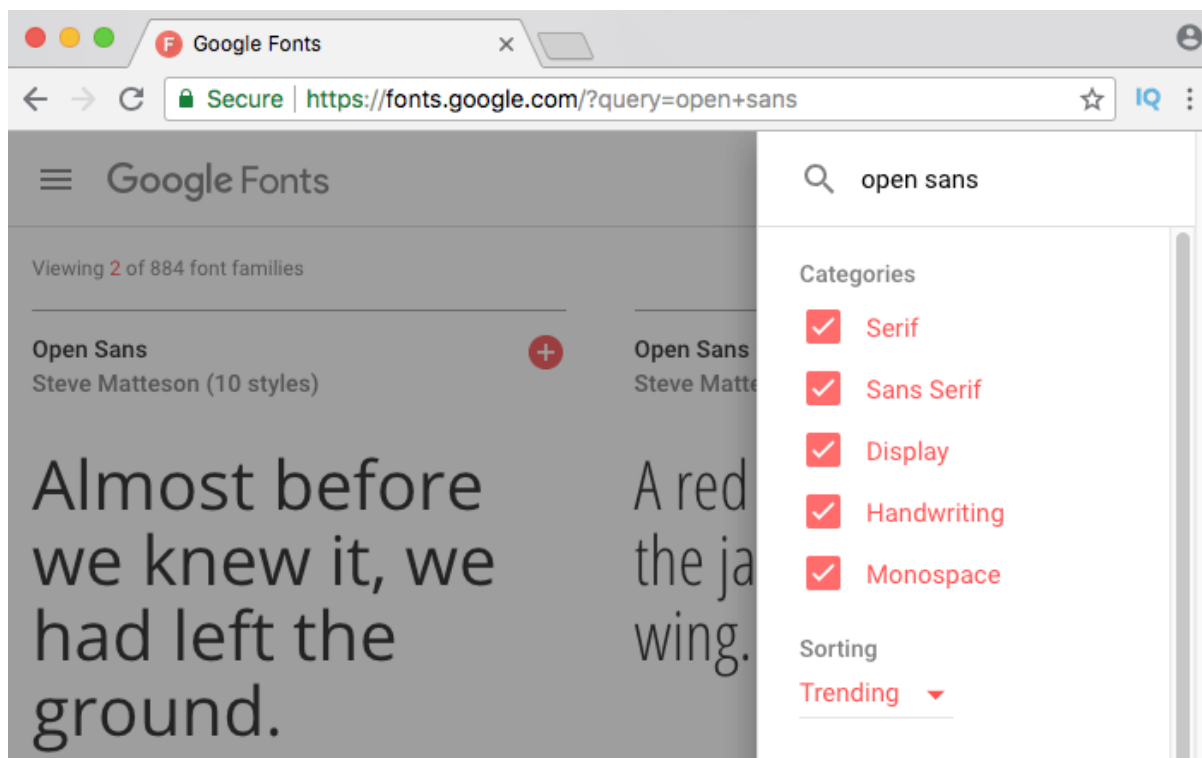
Pour utiliser les Google Fonts, il va tout d'abord nous falloir choisir une police dans la liste proposée sur [le site de Google dédié](https://fonts.google.com).



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

ABCČĆDĐEFGHIJKLMNOPQRSŠTUVWX
 YZŽabcčćdđefghijklmnopqrsštuvwxyz
 ŽАБВГГДЂЕЁЄЖЗСИІЙЈКЛЉМНЊОПР
 СТЋУЎФХЦЧЏШЩЪЫЬЭЮЯабвггдђеё
 єжзсиіійјклљмнњопрстћуўфхцчџшщ
 ѡыьэюяАΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧ
 ΨΩαβγδεζηθικλμνξοπρστυφχψωά'Αέ'Ε
 έ'Ηί'Ι'Ιό'Ο'Ού'Ό'Ό'Υ'Υ'Ω'Ά'Ά'Έ'Έ'Ό'Ό'
 1234567890'?''"!"(%)[#{@}/&\<-+÷×=>®©\$
 €£¥¢::;,. *

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

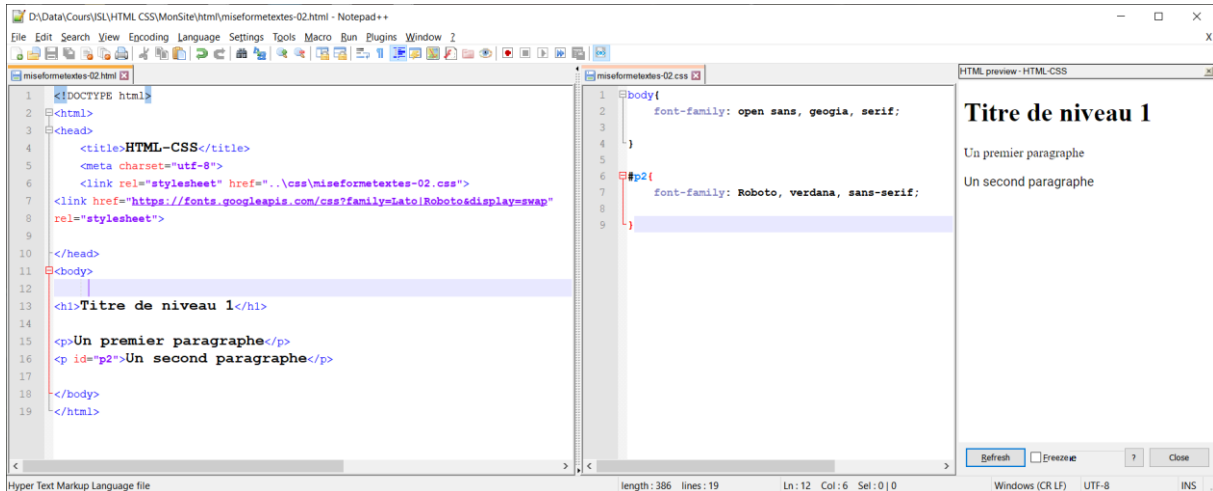
Use the following CSS properties to specify these families:

```
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 feuille 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++ window with three panes. The left pane displays the HTML code for 'miseformetextes-02.html'. The middle pane displays the CSS code for 'miseformetextes-02.css'. The right pane shows a preview of the HTML document, which includes a title 'Titre de niveau 1' and two paragraphs: 'Un premier paragraphe' and 'Un second 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-02.css">
7   <link href="https://fonts.googleapis.com/css?family=Lato|Roboto&display=swap"
8     rel="stylesheet">
9 </head>
10 <body>
11
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>

```

```

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

```

Titre de niveau 1
Un premier paragraphe
Un second paragraphe

length: 386 lines: 19 Ln: 12 Col: 6 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

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.

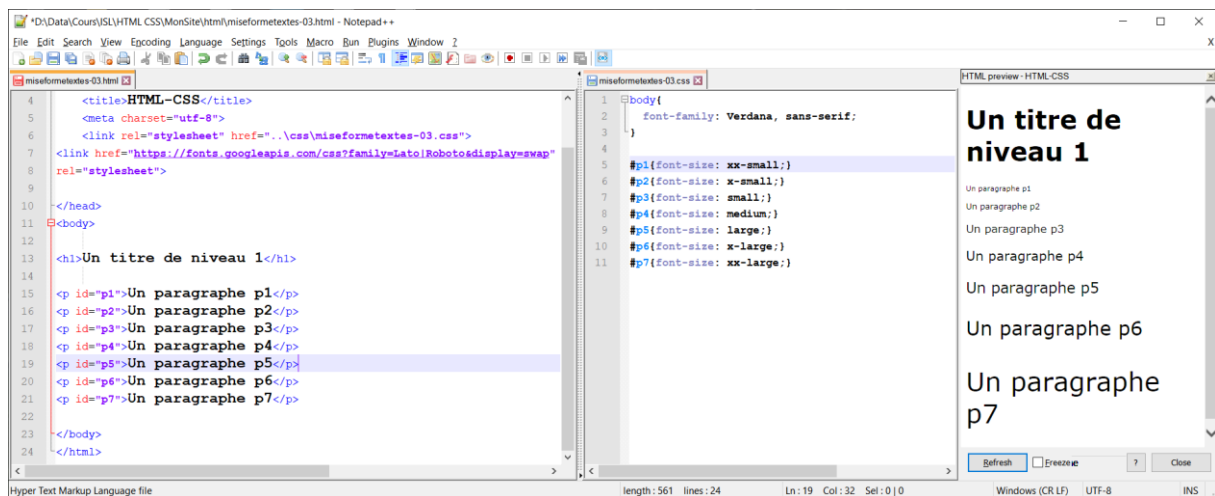
Les valeurs « mot clef »

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.

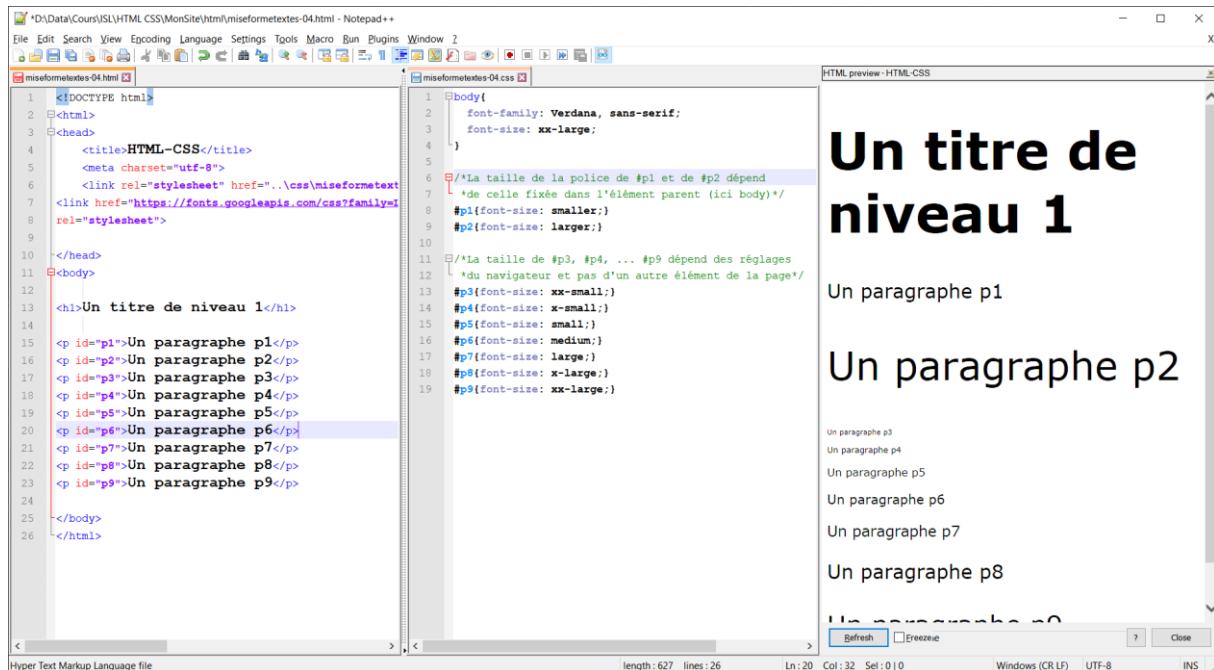
- **xx-small** : la taille de la police sera égale à la moitié de celle définie dans le navigateur ;
- **x-small** : la taille de la police sera égale à 60% de celle définie dans le navigateur ;

- **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.
-



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.



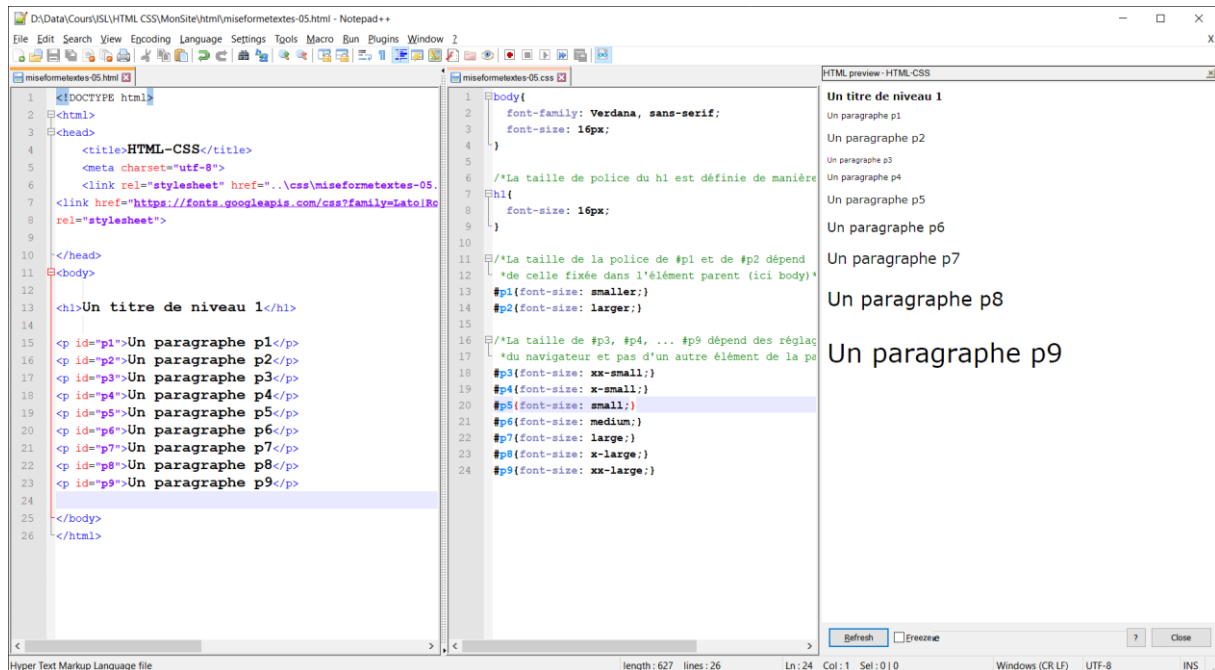
Ici, la taille de la police des paragraphes p1 et p2 dépend de la taille de la police de leur élément parent qui est dans ce cas l'élément body et qui a été réglée à font-size: xx-large.

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.



```

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|Roboto" rel="stylesheet">
8 </head>
9 <body>
10 <h1>Un titre de niveau 1</h1>
11
12 <p id="p1">Un paragraphe p1</p>
13 <p id="p2">Un paragraphe p2</p>
14 <p id="p3">Un paragraphe p3</p>
15 <p id="p4">Un paragraphe p4</p>
16 <p id="p5">Un paragraphe p5</p>
17 <p id="p6">Un paragraphe p6</p>
18 <p id="p7">Un paragraphe p7</p>
19 <p id="p8">Un paragraphe p8</p>
20 <p id="p9">Un paragraphe p9</p>
21 </body>
22 </html>
  
```

```

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

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.

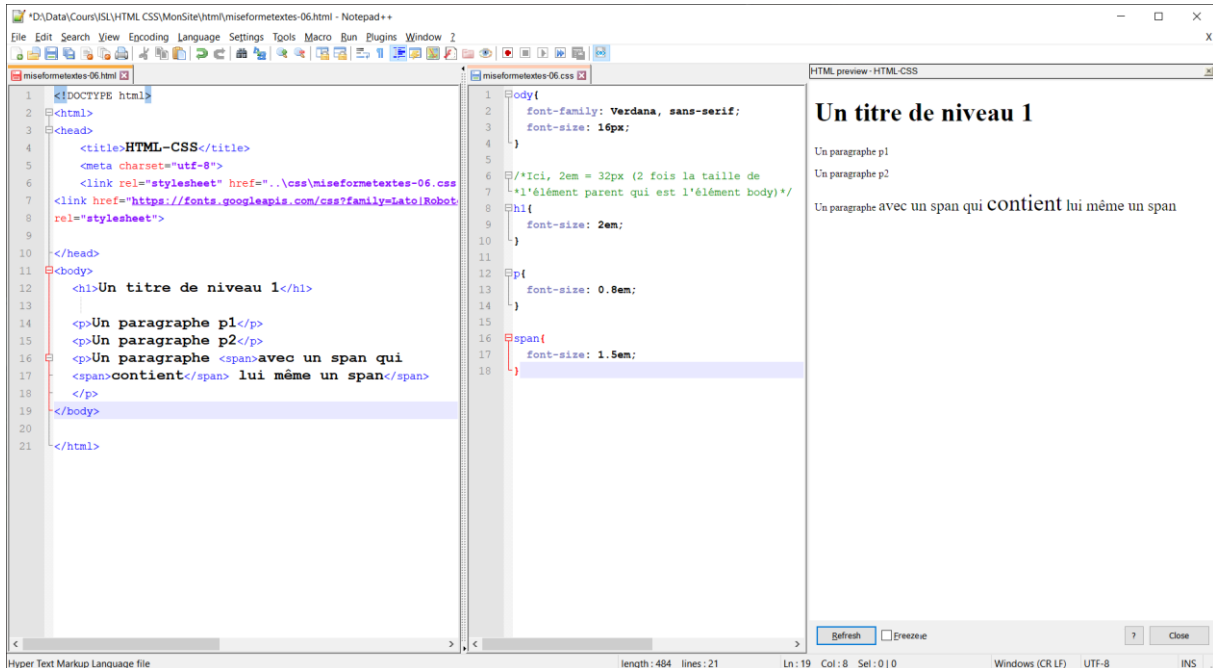
En déclarant font-size : 1em pour un élément on demande à ce que la valeur de la propriété soit la même que celle définie pour l'élément parent, en déclarant font-size : 1.5em on demande à ce que la valeur de la propriété soit égale à 1,5 fois celle définie pour l'élément parent et etc.

Note : Attention ici à bien utiliser des points à la place des virgules (notations anglo-saxonnes) lorsque vous précisez 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 entrainer 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 le taille du span parent.



```

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"
8   rel="stylesheet">
9 </head>
10 <body>
11 <h1>Un titre de niveau 1</h1>
12 <p>Un paragraphe p1</p>
13 <p>Un paragraphe p2</p>
14 <p>Un paragraphe <span>avec un span qui
15 <span>contient</span> lui même un span</span>
16 </p>
17 </body>
18 </html>

```

```

1 body{
2   font-family: Verdana, sans-serif;
3   font-size: 16px;
4 }
5
6 /*Ici, 2em = 32px (2 fois la taille de
7 *l'élément parent qui est l'élément body)*/
8 h1{
9   font-size: 2em;
10 }
11
12 p{
13   font-size: 0.8em;
14 }
15
16 span{
17   font-size: 1.5em;
18 }

```

Un titre de niveau 1

Un paragraphe p1

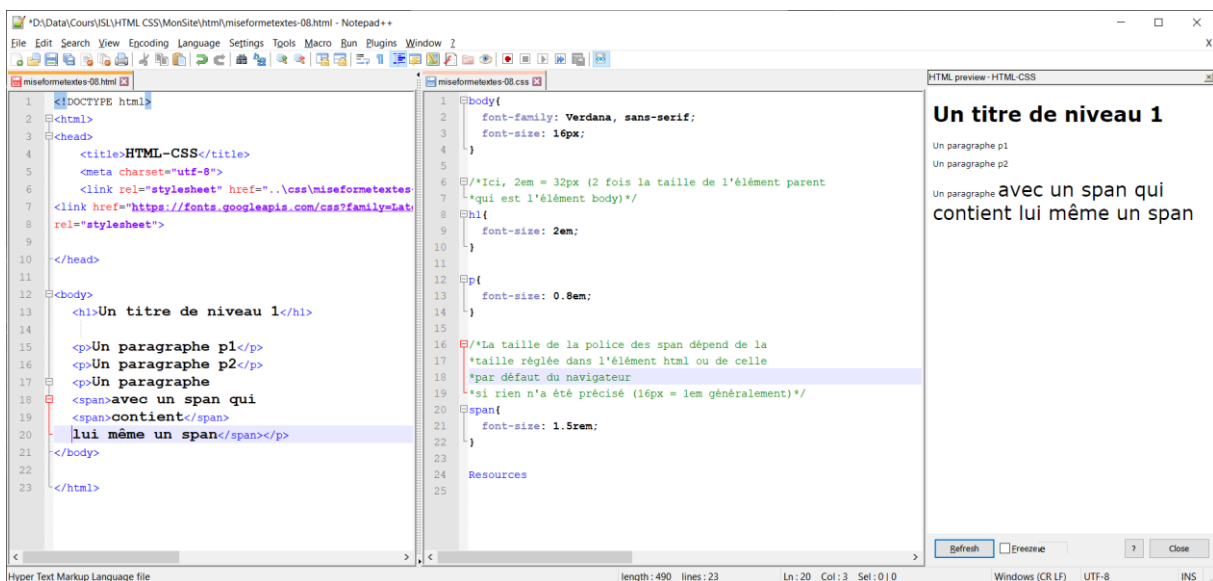
Un paragraphe p2

Un paragraphe avec un span qui contient lui même un span

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.



```

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"
8   rel="stylesheet">
9 </head>
10 <body>
11 <h1>Un titre de niveau 1</h1>
12 <p>Un paragraphe p1</p>
13 <p>Un paragraphe p2</p>
14 <p>Un paragraphe
15 <span>avec un span qui
16 <span>contient</span>
17 <span>lui même un span</span></p>
18 </body>
19 </html>

```

```

1 body{
2   font-family: Verdana, sans-serif;
3   font-size: 16px;
4 }
5
6 /*Ici, 2em = 32px (2 fois la taille de l'élément parent
7 *qui est l'élément body)*/
8 h1{
9   font-size: 2em;
10 }
11
12 p{
13   font-size: 0.8em;
14 }
15
16 /*La taille de la police des span dépend de la
17 *taille réglée dans l'élément html ou de celle
18 *par défaut du navigateur
19 *si rien n'a été précisé (16px = 1em généralement)*/
20 span{
21   font-size: 1.5em;
22 }
23
24 Resources
25

```

Un titre de niveau 1

Un paragraphe p1

Un paragraphe p2

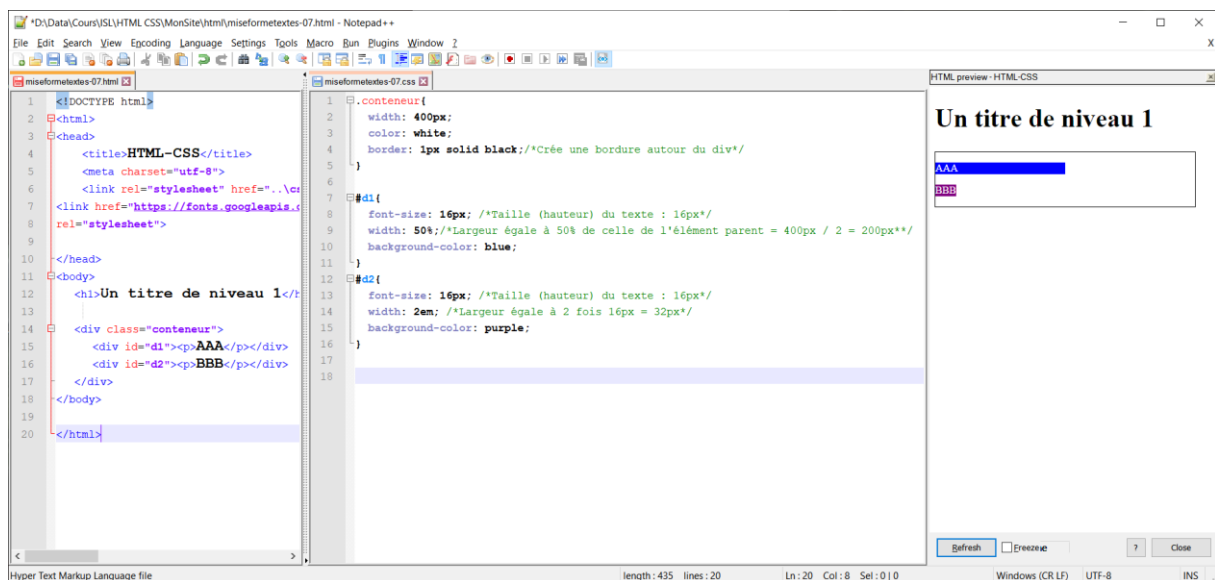
Un paragraphe avec un span qui contient lui même un span

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 équivalent 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 % :



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.

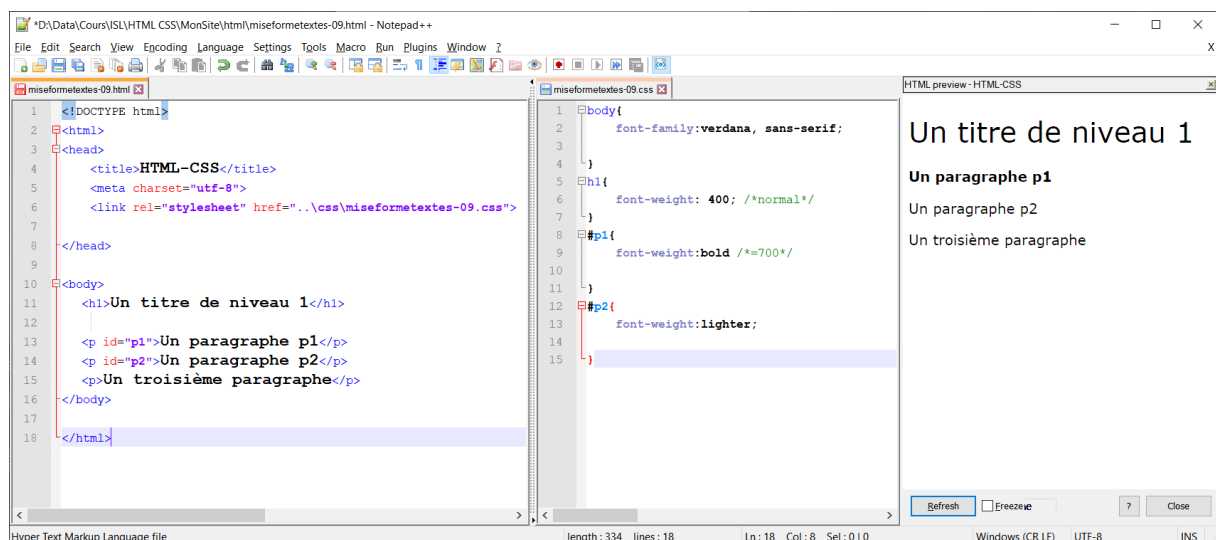
Cette propriété va pouvoir prendre un mot clef ou un chiffre exprimé en centaine(s) en valeur. Nous allons ainsi pouvoir choisir parmi les valeurs suivantes :

- 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.



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 :

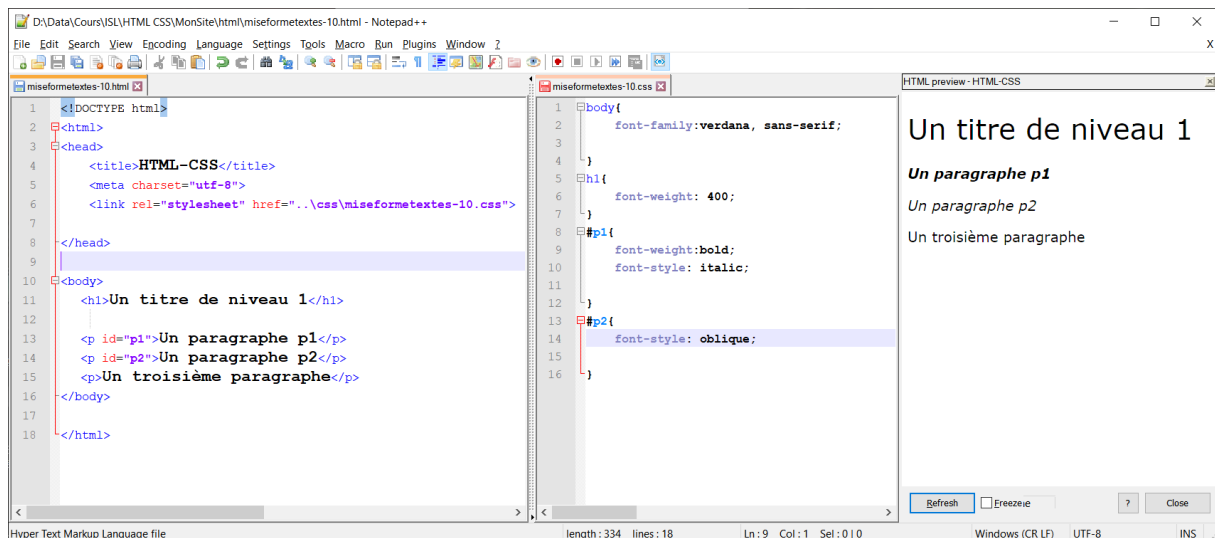
- **normal** : valeur par défaut, les caractères seront droits ;

- **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 mêmes 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).



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>HTML-CSS</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="css/miseformatextes-10.css">
7 </head>
8
9 <body>
10
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 </body>
17
18 </html>

```

```

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

```

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) :

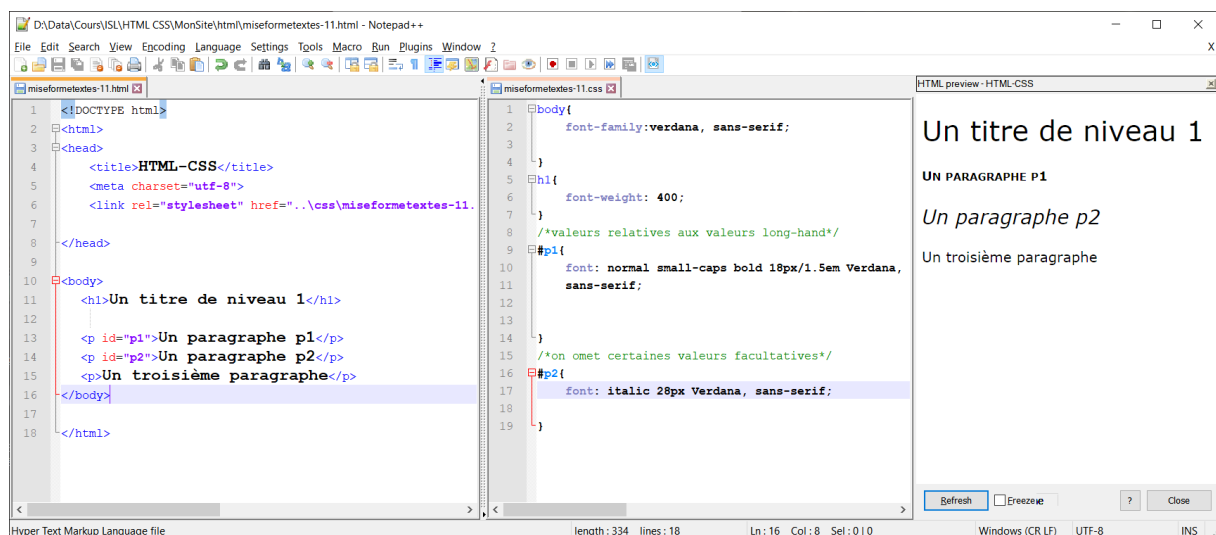
1. La valeur relative à la propriété **font-style** (facultatif) ;
2. La valeur relative à la propriété **font-variant** (facultatif) ;
3. La valeur relative à la propriété **font-weight** (facultatif) ;
4. La valeur relative à la propriété **font-size (obligatoire)/line-height** (facultatif) ;
5. La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



```

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.
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 </body>
17 </html>

```

```

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

```

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 ?

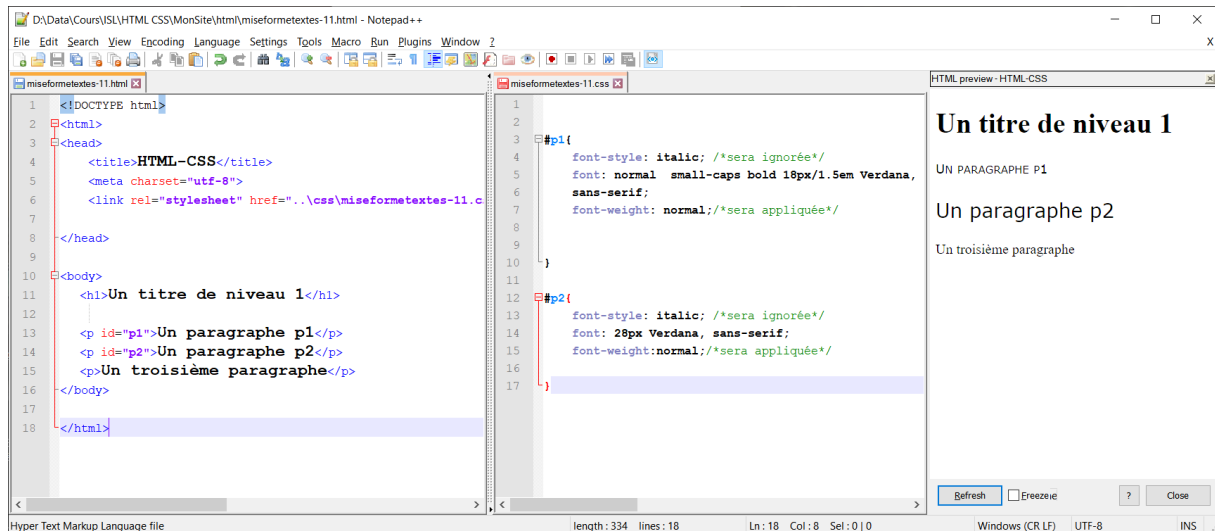
Imaginons ici que j'ai écrit normal à la place de italic. Comment savoir si cette valeur normal a été déclarée comme valeur pour ma propriété font-style, font-variant ou font-weight ?

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.



```

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.c
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 </body>
17
18 </html>

```

```

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
12 #p2{
13 font-style: italic; /*sera ignorée*/
14 font: 28px Verdana, sans-serif;
15 font-weight:normal; /*sera appliquée*/
16
17 }
18

```

Un titre de niveau 1

UN PARAGRAPHE P1

Un paragraphe p2

Un troisième paragraphe

Refresh Freeze ? Close

length: 334 lines: 18 Ln: 18 Col: 8 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

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.

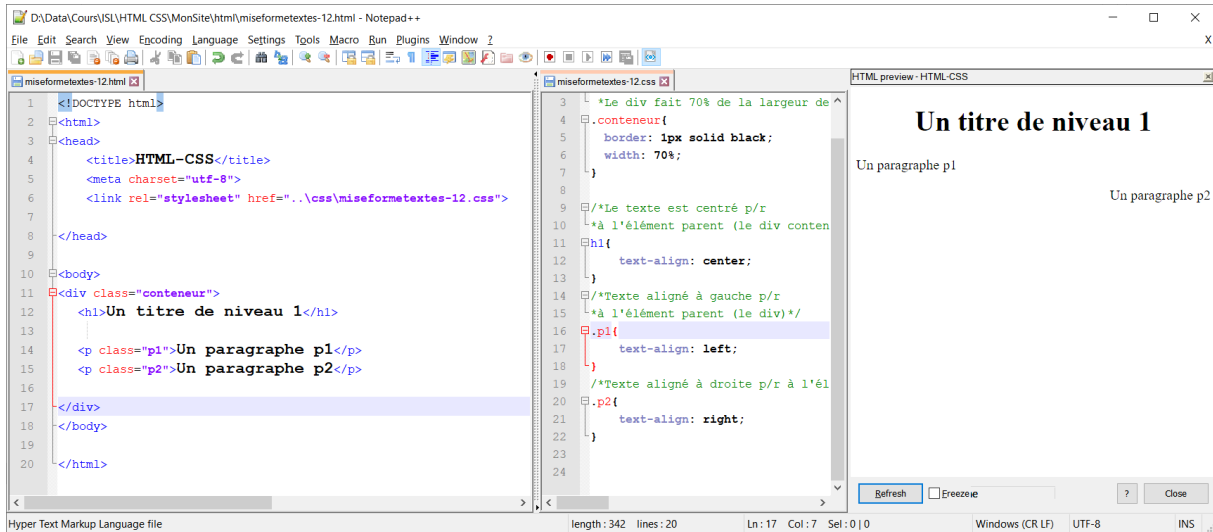
La propriété text-align

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.



```

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
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>

```

```

3 /*Le div fait 70% de la largeur de
4
5 .conteneur{
6   border: 1px solid black;
7   width: 70%;
8 }
9
10 /*Le texte est centré p/r
11 à l'élément parent (le div conten
12
13 h1{
14   text-align: center;
15 }
16
17 /*Texte aligné à gauche p/r
18 à l'élément parent (le div)*/
19
20 p1{
21   text-align: left;
22 }
23
24 /*Texte aligné à droite p/r à l'él
25
26 p2{
27   text-align: right;
28 }

```

Un titre de niveau 1

Un paragraphe p1

Un paragraphe p2

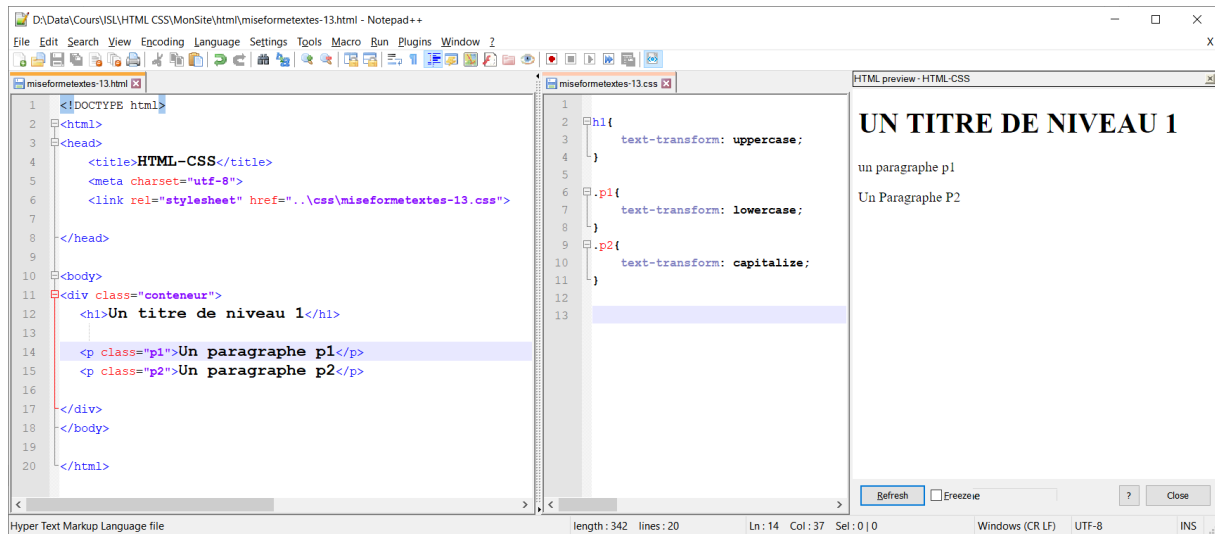
La propriété text-transform

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 ;



La propriété text-decoration

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

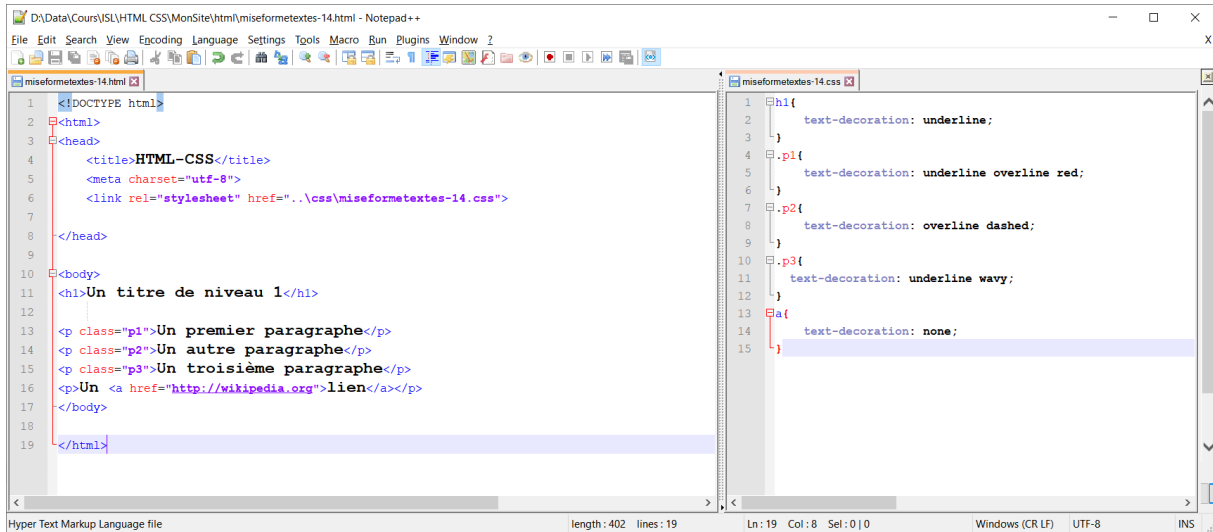
On va pouvoir lui passer jusqu'à trois valeurs qui vont correspondre au type de décoration (valeur obligatoire), à la couleur de la décoration (valeur facultative, couleur actuelle utilisée par défaut) et au style de la décoration (solide par défaut).

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.



```

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 <a href="http://wikipedia.org">lien</a></p>
17 </body>
18
19 </html>

```

```

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 }

```



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 :

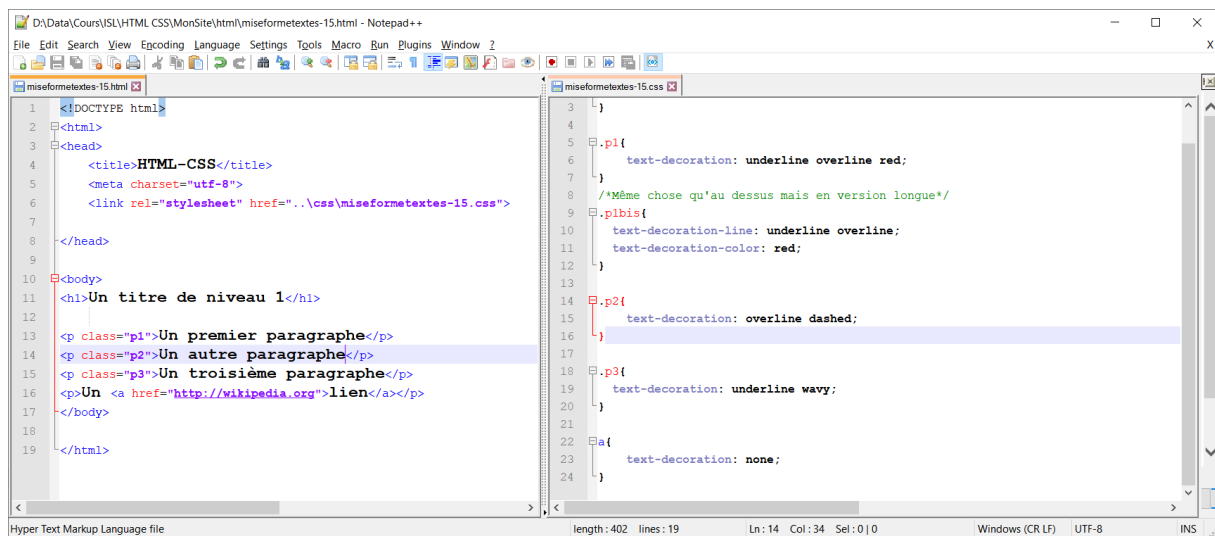
- **underline** : ajoute un trait de soulignement au texte ;
- **overline** : ajoute un trait de surlignement au texte ;

- **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 :

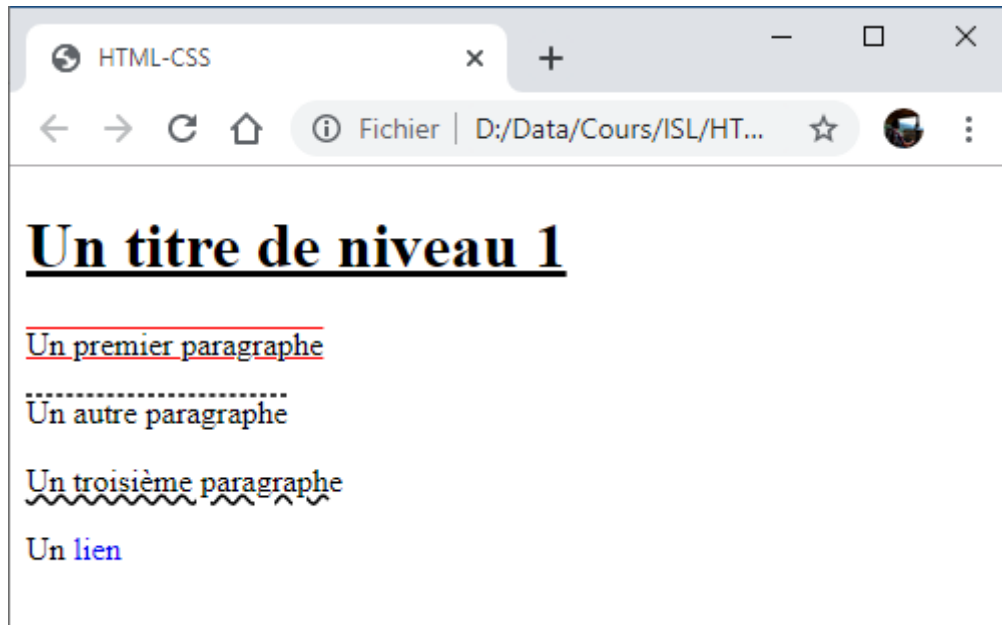
- **solid** : valeur par défaut ; le trait de décoration sera solide ;
- **double** : le trait de décoration sera double ;
- **dotted** : le trait de décoration sera en pointillés ;
- **dashed** : le trait de décoration sera en tirets ;
- **wavy** : le trait de décoration sera courbé.

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.



```
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 </html>
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543

```

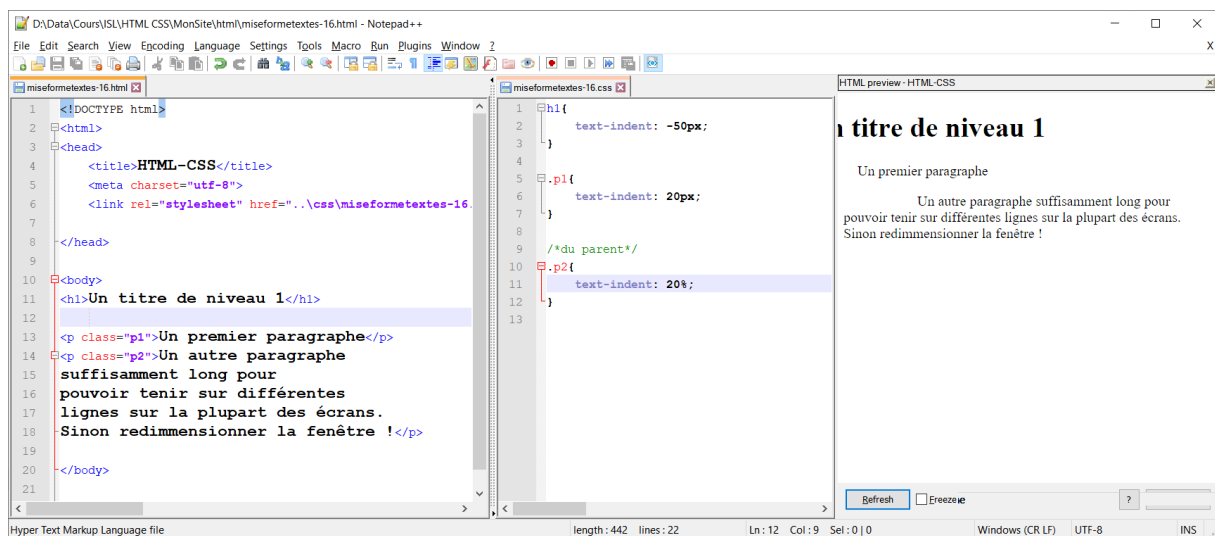


La propriété text-indent

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.

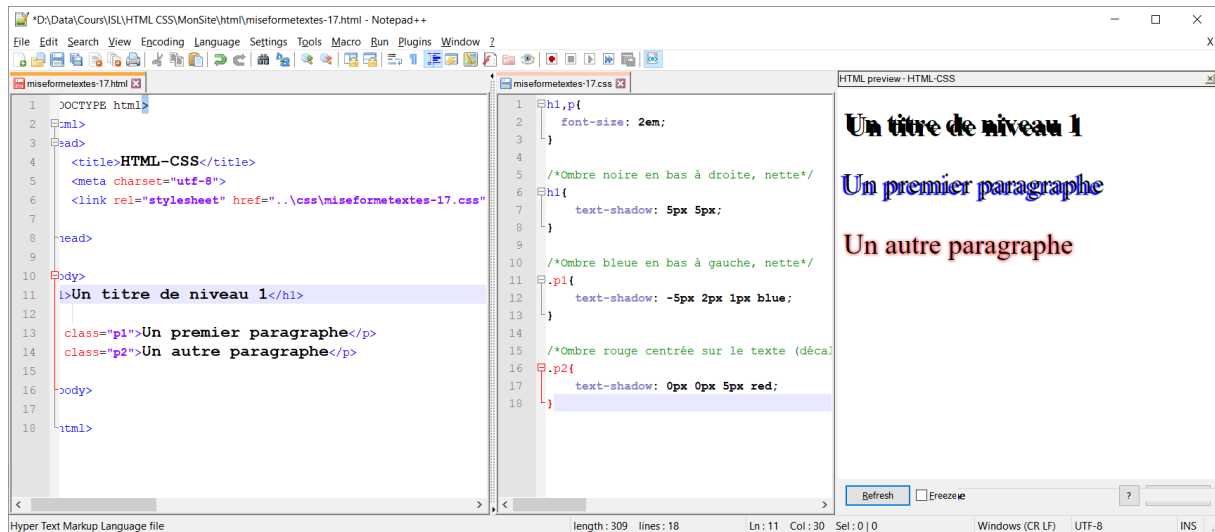


La propriété text-shadow

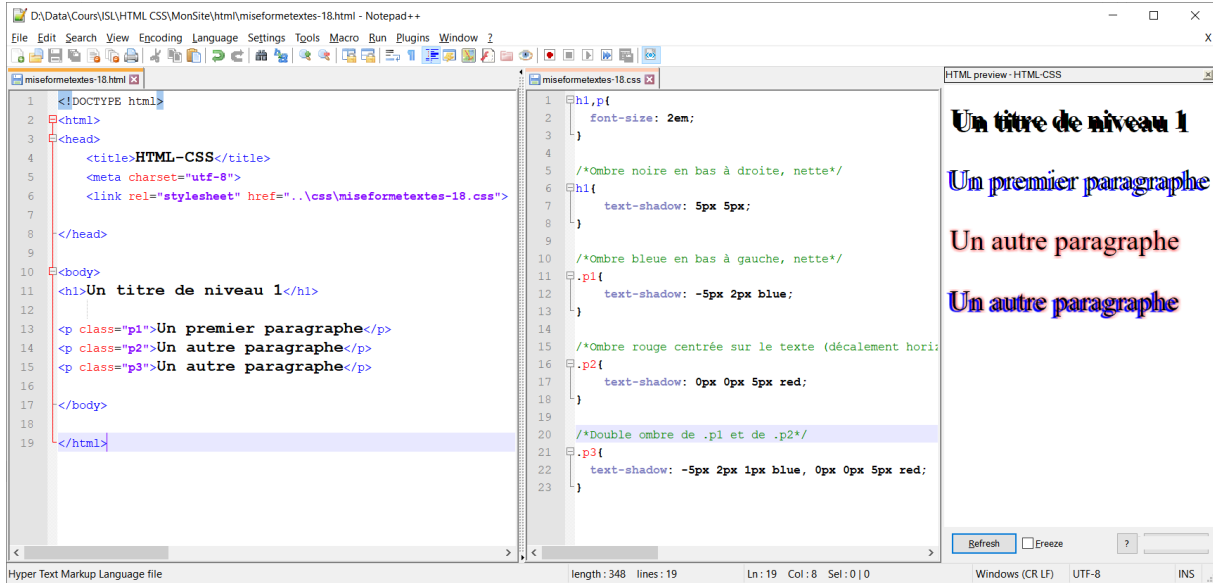
La propriété **text-shadow** va nous permettre d'ajouter des ombres autour de nos textes. Cette propriété est relativement complexe à maîtriser puisqu'elle va pouvoir utiliser jusqu'à 4 valeurs pour définir précisément une ombre et surtout puisqu'on va pouvoir appliquer plusieurs ombres différentes à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



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.



```

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 </head>
8
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>

```

```

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 }

```

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 :

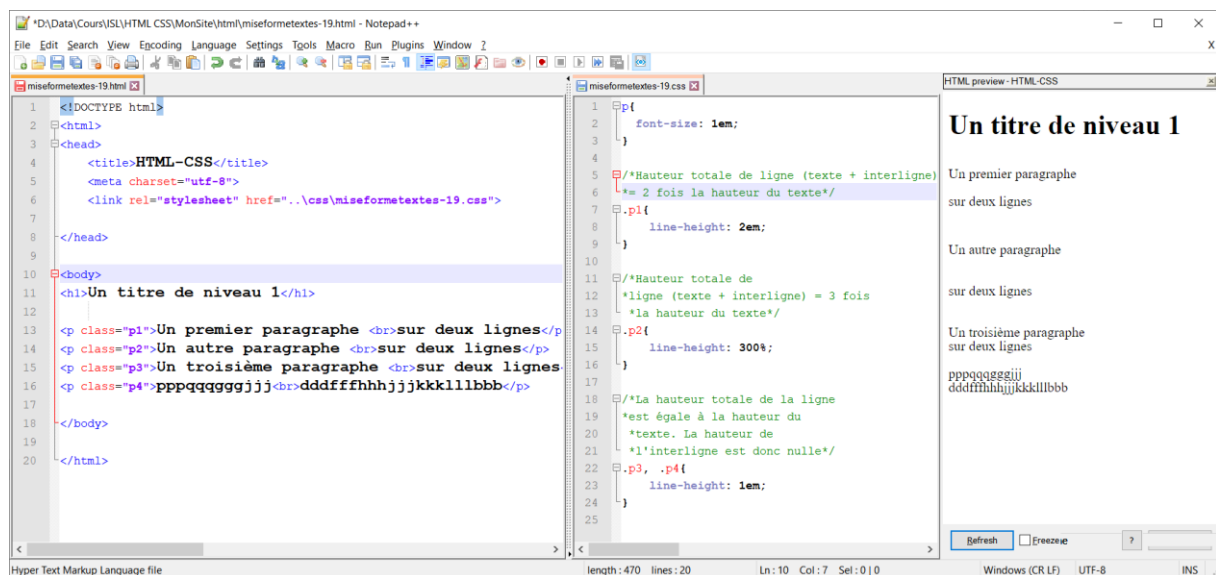
- La propriété **line-height** ;
- La propriété **letter-spacing** ;
- La propriété **word-spacing**.

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, indique `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.



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

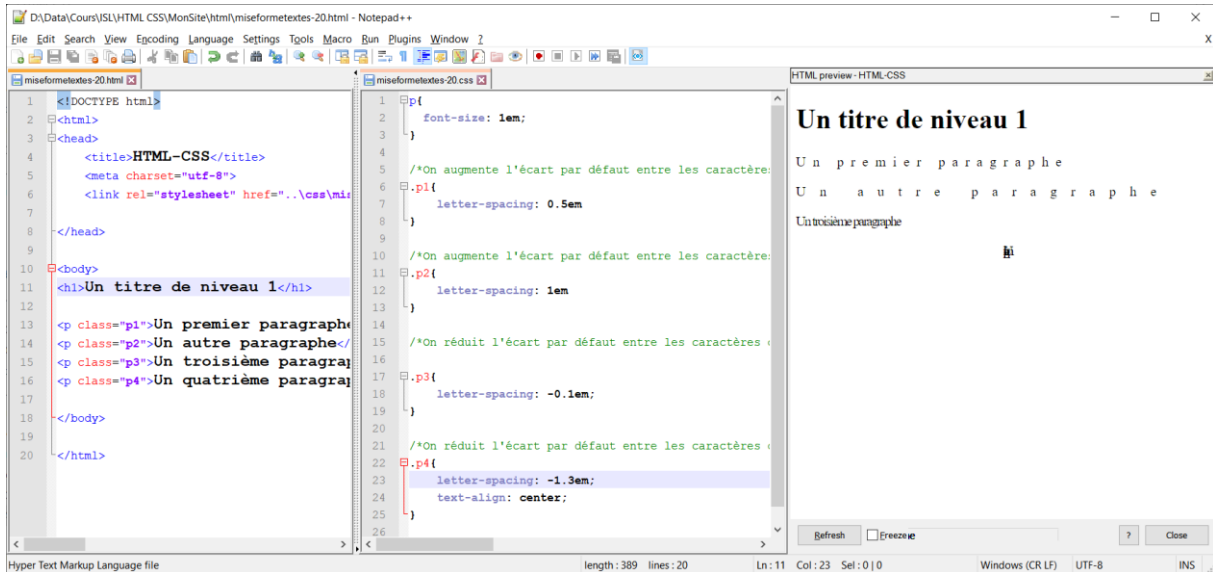
La propriété **letter-spacing** va nous permettre de définir l'espace entre les différents caractères de notre police.

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.



```

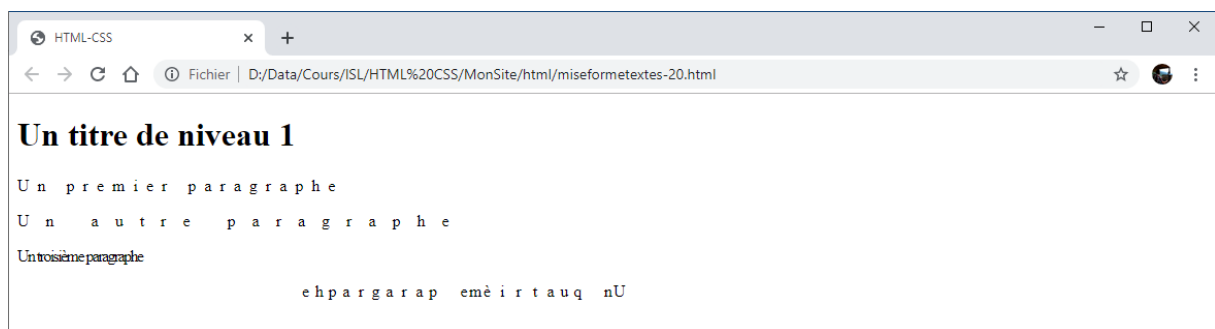
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>HTML-CSS</title>
5 <meta charset="utf-8">
6 <link rel="stylesheet" href="..\css\miseformetextes-20.css">
7 </head>
8 <body>
9 <h1>Un titre de niveau 1</h1>
10 <p class="p1">Un premier paragraphe</p>
11 <p class="p2">Un autre paragraphe</p>
12 <p class="p3">Un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

```

```

1 p{
2   font-size: 1em;
3 }
4
5 /*On augmente l'écart par défaut entre les caractères:
6 .p1{
7   letter-spacing: 0.5em
8 }
9
10 /*On augmente l'écart par défaut entre les caractères:
11 .p2{
12   letter-spacing: 1em
13 }
14
15 /*On réduit l'écart par défaut entre les caractères:
16 .p3{
17   letter-spacing: -0.1em;
18 }
19
20 /*On réduit l'écart par défaut entre les caractères:
21 .p4{
22   letter-spacing: -1.3em;
23   text-align: center;
24 }
25
26

```

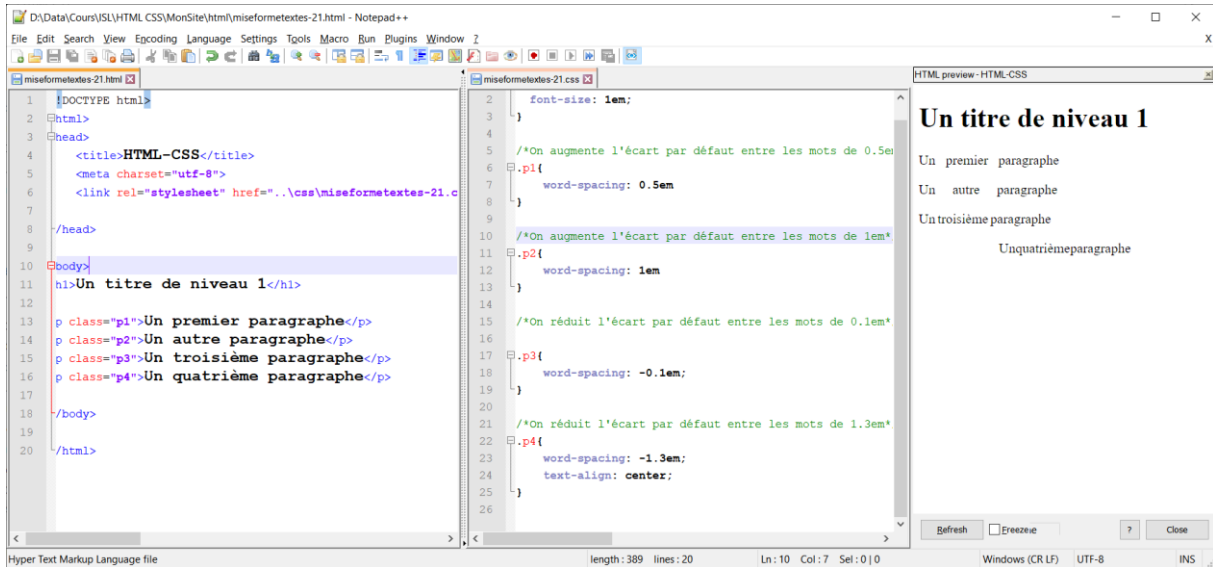


La propriété word-spacing

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.



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>HTML-CSS</title>
5 <meta charset="utf-8">
6 <link rel="stylesheet" href="..\css\miseformatextes-21.c
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>

```

```

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
21 /*On réduit l'écart par défaut entre les mots de 1.3em
22 .p4{
23   word-spacing: -1.3em;
24   text-align: center;
25 }
26

```

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

Un troisième paragraphe

Un quatrième paragraphe

length: 389 lines: 20 Ln: 10 Col: 7 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

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.

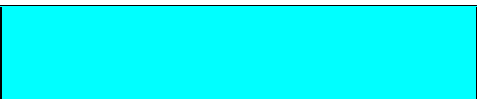

Les valeurs de type « nom de couleur »















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.

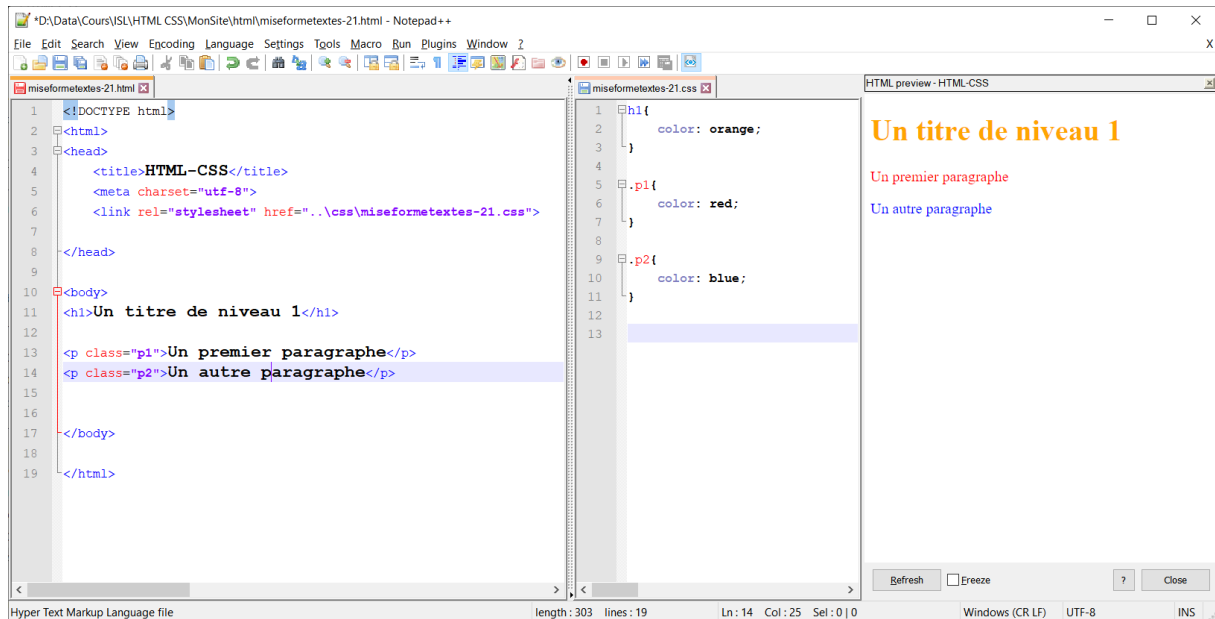
Ainsi, au départ, seules 16 couleurs ont été normalisées en CSS. C'était déjà un grand pas en avant pour les utilisateurs : ils n'avaient plus qu'à passer le nom (en anglais) de la couleur normalisée en valeur de la propriété CSS **color** afin de changer la couleur d'un élément.

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	Hexadécimal	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	#C0C0C0	
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 :



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.


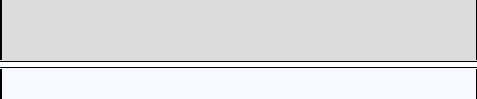
Ainsi, aujourd'hui, les navigateurs reconnaissent et supportent l'utilisation de plus de 140 noms de couleurs différents. Nous allons donc pouvoir passer chacune de ces valeurs à la propriété color pour définir une nouvelle couleur pour un texte.

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

Nom de la couleur	Hexadécimal	Couleur
AliceBlue	#F0F8FF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFD4	

Azure	#F0FFFF	
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	
DarkCyan	#008B8B	
DarkGoldenRod	#B8860B	
DarkGray	#A9A9A9	
DarkGrey	#A9A9A9	
DarkGreen	#006400	
DarkKhaki	#BDB76B	
DarkMagenta	#8B008B	
DarkOliveGreen	#556B2F	
DarkOrange	#FF8C00	
DarkOrchid	#9932CC	
DarkRed	#8B0000	
DarkSalmon	#E9967A	
DarkSeaGreen	#8FBC8F	
DarkSlateBlue	#483D8B	
DarkSlateGray	#2F4F4F	
DarkSlateGrey	#2F4F4F	

DarkTurquoise	#00CED1	
DarkViolet	#9400D3	
DeepPink	#FF1493	
DeepSkyBlue	#00BFFF	
DimGray	#696969	
DimGrey	#696969	
DodgerBlue	#1E90FF	
FireBrick	#B22222	
FloralWhite	#FFFAF0	
ForestGreen	#228B22	
Fuchsia	#FF00FF	
Gainsboro	#DCDCDC	
GhostWhite	#F8F8FF	
Gold	#FFD700	
GoldenRod	#DAA520	
Gray	#808080	
Grey	#808080	

Green	#008000	
GreenYellow	#ADFF2F	
HoneyDew	#F0FFF0	
HotPink	#FF69B4	
IndianRed	#CD5C5C	
Indigo	#4B0082	
Ivory	#FFFFF0	
Khaki	#F0E68C	
Lavender	#E6E6FA	
LavenderBlush	#FFF0F5	
LawnGreen	#7CFC00	
LemonChiffon	#FFFACD	
LightBlue	#ADD8E6	
LightCoral	#F08080	
LightCyan	#E0FFFF	
LightGoldenRodYellow	#FAFAD2	
LightGray	#D3D3D3	

LightGrey	#D3D3D3	
LightGreen	#90EE90	
LightPink	#FFB6C1	
LightSalmon	#FFA07A	
LightSeaGreen	#20B2AA	
LightSkyBlue	#87CEFA	
LightSlateGray	#778899	
LightSlateGrey	#778899	
LightSteelBlue	#B0C4DE	
LightYellow	#FFFFE0	
Lime	#00FF00	
LimeGreen	#32CD32	
Linen	#FAF0E6	
Magenta	#FF00FF	
Maroon	#800000	
MediumAquaMarine	#66CDAA	
MediumBlue	#0000CD	

MediumOrchid	#BA55D3	
MediumPurple	#9370DB	
MediumSeaGreen	#3CB371	
MediumSlateBlue	#7B68EE>	
MediumSpringGreen	#00FA9A	
MediumTurquoise	#48D1CC<	
MediumVioletRed	#C71585	
MidnightBlue	#191970	
MintCream	#F5FFFA	
MistyRose	#FFE4E1	
Moccasin	#FFE4B5	
NavajoWhite	#FFDEAD	
Navy	#000080	
OldLace	#FDF5E6	
Olive	#808000	
OliveDrab	#6B8E23	
Orange	#FFA500	

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	

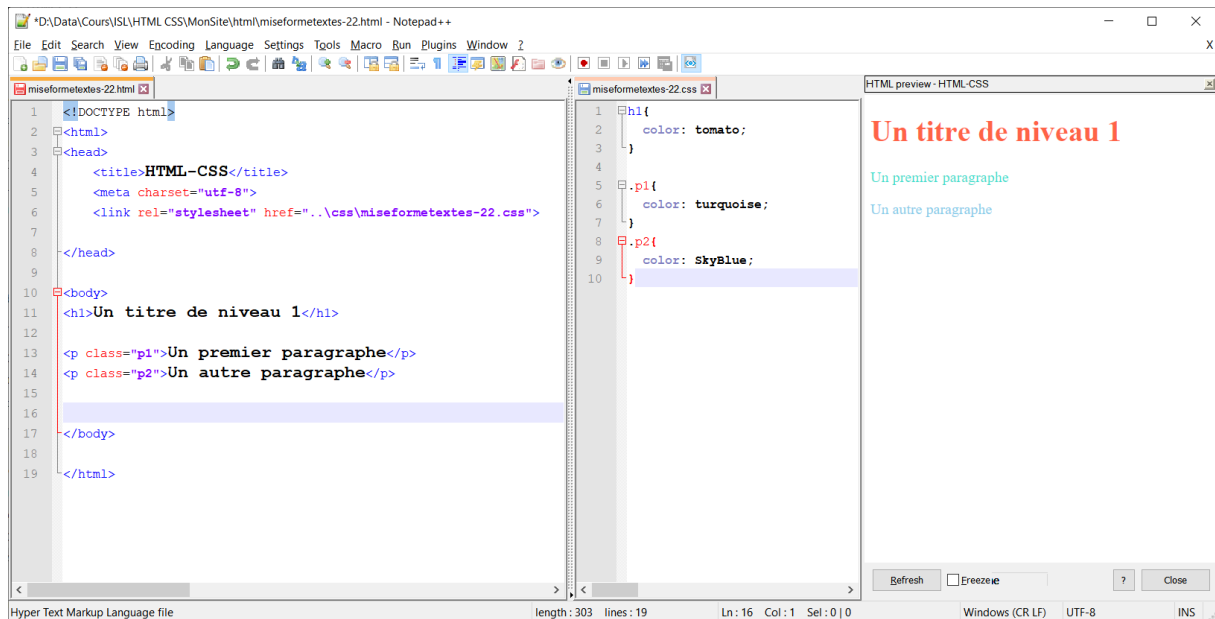
SaddleBrown	#8B4513	
Salmon	#FA8072	
SandyBrown	#F4A460	
SeaGreen	#2E8B57	
SeaShell	#FFF5EE	
Sienna	#A0522D	
Silver	#C0C0C0	
SkyBlue	#87CEEB	
SlateBlue	#6A5ACD	
SlateGray	#708090	
SlateGrey	#708090	
Snow	#FFFAFA	
SpringGreen	#00FF7F	
SteelBlue	#4682B4	
Tan	#D2B48C	
Teal	#008080	
Thistle	#D8BFD8	

Tomato	#FF6347	
Turquoise	#40E0D0	
Violet	#EE82EE	
Wheat	#F5DEB3	
White	#FFFFFF	
WhiteSmoke	#F5F5F5	
Yellow	#FFFF00	
YellowGreen	#9ACD32	

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 :



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 voudrions 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.

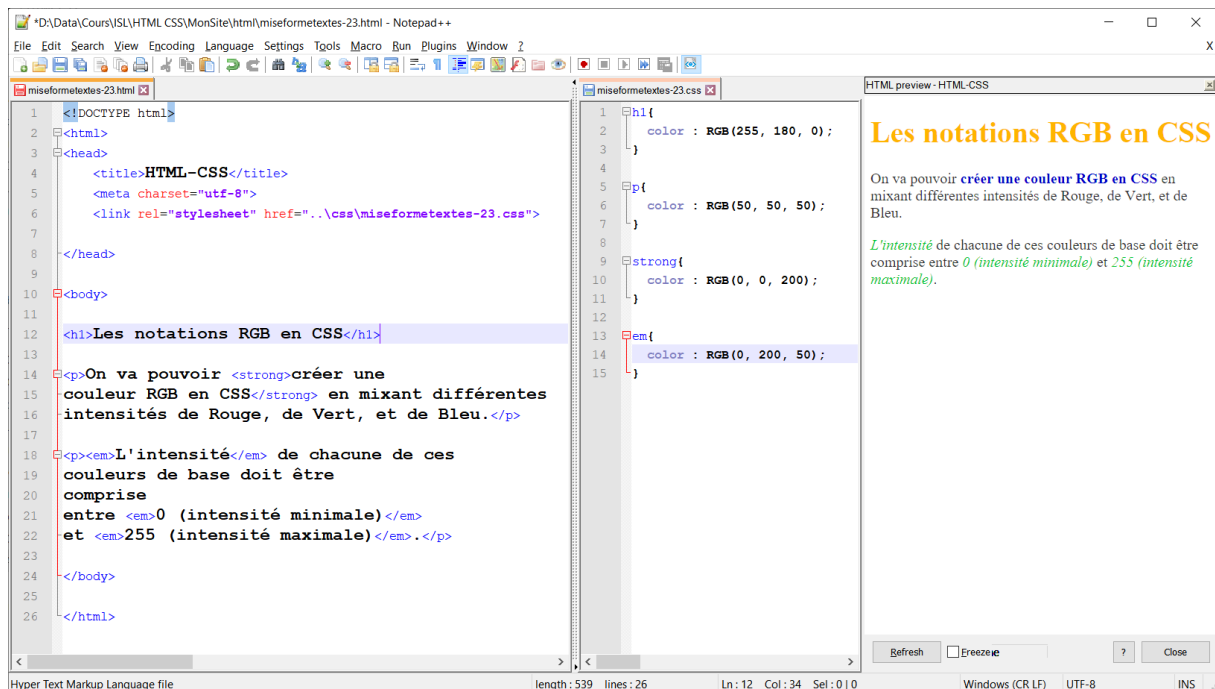
En précisant une intensité de 255 de rouge, en revanche, on indique qu'on souhaite utiliser beaucoup de rouge pour créer notre couleur finale.

Vous vous rappelez à l'école quand vous mélangiez 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.



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, reprenez 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 hexadécimale 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 seizaines et non pas des dizaines. Ainsi, pour représenter le « 16 » en hexadécimal, nous utiliserons le symbole « 10 » (comprendre 1 seizaine + 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

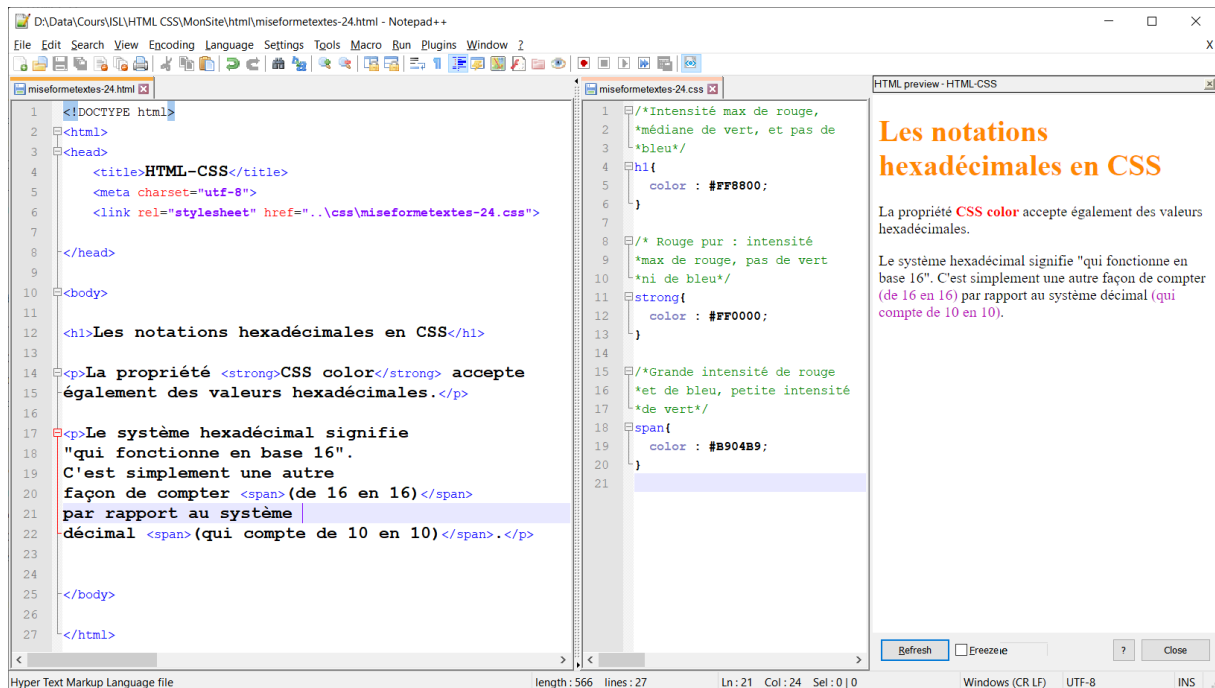
fait très simple. Faites l'effort de comprendre cela pour bien comprendre l'utilisation des valeurs hexadécimales avec les couleurs en CSS !

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 devons faire précéder cette notation par un dièse (symbole #).

Prenons immédiatement quelques exemples pour illustrer cela :



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>HTML-CSS</title>
5 <meta charset="utf-8">
6 <link rel="stylesheet" href="..\css\miseformatextes-24.css">
7 </head>
8 <body>
9
10 <h1>Les notations hexadécimales en CSS</h1>
11
12 <p>La propriété <strong>CSS color</strong> accepte également des valeurs hexadécimales.</p>
13
14 <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>
15
16 </body>
17 </html>

```

```

1 /*Intensité max de rouge,
2 *médiane de vert, et pas de
3 *bleu*/
4 h1{
5   color : #FF8800;
6 }
7
8 /* Rouge pur : intensité
9 *max de rouge, pas de vert
10 *ni de bleu*/
11 strong{
12   color : #FF0000;
13 }
14
15 /*Grande intensité de rouge
16 *et de bleu, petite intensité
17 *de vert*/
18 span{
19   color : #B904B9;
20 }
21

```

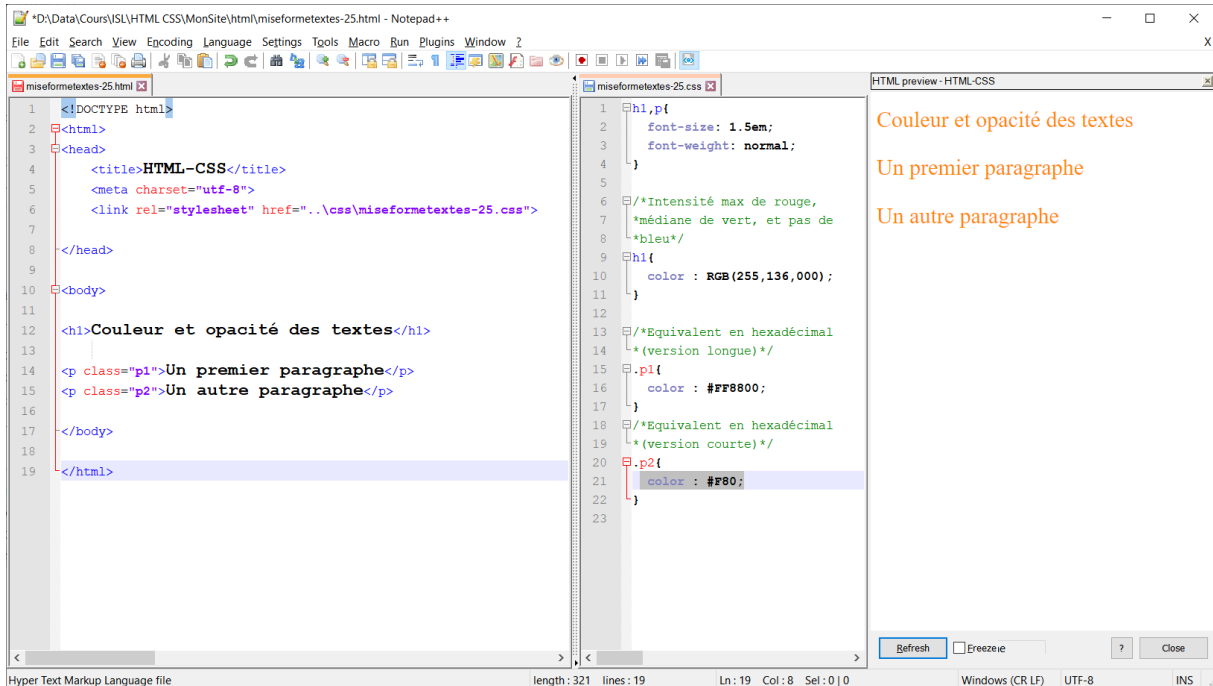
Les notations hexadécimales en CSS

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).

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.



```

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

```

```

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 /*Equivalent en hexadécimal
14 *(version longue)*/
15 .p1{
16  color : #FF8800;
17 }
18 /*Equivalent en hexadécimal
19 *(version courte)*/
20 .p2{
21  color : #F80;
22 }
23

```

Couleur et opacité des textes

Un premier paragraphe

Un autre paragraphe

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.

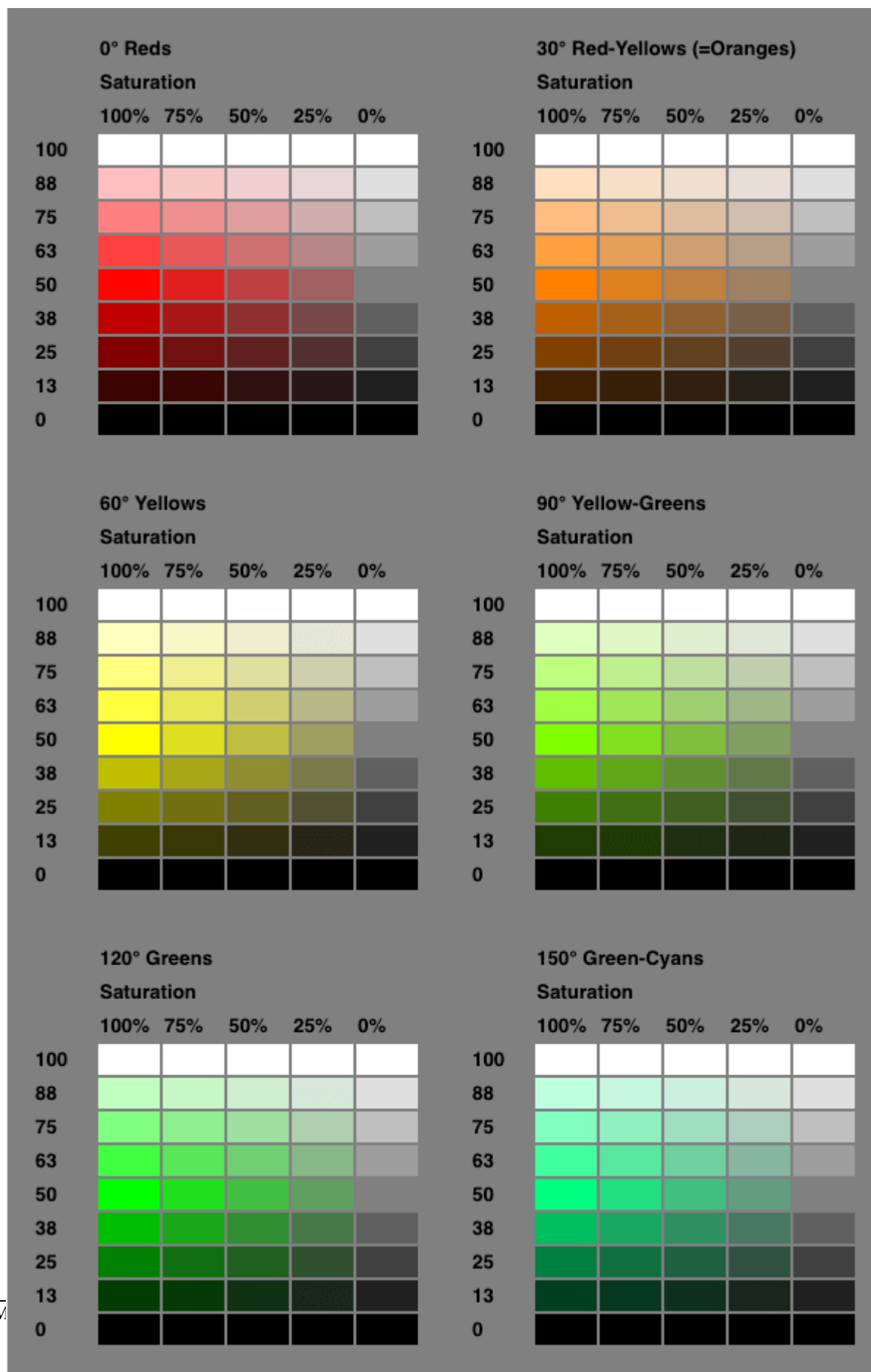
Les notations de type HSL

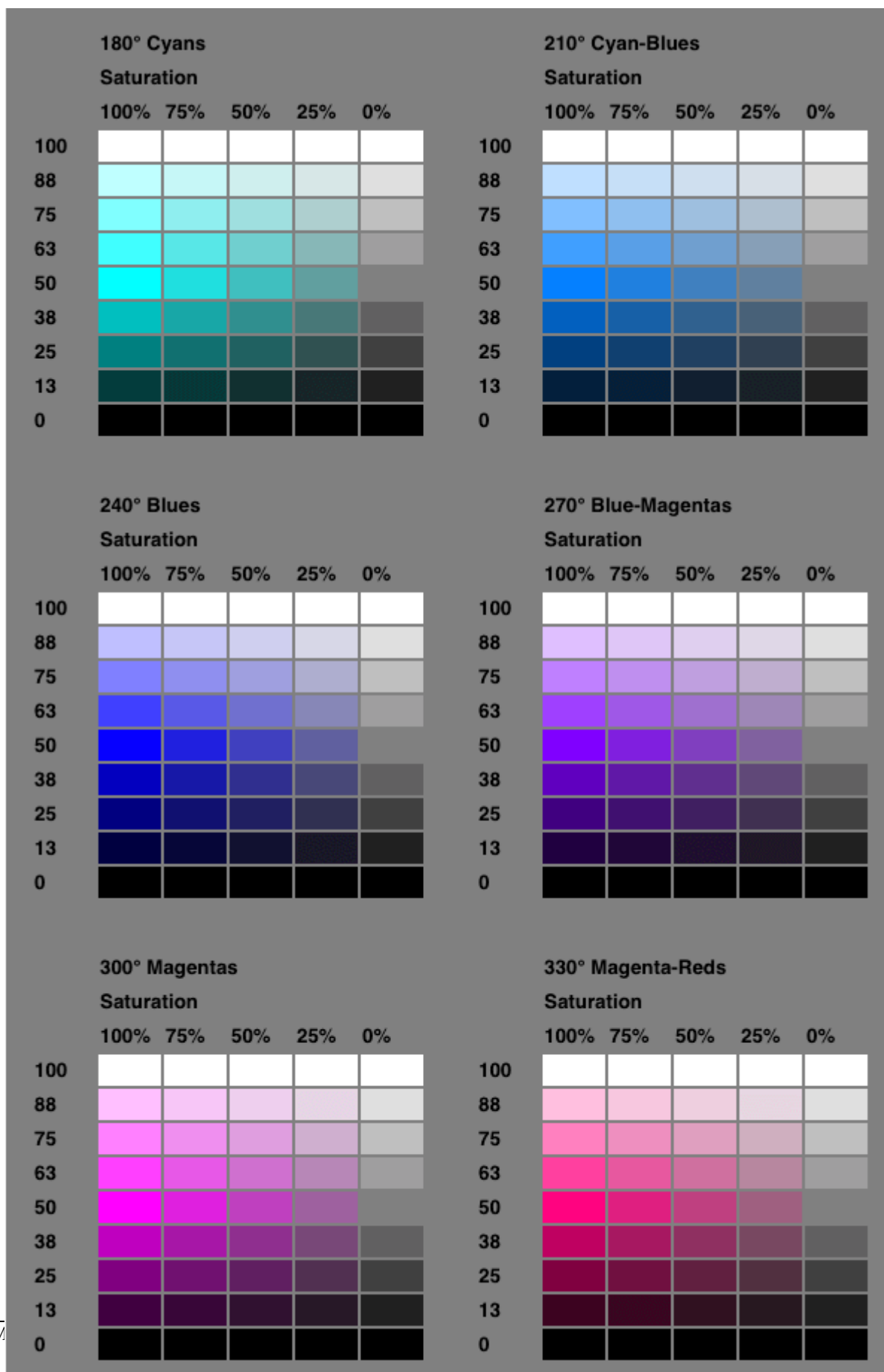
« HSL » est l'abréviation de « **Hue-Saturation-Lightning** », c'est-à-dire « teinte-saturation-luminosité » en français.

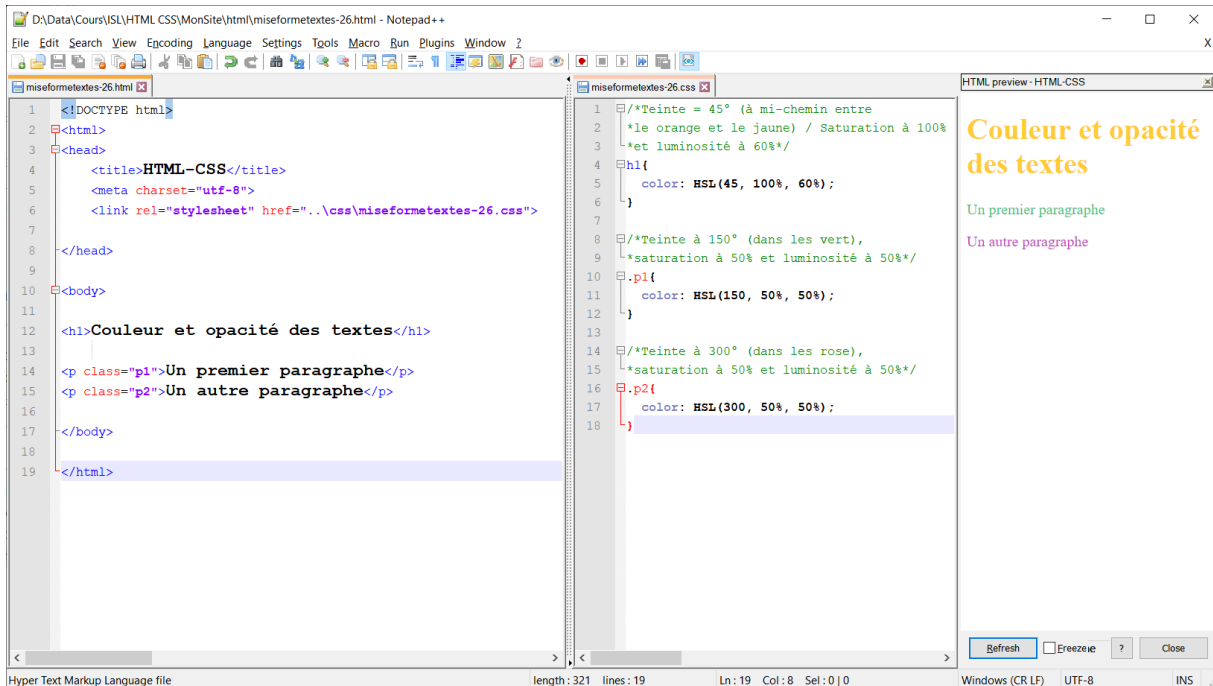
Pour créer une couleur en utilisant les notations HSL, nous allons devoir renseigner trois valeurs :

- 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 :







```

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 </head>
8 <body>
9 <h1>Couleur et opacité des textes</h1>
10 <p class="p1">Un premier paragraphe</p>
11 <p class="p2">Un autre paragraphe</p>
12 </body>
13 </html>

```

```

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 }

```

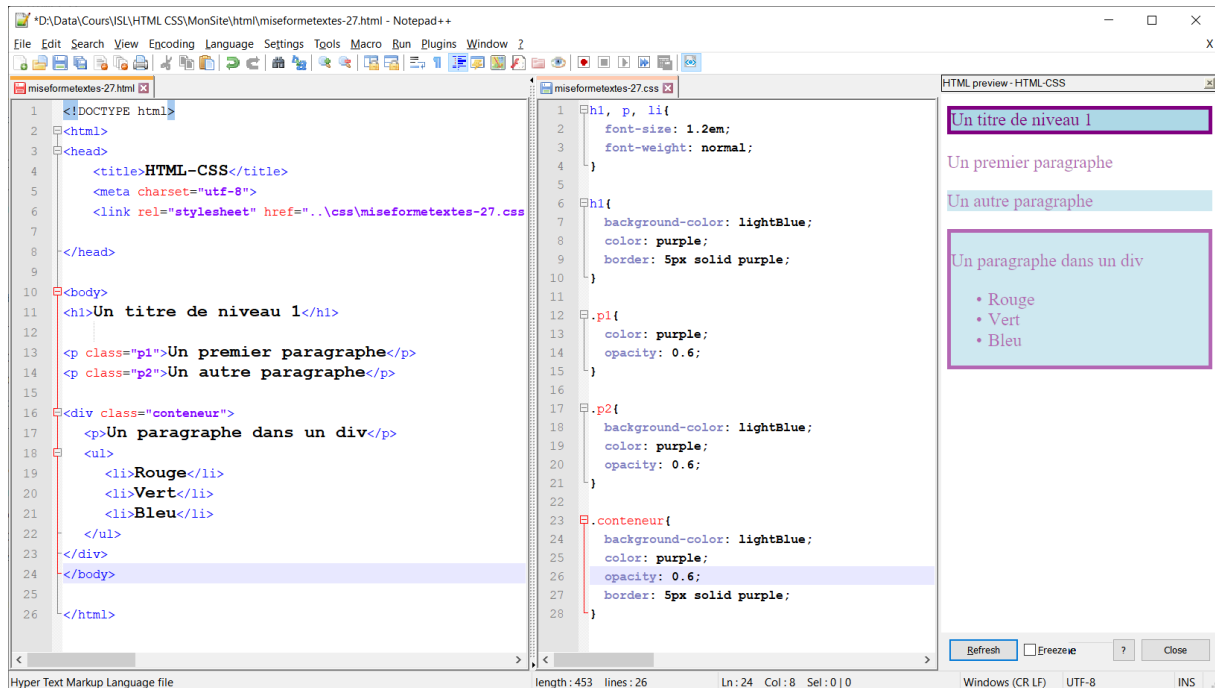
L'opacité des éléments et des textes en CSS

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.



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).

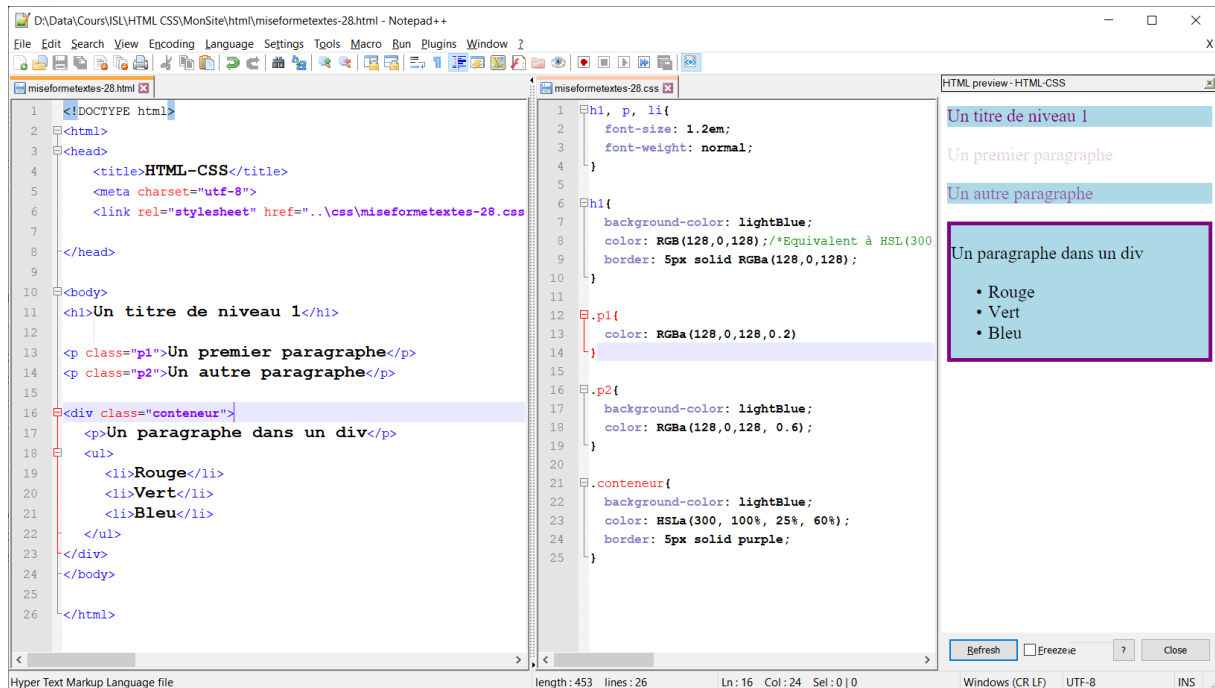
Ce comportement va parfois être le comportement voulu mais ce n'est pas celui attendu dans le cas où l'on souhaite simplement gérer l'opacité des textes (et uniquement des textes) de nos éléments.

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).



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.



INSTITUT SAINT-LAURENT

ENSEIGNEMENT DE PROMOTION SOCIALE

Baccalauréat en informatique
