

HTML5



CSS Tutorial

MISE EN FORME DES TEXTES

CSS Tutorial

FONT-FAMILY ET LES GOOGLE FONTS

La propriété CSS font-family

- La propriété CSS font-family permet de définir la police du texte.
- Il faut savoir que tous les navigateurs ne supportent pas tous les mêmes polices.
- Pour cette raison, il faut toujours indiquer plusieurs noms de police à utiliser en valeur de la propriété font-family, en commençant par celle souhaitée, 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.
- Si un nom de police contient des espaces, il faut le mettre entre guillemets ou apostrophes.

La propriété CSS font-family

- Voici un premier exemple d'utilisation de **font-family**. On va définir une même police pour toute la page HTML en appliquant la propriété directement à l'élément body.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Propriétés de type font</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Les propriétés de type font</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

La propriété CSS font-family

- Ici on indique trois polices d'écriture différentes en valeur de la propriété font-family : **Source code pro**, **Verdana** et **sans-serif**.
- Les deux dernières polices indiquées sont un peu particulières : **Verdana** fait partie des « web safe fonts » (en français "famille de polices sûres") tandis que **sans-serif** est ce qu'on appelle une « famille générique ».



- Pour tester le code : <http://jsbin.com/sipujup/22/edit?html,css,output>

Web safe fonts et familles génériques

- Les **web safe fonts** sont un ensemble de polices qui sont lues par tous les navigateurs quels qu'ils soient.
- Les **familles génériques** regroupent un ensemble de polices ayant une allure similaire, comme les polices de la famille **serif** par exemple.
- En général, on indiquera toujours au moins une police appartenant aux web safe fonts en valeur de la propriété **font-family** ainsi qu'une famille générique associée afin d'être certain qu'au moins une valeur puisse être lue par le navigateur.

Web safe fonts et familles génériques

- Voici une liste des web safe fonts les plus utilisées avec leur famille générique associée :

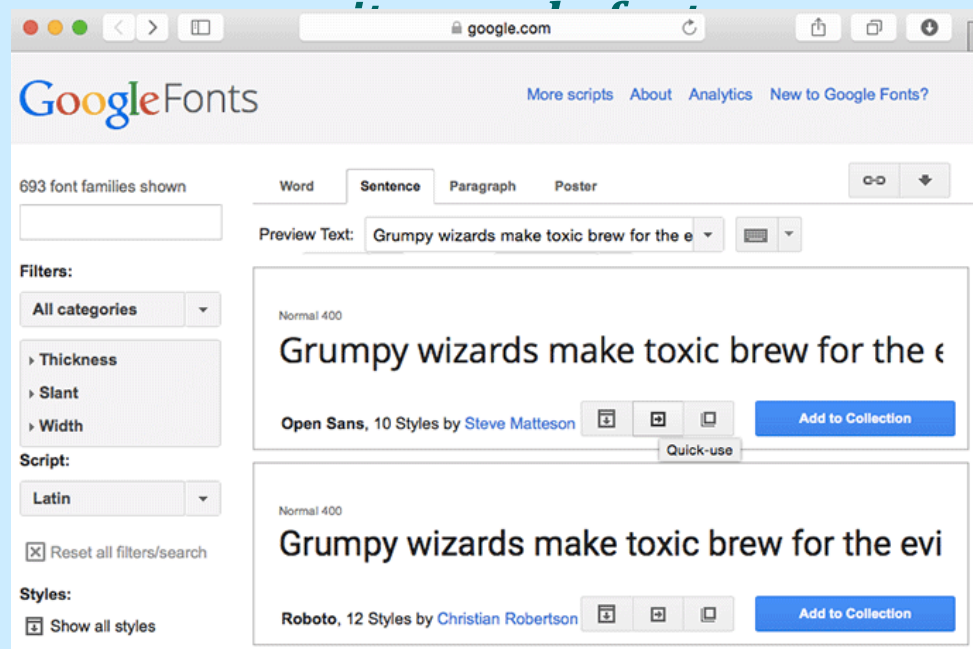
Famille générique	Police
Serif	Times New Roman, Georgia
Sans-serif	Arial, Verdana
Monospace	Courier New, Lucida Console
Cursive	Comic sans MS

Intégration 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.
- Le grand intérêt des Google Fonts est qu'elles ne dépendent pas d'un navigateur mais peuvent être lues par (quasiment) tous les navigateurs.
- L'autre grand avantage est que vous pouvez faire votre choix parmi une très longue liste de polices et ainsi totalement personnaliser l'écriture sur votre site.

Intégration des Google Fonts

- 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://www.google.com/fonts>
- Ici, on va choisir la police « Open sans ». Cliquez sur le bouton « quick use » pour obtenir le lien de la police.



Intégration des Google Fonts

- Ensuite, il faut insérer le lien dans une page HTML avec l'élément link et il ne restera alors plus qu'à utiliser la police en CSS :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Propriétés de type font</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
    <link rel="stylesheet" href="http://fonts.googleapis.com/css?family=Open+Sans">
  </head>

  <body>
    <h1>Les propriétés de type font</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

Intégration des Google Fonts

- Ensuite, il faut insérer le lien dans une page HTML avec l'élément link et il ne restera alors plus qu'à utiliser la police en CSS :

```
body{  
  font-family: "Open sans", Georgia, serif;  
}
```



- Pour tester le code : <http://jsbin.com/gopega/48/edit?html,css,output>

CSS Tutorial

LES PROPRIETES CSS DE TYPE FONT

Les propriétés de type font

- Les propriétés CSS de type **font-** permettent de modifier l'apparence de notre police d'écriture, et donc des textes.
- On peut par exemple pouvoir transformer la taille, le poids ou le style de la police d'écriture.
- Dans cette partie, nous allons voir les propriétés de type **font-** les plus utilisées :
 - font-size pour modifier la taille des textes ;
 - font-style pour modifier le style des textes ;
 - font-weight pour modifier le poids des textes.

La propriété CSS font-size

- La propriété CSS **font-size** permet de modifier la taille de la police d'écriture. Elle accepte deux grands types de valeurs : des tailles de polices absolues ou fixes et des tailles relatives ou variables.
- Les valeurs de type absolu sont généralement exprimées en px (pixels) ou éventuellement en pt (points). Une taille absolue ne bougera jamais.
- Les valeurs de type relatif vont être exprimées soit en % (pourcentage), en em ou en ex. L'intérêt des valeurs relatives est qu'elles vont s'adapter relativement à certains paramètres : soit par rapport aux préférences enregistrées dans le navigateur par un visiteur, soit par rapport à la taille d'un élément parent.
- (cela facilite la lecture pour les personnes à faible vision)

La propriété CSS font-size

- Les tailles relatives vont cependant être moins faciles à gérer dans la mise en forme globale d'une page (car si on ne prend pas de précautions, un texte peut dépasser d'une partie de la page par exemple).
- La propriété **font-size** peut également prendre la valeur « **inherit** », ce qui veut dire que l'élément ciblé héritera de la valeur de son parent.
- Cette valeur fonctionne avec beaucoup de propriétés CSS et peut être utile pour annuler un comportement par défaut ou en cas de conflit de styles.

La propriété CSS font-size

- Voici quelques exemples utilisant **font-size** et ses différents types de valeurs .
- Attention : lorsque l'on code, on utilise les notations anglo-saxonnes. Penser donc à utiliser un point pour les chiffres décimaux à la place de la virgule.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Propriétés de type font</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Les propriétés de type font</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

La propriété CSS font-size

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/45/edit?html,css,output>

La propriété CSS font-style

- La propriété CSS **font-style** permet de forcer le style de notre police, par exemple pour mettre des textes en italique.
- La propriété font-style accepte quatre valeurs différentes :
 - normal (par défaut) ;
 - italic (italique) ;
 - oblique (penché) ;
 - inherit (l'élément ciblé hérite du style de son élément parent).
- Les valeurs italic et oblique produisent souvent un résultat très similaire.

La propriété CSS font-style

- Les valeurs italic et oblique produisent souvent un résultat très similaire.
- Il faut savoir cependant que toutes les polices ne supportent pas l'état italique (certaines n'ont pas été conçues pour pouvoir être mises en italique). Dans ces cas là, la valeur oblique va forcer l'inclinaison de la police en question.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Propriétés de type font</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Les propriétés de type font</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

La propriété CSS font-style

- (suite)

```
/*Le texte de notre paragraphe p1 s'affiche en italique*/  
.p1{  
    font-style: italic;  
}  
  
/*Le texte de notre paragraphe p1 s'affiche penché*/  
.p2{  
    font-style: oblique;  
}
```



- Pour tester le code : <http://jsbin.com/gopega/46/edit?html,css,output>

La propriété CSS font-weight

- La propriété CSS **font-weight** permet de définir le poids d'une police, c'est-à-dire son épaisseur.
- Cette propriété peut prendre différentes valeurs :
 - normal (valeur par défaut) ;
 - lighter (la police sera plus fine) ;
 - bold (la police sera plus épaisse) ;
 - bolder (la police sera très épaisse) ;
 - une centaine entre 100 (police très fine) et 900 (police très épaisse). 400 correspond à la valeur « normal » et 700 à « bold » ;
 - inherit (l'élément hérite du style de son parent) ;
 - initial (définit la propriété sur sa valeur d'origine).

La propriété CSS font-weight

- De nouveau, il faut faire attention au fait que certaines polices ne supportent pas certains poids et donc ne supportent pas certaines valeurs.

```
/*Le texte de notre paragraphe p1 est épais (gras)*/  
.p1{  
  font-weight: bold;  
}  
  
/*Le texte de notre paragraphe p2 est normal*/  
.p2{  
  font-weight: 400;  
}
```



- Pour tester le code : <http://jsbin.com/gopega/47/edit?html,css,output>

CSS Tutorial

COULEUR ET OPACITE D'UN TEXTE EN CSS

La couleur en CSS

Soit à changer la couleur d'un texte avec la propriété `color` (vue précédemment dans ce cours).

- Jusqu'à présent, nous n'avons donné que des noms de couleurs (en anglais) comme valeurs à cette propriété. Cependant, `color` accepte différents types de valeurs :
 - Un nom de couleur (`red`, `blue`, etc.) ;
 - Une valeur hexadécimale (`#AA8811`, etc.) ;
 - Une valeur RGB ou RGBa.

Les noms de couleurs

- La façon la plus simple de changer la couleur d'un texte est de donner le nom d'une couleur (en anglais) en valeur de la propriété color.
- Cependant, dans ce cas, le choix de couleurs est très limité.
- En effet, bien que les versions récentes des grands navigateurs supportent jusqu'à 160 couleurs et nuances différentes (ce qui est en fait assez peu), il n'y a que 16 qui soient réellement, universellement supportées.

Les noms de couleurs

- Pour information, les 16 noms de couleurs avec leur équivalent en notation hexadécimale (voir ci-après).

Aqua #00FFFF	Black #000000	Blue #0000FF	Fuschia #FF00FF
Gray #808080	Green #008000	Lime #00FF00	Maroon #800000
Navy #000080	Olive #808000	Purple #800080	Red #FF0000
Silver #C0C0C0	Teal #008080	White #FFFFFF	Yellow #FFFF00

Les noms de couleurs

- Il est très simple de modifier la couleur d'un texte en indiquant le nom d'une couleur :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Couleur et opacité</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Couleur et opacité des textes</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

Les noms de couleurs

- (suite)



- Pour tester le code : <http://jsbin.com/gopega/49/edit?html,css,output>

Les valeurs hexadécimales

- Pour ceux qui ne sont pas familier de la numération informatique, rappelons que le mot « hexadécimal » signifie « qui fonctionne en base 16 ».
- Le système hexadécimal compte non pas par dizaines mais par seizaines. Les « unités » de 10 à 15 sont représentées par les lettres A, B, C, D, E et F.
- Par exemple, l'équivalent de « 12 » en base hexadécimale est « 0C ». Ci-joint un tableau récapitulatif :

[TableauHex.pdf](#)

Les valeurs hexadécimales

- Le grand intérêt des valeurs hexadécimales est qu'elles permettent de choisir parmi 16 millions de nuances de couleurs .
- Pour construire une valeur hexadécimale pour la propriété color, on doit tout d'abord commencer par un dièse (#) , puis indiquer trois suites de valeurs représentant respectivement les intensités de rouge, de vert et de bleu dont va être composée votre couleur.
- Chacune de ces trois intensités va devoir être comprise entre 0 et 255 en base 10, c'est-à-dire entre 00 et FF en notation hexadécimale.

Les valeurs hexadécimales

- Voici un exemple. La notation correspond à la valeur hexadécimale de chaque couleur.



Les valeurs hexadécimales

- La première couleur est un rouge pur. Comme on peut le voir, la notation hexadécimale qui a permis de la créer contient une intensité maximale de rouge (les deux premiers chiffres : FF), une intensité minimale de vert (les deux chiffres du milieu : 00) et une intensité également minimale de bleu (les deux derniers chiffres : 00).
- La deuxième couleur est un jaune pur. On l'obtient en mixant une intensité maximale de rouge (les deux premiers chiffres : FF), une intensité maximale de vert (les deux chiffres du milieu : FF) et une intensité minimale de bleu (les deux derniers chiffres : 00).
- De même, on obtient les couleurs « lime » (vert pur), « aqua » (vert-bleu) et « blue » (bleu pur) en réglant différemment les intensités de chacune de nos trois couleurs de base.

Les valeurs hexadécimales

- Noter que la couleur noire correspond à une intensité minimale de rouge, vert et bleu (donc #000000) tandis que la couleur blanche correspond à une intensité maximale de rouge, vert et bleu (donc #FFFFFF).
- Noter également que lorsque chacune de vos intensités possède le même chiffre des dizaines et des unités, vous pouvez ne préciser qu'un chiffre. Par exemple, écrire #FFFFFF ou plus simplement #FFF revient au même, tout comme écrire #AA8800 et #A80.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Couleur et opacité</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Couleur et opacité des textes</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

Les valeurs hexadécimales

- (suite)

```
/*Une nuance de orange (#F80 = #FF8800)*/  
h1{  
    color: #F80;  
}  
  
/*Une sorte de violet ?*/  
.p1{  
    color: #7A01F8;  
}  
  
/*Couleur "teal"*/  
.p2{  
    color: #008080;  
}
```



- Pour tester le code : <http://jsbin.com/gopega/50/edit?html,css,output>

Les valeurs RGB

RGB signifie « Red Green Blue » (rouge, vert, bleu).

- Ce type de valeur utilise le même système de nuances que les valeurs hexadécimales, à part que cette fois-ci il est possible de compter «normalement ».
- Pour créer une couleur RGB, on indique trois intensités de rouge, de vert et de bleu qui vont être mixées pour créer une couleur.
- La différence est que cette fois-ci on indique bien l'intensité entre 0 et 255, et non pas entre 00 et FF.

Les valeurs RGB

- Voici comment on crée des couleurs en utilisant les valeurs de type RGB :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Couleur et opacité</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Couleur et opacité des textes</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

Les valeurs RGB

- Le grand intérêt des valeurs de type RGB, en plus de la création possible de plus de 16 millions de couleurs différentes, est que cette notation permet également de gérer l'opacité de nos textes.

```
/*Une nuance de orange*/  
h1{  
  color: RGB(255,120,000);  
}  
  
/*Du rouge pur*/  
.p1{  
  color: RGB(255,000,000);  
}  
  
/*Couleur "fuschia" (sorte de rose flash)*/  
.p2{  
  color: RGB(255,000,255);  
}
```



- Pour tester le code : <http://jsbin.com/gopega/51/edit?html,css,output>

L'opacité des textes

Il y a deux manières de gérer l'opacité (le fait d'être plus ou moins transparent) des textes.

- La première est en utilisant la propriété CSS `opacity`. Cette propriété va prendre une valeur entre 0 (texte totalement transparent) et 1 (texte totalement opaque).
- La deuxième façon est d'indiquer une quatrième valeur définissant l'opacité dans nos notations RGB. Dans ce cas là, la notation devient `RGBA`.
- Cette quatrième valeur, tout comme pour `opacity`, doit être comprise entre 0 (texte transparent) et 1 (texte opaque).

L'opacité des textes

- Voici en pratique comment tout cela fonctionne.

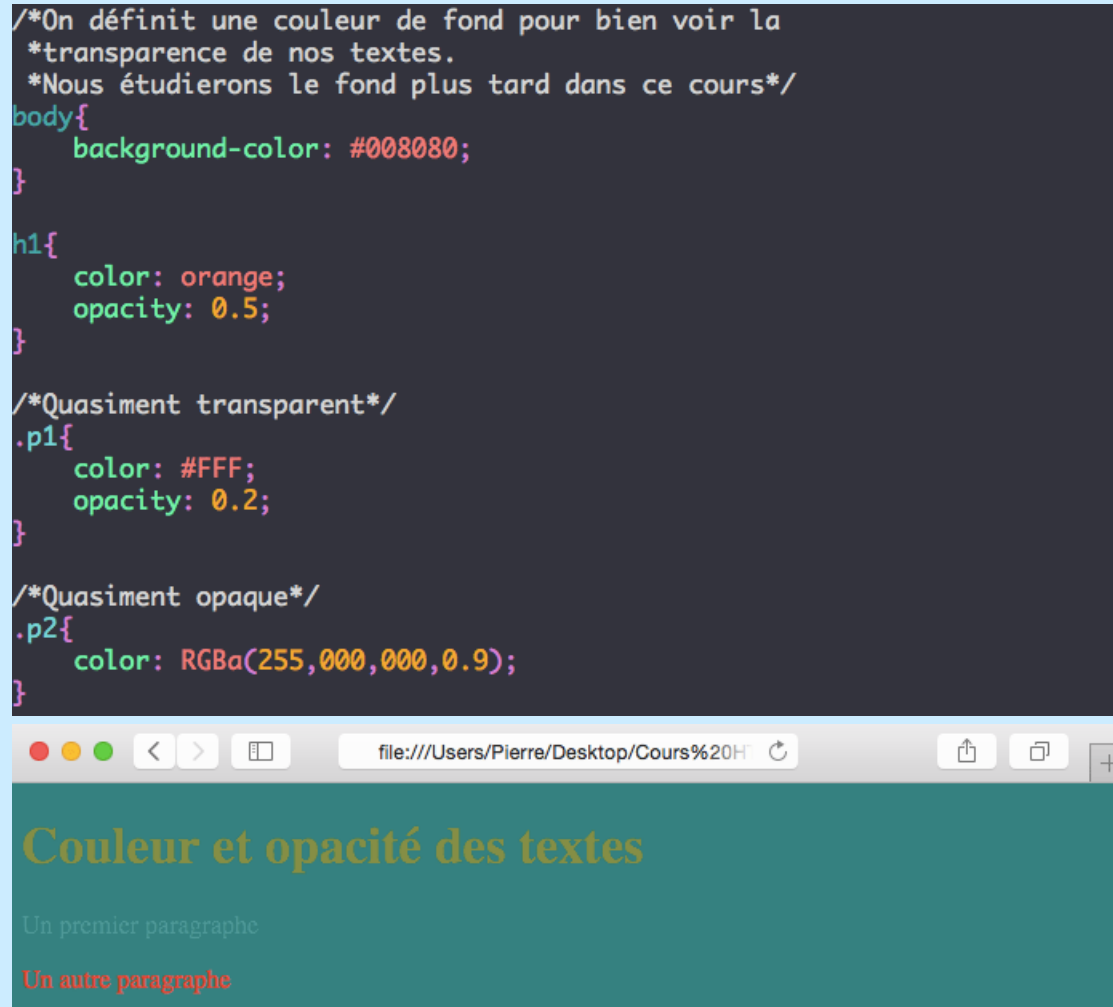
```
<!DOCTYPE html>
<html>
  <head>
    <title>Couleur et opacité</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Couleur et opacité des textes</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```


L'opacité des textes

- (suite)



- Pour tester le code : <http://jsbin.com/gopega/52/edit?html,css,output>

CSS Tutorial

LES PROPRIETES CSS DE TYPE TEXT

La propriété CSS text-align

- La propriété CSS **text-align** permet de définir l'alignement du texte.
- Choix est donné parmi cinq valeurs :
 - Left : Le texte sera aligné à gauche (par défaut) ;
 - Right : Le texte sera aligné à droite ;
 - Center : Le texte est centré ;
 - Justify : Le texte est justifié ;
 - Inherit : Hérite de la valeur de l'élément parent.
- Noter que le texte est aligné par rapport aux bords de son élément parent par défaut.

La propriété CSS text-align

- Exemples.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Propriétés "text-"</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Propriétés CSS "text-"</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

La propriété CSS text-align

- (suite)

```
h1{  
  text-align: center;  
}  
  
.p1{  
  text-align: left;  
}  
  
.p2{  
  text-align: right;  
}
```



- Pour tester le code : <http://jsbin.com/gopega/54/edit?html,css,output>

La propriété CSS text-transform

- La propriété CSS text-transform permet de transformer un texte ou une partie d'un texte en majuscules ou en minuscules.
- Choix est donné parmi cinq valeurs :
 - Lowercase : Met tout le texte en minuscules ;
 - Uppercase : Met tout le texte en majuscules ;
 - Capitalize : Met la première lettre de chaque mot en majuscule ;
 - Inherit : Hérite de la valeur de l'élément parent ;
 - None : Pas de transformation du texte. Utile pour annuler une transformation par défaut donnée par héritage.

La propriété CSS text-transform

- Exemple :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Propriétés "text-"</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Propriétés CSS "text-"</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

La propriété CSS text-transform

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/55/edit?html,css,output>

La propriété CSS text-decoration

- La propriété CSS **text-decoration** permet d'ajouter ou d'enlever des décorations aux textes.
- Cette propriété accepte six valeurs différentes :
 - Underline : Le texte sera souligné ;
 - Overline : Une ligne apparaît au dessus du texte ;
 - Line-through : Le texte sera barré ;
 - Inherit : Hérite de la valeur de l'élément parent ;
 - Initial : Utilise la valeur par défaut de la propriété ;
 - None : Pas de décoration.
- Cette propriété va par exemple permettre d'enlever le soulignement automatique qui apparaît dès que l'on crée un lien.

La propriété CSS text-decoration

- Exemple :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Propriétés "text-"</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Propriétés CSS "text-"</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
    <p>Un <a href="http://wikipedia.org">lien</a></p>
  </body>
</html>
```

La propriété CSS text-decoration

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/56/edit?html,css,output>

La propriété CSS text-indent

- La propriété CSS **text-indent** permet de gérer l'indentation d'un texte.
- La propriété text-indent accepte aussi bien des valeurs absolues (en px par exemple) que des valeurs relatives (en % ou em par exemple).
- On peut également lui attribuer des valeurs négatives afin de décaler un texte vers la gauche.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Propriétés "text-"</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Propriétés CSS "text-"</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

La propriété CSS text-indent

- (Suite)

```
h1{  
  text-indent: -50px;  
}  
  
.p1{  
  text-indent: 20px;  
}  
  
.p2{  
  text-indent: 20%;  
}
```



- Pour tester le code : <http://jsbin.com/gopega/57/edit?html,css,output>

La propriété CSS text-shadow

- La propriété CSS **text-shadow** permet de créer des ombres autour de nos textes, afin que ceux-ci se détachent de l'arrière plan.
- On va devoir indiquer quatre valeurs dans un ordre précis à la propriété text-shadow afin que celle-ci fonctionne correctement :
 - La projection horizontale de l'ombre (en px) ;
 - La projection verticale de l'ombre (en px) ;
 - Le rayon de propagation de l'ombre (le « radius », en px) ;
 - La couleur de l'ombre. Accepte les mêmes valeurs que la propriété color.
- Si l'on indique des valeurs positives pour les projections horizontales et verticales de l'ombre, l'ombre sera projetée en bas à droite du texte. En indiquant des valeurs négatives, on peut modifier l'orientation de l'ombre.

La propriété CSS text-shadow

- Exemple :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Propriétés "text-"</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Propriétés CSS "text-"</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

La propriété CSS text-shadow

- (Suite)

```
/*Ombre gris-foncé en bas à droite*/  
h1{  
  text-shadow: 1px 1px 4px #444;  
}  
  
/*Ombre bleue en bas à gauche*/  
.p1{  
  text-shadow: -1px 1px 1px blue;  
}  
  
/*Ombre rouge centrée sur le texte*/  
.p2{  
  text-shadow: 0px 0px 2px RGB(200,000,025);  
}
```



- Pour tester le code : <http://jsbin.com/gopega/58/edit?html,css,output>

CSS Tutorial

GESTION DES ESPACES AU SEIN D'UN TEXTE EN CSS

La propriété CSS line-height

- La propriété CSS **line-height** permet de choisir la distance ou l'écartement entre deux lignes de texte.
- Cette propriété accepte des valeurs absolues ou relatives. Généralement, on fixera la valeur de line-height à 1,5 fois la taille du texte.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Les espaces</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

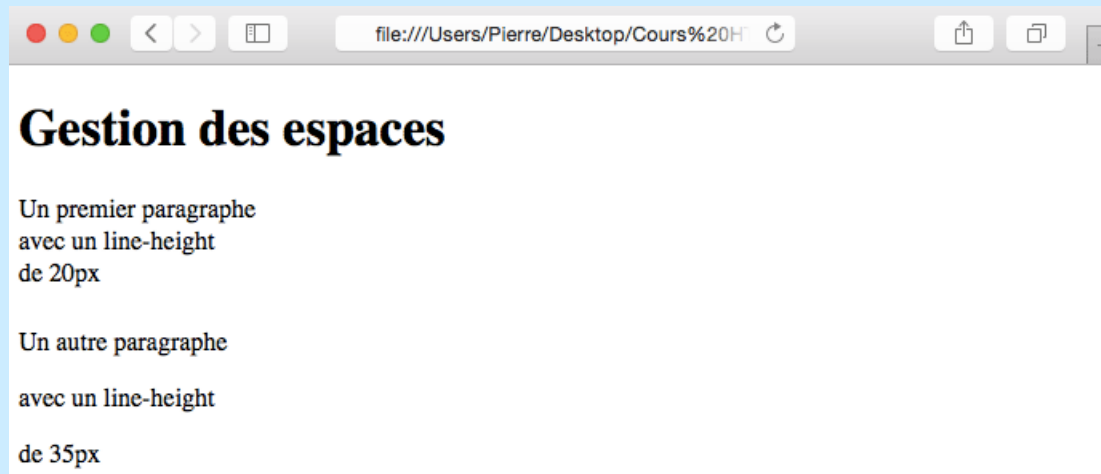
  <body>
    <h1>Gestion des espaces</h1>

    <p class="p1">Un premier paragraphe <br>avec un line-height<br>de 20px</p>
    <p class="p2">Un autre paragraphe <br>avec un line-height<br>de 35px</p>
  </body>
</html>
```

La propriété CSS line-height

- (Suite)

```
.p1{  
  line-height: 20px;  
}  
  
.p2{  
  line-height: 35px;  
}
```



- Pour tester le code : <http://jsbin.com/gopega/59/edit?html,css,output>

Les propriétés CSS letter-spacing et word-spacing

- Les propriétés CSS **letter-spacing** et **word-spacing** permettent respectivement de définir l'espace entre les lettres et entre les mots dans un texte.
- A noter que les valeurs des espaces précisées pour letter-spacing et word-spacing viendront s'ajouter à l'espace par défaut. Pour enlever une partie de l'espace ou même inverser les lettres ou les mots, on peut donner des valeurs négatives.
- Voici comment ces deux propriétés fonctionnent ensemble

```
<!DOCTYPE html>
<html>
  <head>
    <title>Les espaces</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Gestion des espaces</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un autre paragraphe</p>
  </body>
</html>
```

Les propriétés CSS letter-spacing et word-spacing

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/60/edit?html,css,output>

X

- X