

# Tutoriel CSS3

# Sommaire

1.	Le CSS qu'est-ce que c'est ? .....	4
2.	Comment fonctionne le CSS ? .....	4
3.	Comment inclure le CSS dans son site ? .....	4
4.	Comment remplir un fichier CSS ? .....	5
5.	Un arrière-plan uni .....	6
6.	Un arrière-plan uni, utiliser les codes couleurs .....	7
7.	Un arrière-plan avec une image .....	8
8.	Positionner une image .....	9
9.	Positionner une image – utiliser les pixels .....	9
10.	Une autre méthode de positionnement, « x » et « y » .....	10
11.	Mon arrière-plan se répète, comment faire ? .....	11
12.	Changer la couleur du texte .....	12
13.	Changer la police d'écriture .....	13
14.	Changer le style d'écriture .....	13
15.	Changer la forme de l'écriture .....	14
16.	Changer la taille du texte .....	14
17.	Modifier un certain type de texte .....	15
18.	Les bordures .....	17
18.1.	Changer le style des bordures .....	17
18.2.	Changer la taille d'une bordure .....	18
18.3.	Changer la couleur de la bordure .....	18
18.4.	Les bordures « one-sided » .....	19
19.	Modifier qu'une seule balise .....	20
19.1.	Les identifiants en HTML .....	20
19.2.	Les identifiants en CSS .....	21
19.3.	Modifier les balises, les identifiants existant .....	22
20.	Concevoir un site web .....	23
21.	La barre de navigation .....	25
21.1	Les barres de navigations horizontales .....	26
21.2	Les barres de navigations verticales .....	29
22.	Pour aller plus loin .....	30



## 1. Le CSS qu'est-ce que c'est ?

Avec le html, vous avez vu comment écrire dans un page, vous avez vu également comment ajouter un titre à votre page web.

Cependant, hormis certaines images que vous pouvez insérer avec la balise prévue à cet effet, il vous est impossible de modifier l'arrière-plan ou encore créer un « en-tête » ou un « pied de page ». Eh bien pas de panique ! Nous sommes là pour vous expliquer tout ça !

Tous les problèmes cités ci-dessus sont en réalité « réglés » par le CSS, ce langage web permet de mettre en forme votre site web. En effet, prenez n'importe quel site (disons Facebook) et essayez d'y enlever le CSS, vous verrez que ce n'est rien de plus qu'une simple page blanche avec du texte.

## 2. Comment fonctionne le CSS ?

Le CSS est un « complément » au HTML, vous pouvez, soit faire du CSS à l'intérieur de votre fichier HTML, soit utiliser un fichier à part. Pour éviter de se mélanger les pinceaux et pour faire la distinction entre les deux langages, nous allons utiliser un autre fichier.

## 3. Comment inclure le CSS dans son site ?

Vous l'aurez compris, si on utilise le CSS dans un fichier à part, il va falloir expliquer au HTML de prendre en compte ce CSS. Retournez dans votre fichier html, si vous avez bien suivi les challenges jusqu'au CSS, vous savez dès-lors que le fichier HTML est divisé en 2 grandes parties, la balise `<head>` contenant le titre (pour l'instant) et la balise `<body>` contenant votre site.

Vous souvenez-vous lorsque nous vous avons dit que la balise `<head>` était pour contenir le titre ? Eh bien, on vous a menti ! Mais ce mensonge était principalement fait pour faciliter votre compréhension. En réalité vous pouvez également appeler votre fichier CSS depuis cette balise, comme ceci :

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="mystyle.css">
    <title>Une simple page</title>
  </head>
  <body>
    <p>Un simple texte</p>
  </body>
</html>
```

Remplacer par le nom de  
votre fichier

## 4. Comment remplir un fichier CSS ?

Maintenant, nous avons créés et même inclus notre premier fichier CSS, mais il ne contient ... Rien ! Il est temps de le remplir. Ouvrez-le et remplissez-le comme suit :

```
body {  
  background-color: lightblue;  
}
```

On applique ce qu'il y a entre accolades à la balise body.

Cette ligne est ici pour dire que la couleur du background devient bleu clair.

Vous l'aurez compris, un fichier CSS ne se remplit pas de manière aléatoire, il y a une syntaxe derrière. Mais rassurez-vous elle est très facile !

1. Le nom de la balise (exemple : *body*, *head*, *p*, *b*, *h1*, *h2*, ...)
2. Une accolade ouvrante et une fermante (elle peut être devant la balise comme dans l'exemple ci-dessus ou en bas, faites comme vous le souhaitez).
3. Et à l'intérieur les propriétés que l'on souhaite appliquer à la balise.

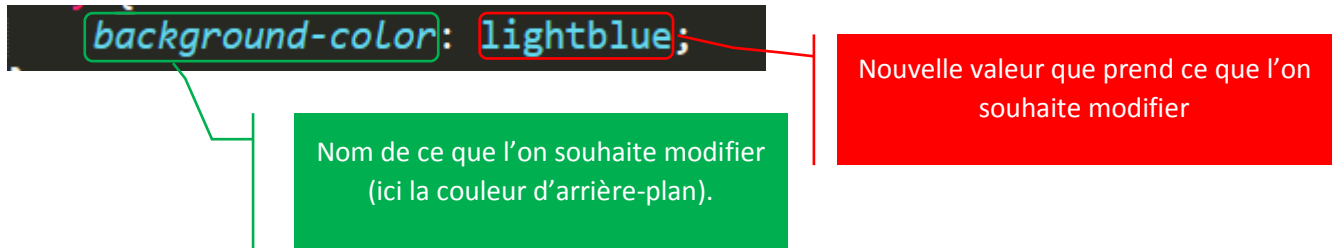
Si vous copiez-collez les lignes ci-dessus dans votre fichier CSS, vous devez avoir un fond bleu très clair sur toute la page (nous avons modifiés le *body*, donc c'est normal), comme ceci :



Ce n'est pas très joli, mais c'est un début ! Maintenant, essayons de mettre un fond plus esthétique !

## 5. Un arrière-plan uni

Nous allons rester dans le thème d'un arrière-plan coloré mais pour l'instant d'une seule-et-même couleur. Reprenons l'exemple ci-dessus, mais détaillons un peu plus ce qu'il y a entre les accolades.



J'ai oublié de mentionner ceci : Lorsque vous allez modifier par exemple le `background-color` de `<body>` il n'y aura que celui du `<body>` qui sera modifié, vous verrez que cette notion aura de l'importance plus tard, ne nous attardons pas trop sur ce sujet !

Maintenant, observons d'un peu plus près ce que fait cette ligne :

- Vous récupérez la « variable » `background-color` (en Français : Couleur d'arrière-plan)
- Vous attribuez la valeur `lightblue` (en Français : Bleu clair) à cette « variable »

Mais si le CSS avait gardé juste le `lightblue` comme valeur possible, eh bien, dites-vous que tous les sites web contiendrait uniquement un fond bleu clair. Pas très original ! Mais heureusement, le langage CSS est complet ! Vous pouvez mettre la couleur que vous souhaitez ! Absolument toutes les couleurs sont disponibles.

Par exemple, vous souhaitez mettre la couleur d'arrière-plan en noir ? Aucun problème :

```
body {  
    background-color: black;  
}
```



Vous l'aurez compris, vous pouvez mettre toutes les composantes « basiques », vous avez juste à changer la valeur ! Mais maintenant, une question se pose toujours, comment faire pour dire au CSS par exemple « donne-moi une couleur qui se situe entre le rouge et l'orange ». De cette manière, on ne peut pas ! Il faut faire différemment.

## 6. Un arrière-plan uni, utiliser les codes couleurs


Comme je l'ai dit ci-dessus, le CSS propose une autre manière de modifier la couleur de l'arrière-plan, il s'agit en fait d'utiliser ce que l'on appelle un « code couleur ». Et vous n'avez même pas besoin de calculer le code par vous-même, des superbes personnes se sont occupés de ça à votre place, par exemple sur le site [code-couleur.com](http://code-couleur.com) qui contient des codes couleurs comme ceci :

**CODE COULEUR**

Dans le langage de l'imprimerie, les couleurs se traduisent par des codes. Les premières formules se destinaient aux presses traditionnelles puis aux presses numériques. Il a donc fallu munir les couleurs de codes pour l'informatique. Apprenez maintenant la conversion html.



**CODE COULEUR ( HTML )**

**#D97A1B**

CODE RVB	CODE CMJN
ROUGE :217	TEINTE :30
VERT :122	SATURATION :88
BLEU :27	VALEUR :85

Si on souhaite maintenant modifier le code couleur dans le CSS, il suffit de remplacer la valeur de `background-color` par le code couleur, comme ceci par exemple :

```
body {  
    background-color: #D97A1B;  
}
```

Ça offre beaucoup de possibilités, vous ne trouvez pas ?

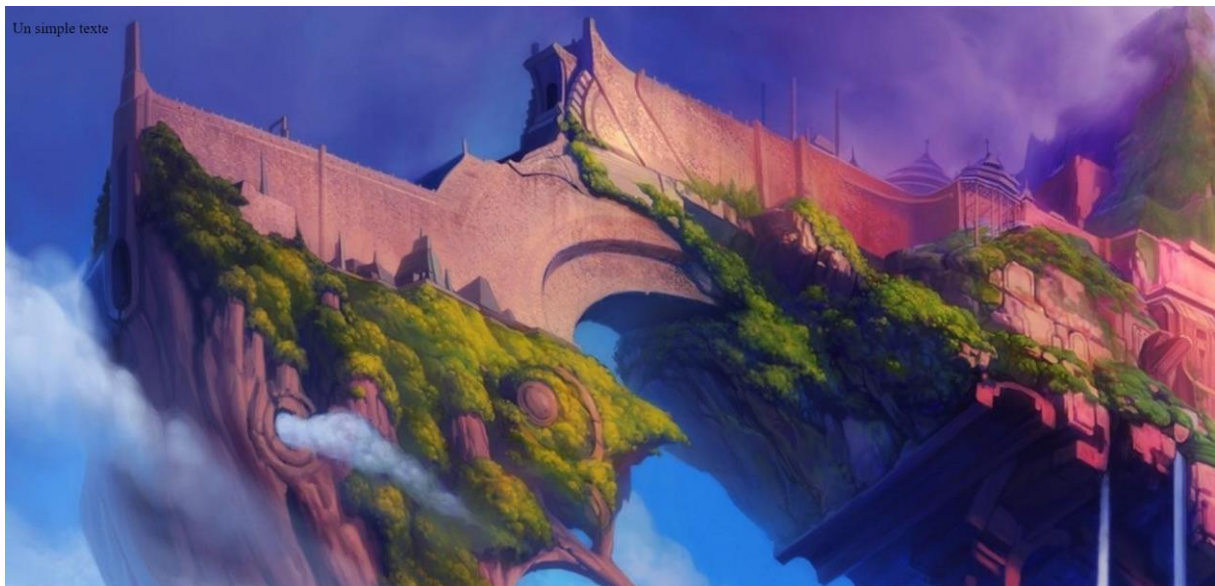
## 7. Un arrière-plan avec une image

Maintenant que vous avez vu comment modifier la couleur de l'arrière-plan, il est temps de mettre des images, et devinez quoi ? Ce n'est pas plus compliqué ! Il s'agit de modifier autre chose que le `background-color`. A vrai dire, ce que l'on va modifier à un nom assez explicite : `background-image`.

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

Respectez toujours la notation `url("nomdufichier")`. Il faut également toujours mettre le nom de l'image dans ces parenthèses. Assurez-vous également que l'image que vous ajoutez se situe au même endroit que le fichier CSS que vous avez créée.

Maintenant nous avons donc un `body` contenant une image comme ceci :



Ici l'image que j'ai utilisée est plus grande que mon écran, mais il faut savoir que lorsqu'on ajoute une image en arrière-plan elle va toujours se positionner en haut à droite de la page.



## 8. Positionner une image

Lorsque l'on change l'arrière-plan en CSS, les développeurs aiment bien mettre l'arrière-plan où ils le souhaitent, de base, comme dit plus haut, une image se situe toujours en haut à droite de la page, il suffit donc de placer l'image où on souhaite.

Comme pour modifier la couleur, il faut donc modifier un élément, et comme pour modifier la couleur, il y a plusieurs manières de faire.

L'élément à modifier est le `background-position`, par exemple :

```
body {  
  background-image: url("image.jpg");  
  background-position: center;  
}
```

Ici, pour résumer, voici ce que l'on a fait :

- Dans la balise `body`, nous avons ajouté une image d'arrière-plan grâce à l'élément `background-image`.
- Dans la balise `body`, on a positionné l'image d'arrière-plan au centre.

Pareil que pour l'élément `background-color`, si vous souhaitez positionner l'image à droite il suffit de changer le `center` par `right`.

## 9. Positionner une image – utiliser les pixels

Vous l'aurez compris, ci-dessus vous pouvez placer votre image à gauche, à droite, en haut, en bas, et au centre. Mais maintenant si on souhaite positionner son image à 30 centimètres à droite, comment peut-on faire ?

Eh bien ce n'est pas plus compliqué, il faut juste modifier le `background-position` d'une différente manière, et c'est là que les pixels entrent en jeux.

Un pixel est en fait le système métrique des sites web, il permet de mesurer la distance entre deux points de l'écran. Je vous rassure ce n'est pas très compliqué, à vrai dire lorsqu'on déplace une image avec les pixels, c'est souvent à coup de tâtonnement.

Mais concrètement comment peut-on faire ? Eh bien, ce n'est pas plus compliqué, par exemple, vous souhaitez déplacer votre image de 20 pixels, il suffit d'écrire :

```
body {  
  background-image: url("image.jpg");  
  background-position: 20px;  
}
```

« px » représente l'unité « pixel ».

Cependant, cette technique règle un problème, mais ça reste relativement simpliste. Si l'on souhaite modifier par exemple la position de l'image en hauteur et en largeur, on ne peut pas avec une telle méthode, comme vous pouvez le constater, on ne peut que modifier la position en largeur actuellement.

## 10. Une autre méthode de positionnement, « x » et « y »

Ne fuyez pas ! On n'est pas en mathématique nous parlons par ici d'inconnues que vous devez trouver, mais bel et bien de ce que l'on souhaite modifier.

Il faut savoir qu'une page HTML ou CSS est un rectangle qui a une hauteur (y) et une largeur (x). Et comme les développeurs sont très fainéants, ils ne s'embêtent pas à dire hauteur ou encore largeur, ils préfèrent utiliser la lettre « x » pour la largeur, et « y » pour la hauteur.

Vous l'aurez compris, [background-position](#) permet de modifier la largeur (on peut également modifier la hauteur, mais je préfère faire la distinction entre les deux pour le moment !).

Mais pour modifier la hauteur ce n'est pas plus compliqué, il suffit de préciser que l'on souhaite modifier les « y » (rappelez-vous, développeur = fainéant), comme ceci :

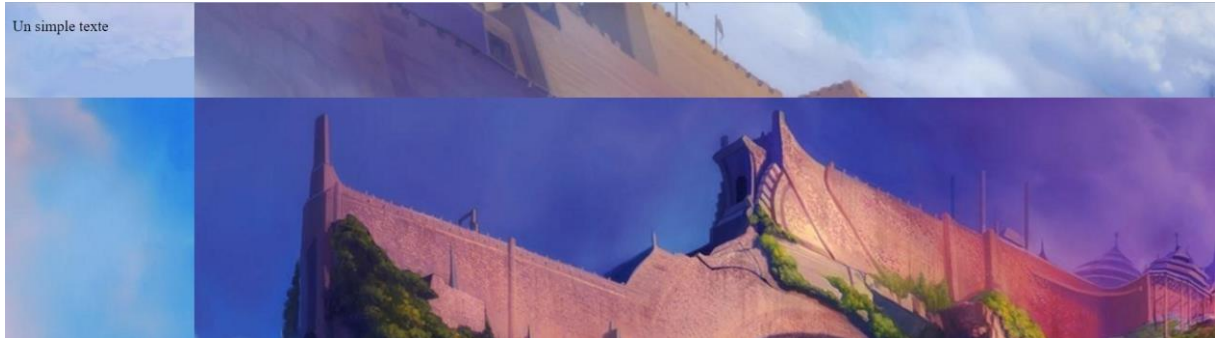
```
body {  
    background-image: url("image.jpg");  
    background-position: 20px;  
    background-position-y: 100px;  
}
```

Vous voyez ? Pas si compliqué que ça ! Il suffit juste de spécifier dans le [background-position](#) ce que l'on souhaite modifier et de combien de pixels nous souhaitons le décaler. Et pour la largeur ? Exactement le même principe.

```
body {  
    background-image: url("image.jpg");  
    background-position-x: 200px;  
    background-position-y: 100px;  
}
```

## 11. Mon arrière-plan se répète, comment faire ?

Normalement, depuis le début vous devez avoir un souci avec votre arrière-plan, ce n'est pas vraiment un problème, mais ça doit vous gêner. Votre arrière-plan doit se répéter un peu comme ceci :



Comment peut-on changer ceci ? Eh bien, c'est très simple, il suffit de dire au CSS que l'on ne doit pas répéter l'arrière-plan comme ceci :

```
body {  
  background-image: url("image.jpg");  
  background-position-x: 200px;  
  background-position-y: 100px;  
  background-repeat: no-repeat;  
}
```

Considérons ça suffisamment explicite pour ne pas avoir à l'expliquer. Maintenant vous devez avoir un fond blanc avec votre image positionné où vous le souhaitez. Vous pouvez également ajouter une couleur si vous ne voulez pas de ce fond blanc ! (Rappelez-vous quelques chapitres plus haut) :

```
body {  
  background-image: url("image.jpg");  
  background-position-x: 200px;  
  background-position-y: 100px;  
  background-repeat: no-repeat;  
  background-color: black;  
}
```

## 12. Changer la couleur du texte

Maintenant que nous avons vu comment modifier l'arrière-plan (comment le positionner, mettre une image, une couleur, etc...) nous allons maintenant voir comment modifier le texte dans la balise `<body>`. Rappelez-vous que nous travaillons dans la balise `<body>` depuis le début, alors, continuons ainsi.

Modifier la couleur du texte n'est pas plus compliqué que modifier la couleur de l'arrière-plan. Par exemple, nous avons utilisés `background-color` pour modifier la couleur de l'arrière-plan, c'est-à-dire que l'on a spécifié au CSS que c'était la couleur de l'arrière-plan que l'on va modifier.

Donc si on suit cette logique, pour modifier la couleur du texte il suffit simplement de faire comme-suit :

```
body {  
  background-image: url("image.jpg");  
  background-position-x: 200px;  
  background-position-y: 100px;  
  background-repeat: no-repeat;  
  background-color: black;  
  color: white;  
}
```

Il n'est pas nécessaire, et même inutile de spécifier au CSS que l'on souhaite modifier la couleur du texte, tout simplement parce qu'on ne peut que modifier la couleur de l'arrière-plan ou celle du texte, si ce n'est pas `background-color` que l'on modifie, alors c'est le texte.

Et comme le `background-color`, `color` peut avoir la couleur écrite en anglais, ou en code couleur.

## 13. Changer la police d'écriture

Sur beaucoup de site web, les créateurs du site aiment bien changer la police d'écriture, par exemple, si on souhaite utiliser la police d'écriture que j'ai utilisé pour faire ce tutoriel (Calibri), eh bien on peut ! Toujours pareil, il suffit de spécifier au CSS que l'on souhaite modifier la police !

```
body {  
    background-image: url("image.jpg");  
    background-position-x: 200px;  
    background-position-y: 100px;  
    background-repeat: no-repeat;  
    background-color: black;  
    color: white;  
    font-family: Calibri;  
}
```

*Font-family* va spécifier au CSS quelle police d'écriture on souhaite utiliser. Pratique non ?

## 14. Changer le style d'écriture

Vous pouvez également changer le style d'écriture, mais avant tout qu'est-ce qu'un style d'écriture ? Eh bien c'est relativement simple, il existe 3 styles d'écritures, *normal*, *italique*, et *oblique* l'oblique et l'italique se ressemblent mais ce ne sont pas les mêmes.

Pour le changer, c'est pareil que pour la police, il suffit de le spécifier au CSS :

```
body {  
    background-image: url("image.jpg");  
    background-position-x: 200px;  
    background-position-y: 100px;  
    background-repeat: no-repeat;  
    background-color: black;  
    color: white;  
    font-family: Calibri;  
    font-style: italic;  
}
```

Vous pouvez remplacer *italic* par *normal* ou *oblique*. Avec toutes ces modifications voici ce que j'obtiens quand je tape du texte dans le `<body>` :

*Un simple texte*

Cependant, si on veut le mettre en gras, comment faire ?

## 15. Changer la forme de l'écriture

Si l'on souhaite mettre le texte en gras par exemple, il s'agit de modifier la forme et non le style cette fois-ci :

```
body {  
  background-image: url("image.jpg");  
  background-position-x: 200px;  
  background-position-y: 100px;  
  background-repeat: no-repeat;  
  background-color: black;  
  color: white;  
  font-family: Calibri;  
  font-style: italic;  
  font-weight: bold;  
}
```

Ici, on spécifie au CSS qu'on veut que la forme du texte soit en gras (bold en anglais signifie « gras »). Une fois appliqué, voici ce que ça me donne :

***Un simple texte***

## 16. Changer la taille du texte

Pour changer la taille du texte, il suffit de modifier l'élément *font-size*, et devinez quoi ? Lui aussi se modifie en pixel !

```
body {  
  background-image: url("image.jpg");  
  background-position-x: 200px;  
  background-position-y: 100px;  
  background-repeat: no-repeat;  
  background-color: black;  
  color: white;  
  font-family: Calibri;  
  font-style: italic;  
  font-weight: bold;  
  font-size: 35px;  
}
```

## 17. Modifier un certain type de texte

Qu'est-ce que j'entends par « modifier un certain type de texte » ? Eh bien, c'est très simple, par exemple, si vous copiez-collez mon CSS que vous montre depuis le début, vous verrez que je ne modifie que le `<body>` et je le précise depuis le début.

Donc, par exemple, si dans le html, je fais ceci :

```
<body>
  <p>Un simple texte</p>
  <h1>Un simple titre</h1>
  <h2>Un simple titre plus petit</h2>
</body>
```

Eh bien, vous verrez qu'en plus d'avoir les propriétés du html (par exemple, le texte contenu dans `h1` sera plus grand que le texte contenu dans `p`, etc...), ils auront les mêmes propriétés de texte (dans mon cas, un texte blanc, gras en italique).

La question est la suivante : Comment je peux faire pour modifier uniquement les paragraphes par exemple ?

Depuis le début, je n'arrête pas de vous dire que l'on modifie les propriétés de `<body>`, eh bien, si l'on souhaite modifier uniquement les propriétés de `<p>` par exemple, il suffit de dire au CSS que l'on ne modifie plus `<body>`, mais `<p>`. Par exemple, ci-dessous j'ai copié toutes les propriétés de texte du `<body>` dans `<p>` :

```
body {
  background-image: url("image.jpg");
  background-position-x: 200px;
  background-position-y: 100px;
  background-repeat: no-repeat;
  background-color: black;
}
```

On garde uniquement les propriétés pour l'arrière-plan dans le corps de la page.

```
p {
  color: white;
  font-family: Calibri;
  font-style: italic;
  font-weight: bold;
  font-size: 35px;
}
```

On modifie les propriétés du texte dans chaque balise `<p>` utilisés dans le HTML.



Si maintenant, je reprends cet exemple en HTML :

```
<body>
  <p>Un simple texte</p>
  <h1>Un simple titre</h1>
  <h2>Un simple titre plus petit</h2>
</body>
```

Voici à quoi ressemble le résultat :



Ne prenez pas en compte le surlignement en bleu des titres, comme la couleur n'est plus définie pour les balises `<h1>` et `<h2>`, le texte est de base noire. J'ai donc dû sélectionner ces titres pour les voir.

L'important ici est que nous avons bel et bien la modification prise en compte uniquement pour le texte à l'intérieur de la balise `<p>`.



## 18. Les bordures

Très souvent dans les site web, on peut voir que le texte est entouré de quelque chose, par exemple comme ceci :

I have a blue left border.

De temps en temps, ce sont des images d'arrière-plan, mais il est tout à fait possible, en css de faire des bordures. En d'autres termes, si on regarde bien l'exemple ci-dessus, on a juste un arrière-plan bleu avec une bordure à gauche en bleu plus foncé.

Et vous allez voir, ce n'est vraiment pas compliqué à faire.

### 18.1. Changer le style des bordures

Pour changer le style des bordures, il existe un élément en CSS qui se nomme *border-style*. Et des valeurs, il y en a beaucoup, ci-dessous, une liste non-exhaustive des bordures existantes :

- **dotted** – Une bordure constituée de points.
- **dashed** – Une bordure constituée de tirets
- **solid** – Une bordure solide
- **double** – Une bordure constituée de deux bordures solide

Pour cet exemple, j'ai choisi d'utiliser la bordure *solid*. Pour faire une bordure, nous n'allons pas le faire dans le `<body>` tout simplement parce que mettre un contour dans le corps de la page n'a pas vraiment de sens, nous allons plutôt le mettre dans les balises `<p>`, comme ceci :

```
p {  
  color: white;  
  font-family: Calibri;  
  font-style: italic;  
  font-weight: bold;  
  font-size: 35px;  
  border-style: solid;  
}
```

Vous voici maintenant avec une bordure modifiée, comme ceci :

Un simple texte

## 18.2. Changer la taille d'une bordure

Pour changer la taille d'une bordure, rien de plus simple, il suffit de modifier l'élément *border-width* :

```
p {  
  color: white;  
  font-family: Calibri;  
  font-style: italic;  
  font-weight: bold;  
  font-size: 35px;  
  border-style: solid;  
  border-width: 10px;  
}
```

Et comme la position de l'arrière-plan ou encore la taille de l'écriture, cet élément se modifie avec les pixels.

## 18.3. Changer la couleur de la bordure

Pour changer la couleur de la bordure, il suffit de modifier l'élément *border-color*, cet élément, à l'instar de *background-color* et *color* utilise soit la couleur en anglais, soit un code couleur :

```
p {  
  color: white;  
  font-family: Calibri;  
  font-style: italic;  
  font-weight: bold;  
  font-size: 35px;  
  border-style: solid;  
  border-width: 10px;  
  border-color: red;  
}
```

## 18.4. Les bordures « one-sided »

La première fois que vous avez lu ce titre, il vous a peut-être fait peur, mais pas de panique, ce n'est pas très compliqué ! Vous vous rappelez de cet exemple ?

I have a blue left border.

Eh bien, comment peut-on faire pour n'avoir qu'une seule bordure sur le côté gauche ? Ou même une sur le côté gauche et une sur le côté droit ?

En fait c'est ce que l'on appelle des bordures « one-sided » ces bordures n'impactent qu'un seul et même côté.

Vous avez vu tout à l'heure que `border-style` permet de modifier la bordure, mais en fait ce que `border-style` fait réellement c'est que cet élément modifie les 4 bordures, haut, bas, gauche, droite.

Si vous souhaitez modifier le style de la bordure de gauche (par exemple) il faudra utiliser `border-left-style`. Voici la liste des différentes bordures :

- `border-left-style` : permet de modifier la bordure de gauche
- `border-right-style` : permet de modifier la bordure de droite
- `border-top-style` : permet de modifier la bordure du haut
- `border-bottom-style` : permet de modifier la bordure du bas

Par exemple, si je veux la bordure de gauche solide, et la bordure d'en bas également, mais pas de bordure ni en haut, ni à droite, voici ce que je dois faire :

```
p {  
  color: white;  
  font-family: Calibri;  
  font-style: italic;  
  font-weight: bold;  
  font-size: 35px;  
  border-width: 10px;  
  border-color: red;  
  border-left-style: solid;  
  border-bottom-style: solid;  
}
```

Si vous copiez-collez ce bout de code, vous aurez ce rendu :

Un simple texte

## 19. Modifier qu'une seule balise

Dans cette partie nous allons voir que chaque paragraphe par exemple peut être distingué, comment ? C'est ce que nous allons voir.

Ici ça risque d'être légèrement plus compliqué que depuis le début, mais vous allez voir que ce n'est pas si dur que ça.

Lorsque je parle de modifier qu'une seule balise, je ne parle pas de ce que l'on a fait plus haut, car (et vous avez dû le remarquer depuis), si vous modifiez la balise `<p>` dans le CSS, alors, à chaque fois que vous utiliserez la balise `<p>` dans votre code HTML, vous aurez la même modification.

Et si on souhaite modifier une seule balise, comment peut-on faire ? Si on veut par exemple que le premier paragraphe contienne le texte en rouge, et le suivant en bleu, comment peut-on faire ?

### 19.1. Les identifiants en HTML

Dans ce tutoriel, nous allons également fait un peu de HTML, ici, lorsqu'on souhaite modifier un seul paragraphe, il faut pouvoir l'identifier, et vous allez-voir, c'est très simple ! il suffit d'utiliser ce que l'on appelle un « identifiant ».

Un identifiant va se placer dans la balise et commence par le mot clé `class`, et est suivi de cet identifiant :

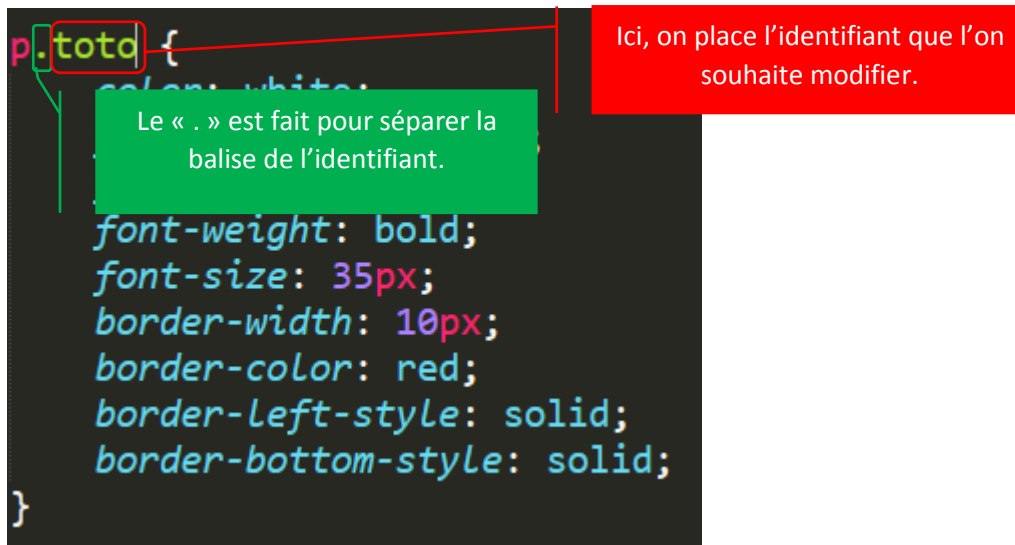
```
<body>
  <p>Un simple texte</p>
  <p class="toto">Un paragraphe identifié qui s'appel toto</p>
  <h1>Un simple titre</h1>
  <h2>Un simple titre plus petit</h2>
</body>
```

Ici, on a bien un paragraphe identifié comme étant le paragraphe « toto ». Le gros avantage de cette méthode est qu'on peut faire des modifications que sur les paragraphes qui vont s'appeler « toto » par exemple, et laisser les autres sans modification par exemple.

## 19.2. Les identifiants en CSS

Mais le but final de cette technique est de pouvoir modifier dans le CSS un paragraphe bien spécifique (j'ai pris le paragraphe en exemple ici, mais on peut le faire avec n'importe quelle autre balise).

Pour identifier en CSS c'est tout aussi simple, il suffit de non plus de spécifier uniquement que l'on modifie `<p>`, mais que l'on modifie `<p.toto>`, comme ceci :



### 19.3. Modifier les balises, les identifiants existant

Lorsque vous naviguez sur des sites, il se peut que vous voyiez des boutons qui changent de couleur, par exemple lorsque vous avez la souris dessus ou lorsque vous cliquez dessus.

Il faut savoir que le CSS a déjà des identifiants prédéfinis, il s'agit en fait d'identifiants permettant « d'animer » votre site en fonction de l'action de l'utilisateur (lorsqu'il clique ou pose la souris sur l'élément etc...).

Nous allons voir seulement lorsque la souris va être sur la zone de texte (pour reprendre l'exemple ci-dessus).

Il suffit simplement de modifier la balise, mais cette fois-ci uniquement du côté du CSS et non plus du HTML :

```
p.toto: hover {  
  color: white;  
  font-  
  font-  
  font-  
  font-size: 35px;  
  border-width: 10px;  
  border-color: red;  
  border-left-style: solid;  
  border-bottom-style: solid;  
}
```

Les « : » sont obligatoires c'est pour dire que l'identifiant est celui du CSS et non le vôtre.

Identifiant qui permet de dire « quand la souris est dessus ».

## 20. Concevoir un site web

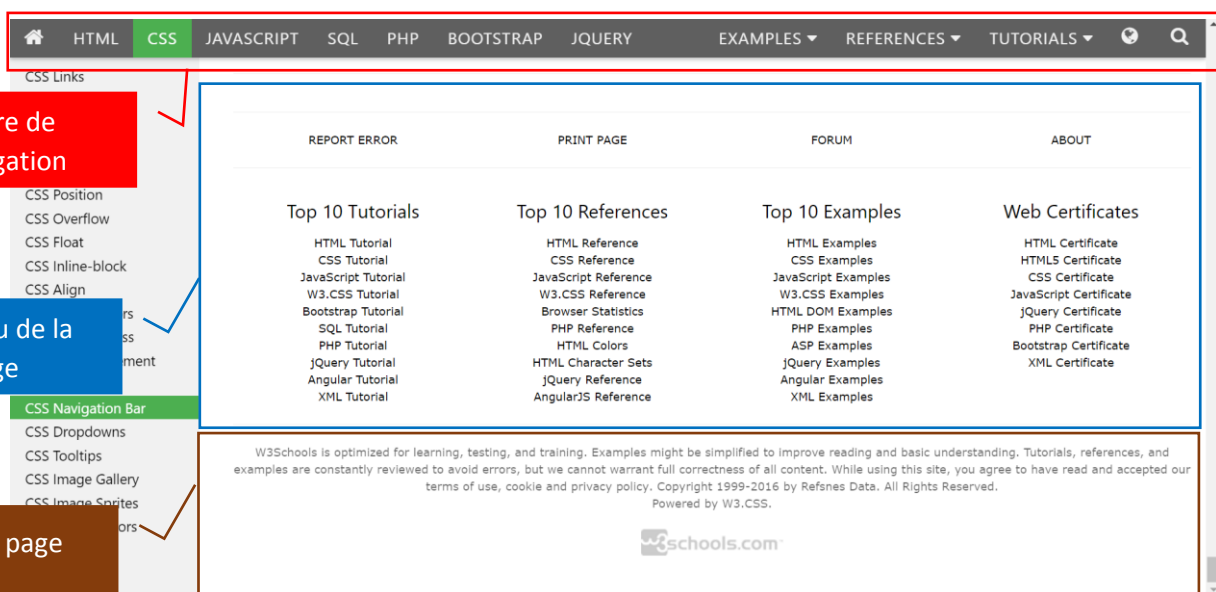
Ce chapitre ne montre pas de « code » à proprement parlé, mais il me semble intéressant de dégrossir ce qu'est un site web en CSS.

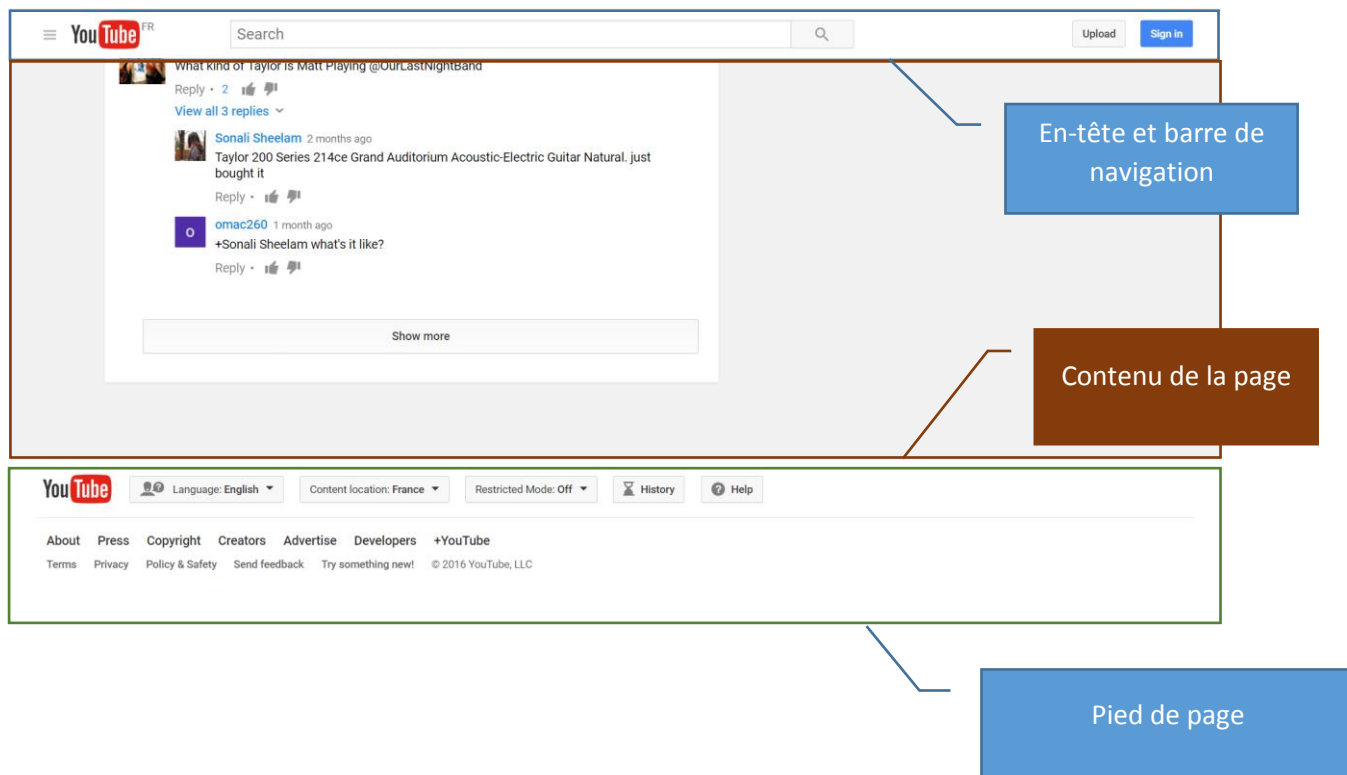
Il faut savoir qu'un site web est avant tout une page interprétée par votre navigateur, comme vous avez pu le voir dans les divers challenges proposés, il y a quand-même une sacrée différence entre un site web ne contenant pas de CSS et le même site web contenant du CSS.

Toutefois, on peut voir qu'un certain « schéma » se dessine sur chaque site web correctement fait sur internet, ce dit-schéma ressemble à :

- Un en-tête (qui contient souvent le logo ou le sujet principal du site)
- Une barre de navigation (qui contient les différentes pages afin de naviguer facilement dans le site)
- Un contenu de page
- Un pied de page (contenant des liens utiles ou encore le nom du créateur du site).

Voici deux site web (j'ai pris intentionnellement deux sites ayant une architecture complètement différente pour que vous voyez que ça n'est qu'un « convention » respectée ou non) et vous pouvez observer que ce schéma est plus ou moins respecté :





Généralement, comme sur le site de youtube, on peut voir que l'en-tête et la barre de navigation sont souvent les deux mêmes choses, comme je l'ai dit, lorsque vous faites un en-tête, vous devez faire en sorte qu'il ne soit pas vide, donc, lorsque vous avez juste à mettre le logo, on préférera le mettre sur la barre de navigation.

Cependant, sur le premier exemple vous voyez qu'il n'y a pas qu'une seule barre de navigation. Tout simplement parce que dans chaque catégorie, il y a également d'autres catégories.

Bien, mais maintenant pourquoi je vous ai parlé de tout ça ? Eh bien c'est simple ! Au début de ce tutoriel nous avons vu les « bases » du CSS, maintenant, entrons dans le vif du sujet et voyons comment on peut construire notre site avec ce que l'on a appris.

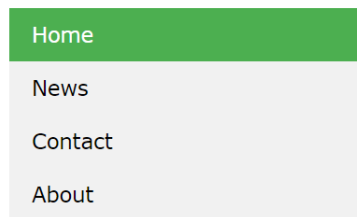


## 21. La barre de navigation

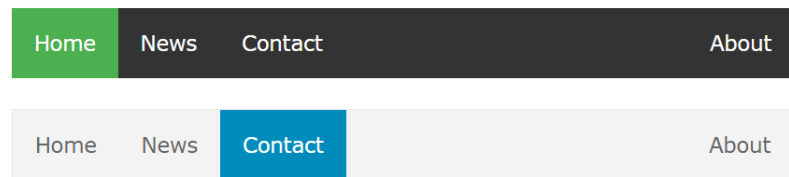
La première chose que j'aime définir personnellement dans un site web est la barre de navigation. En effet, c'est l'une des choses les plus importantes sur un site web. Si vous avez une barre de navigation trop complexe ou peu visible, ça peut entraver la navigation de l'utilisateur et c'est très gênant !

Et des manières de faire des barres de navigation, il y en a plein, toutefois, on aime bien faire deux sortes de barre de navigation : les barres de navigation horizontale et verticale, comme ceci :

Vertical



Horizontal



Mais concrètement, une barre de navigation, c'est quoi ? Eh bien c'est simplement une liste de lien. Ce qui fait qu'une simple barre de navigation en html et sans aucun css ressemble à ceci :

```
<ul>
  <li><a href="default.asp">Home</a></li>
  <li><a href="news.asp">News</a></li>
  <li><a href="contact.asp">Contact</a></li>
  <li><a href="about.asp">About</a></li>
</ul>
```

Cela dit, une liste de lien comme ceci donne un rendu assez basique comme ceci :

- [Home](#)
- [News](#)
- [Contact](#)
- [About](#)

Pas très jolie n'est-ce pas ? Mais ne vous inquiétez pas ! Avec juste quelques `border` et `background-color`, on va changer ça ! Dans un premier temps, on va se débarrasser de la décoration des listes (le point noir à gauche) et des liens (la couleur bleue soulignée), comme ceci :

```
ul.menu {  
    list-style-type: none;  
}  
  
a.menu {  
    text-decoration: none;  
}
```

Vous noterez que j'ai mis les nouveaux styles dans la classe menu, c'est assez important de faire ça, car si on dit que toutes les balises `<a>` n'ont plus de décoration, à chaque fois que vous ferez un lien, il n'y aura aucune distinction entre un lien et du texte normal.

## 21.1 Les barres de navigations horizontales

Dans ce chapitre, nous allons voir comment faire des barres de navigations simple et horizontale. Vous allez voir ce n'est pas compliqué !

La première chose à faire, c'est dire que chaque élément de la liste doit être disposé sur la même ligne, tout simplement comme ceci :

```
li.menu {  
    display: inline;  
}
```

Ici, normalement vous devez avoir quelque chose comme ceci (j'ai fait exprès de laisser la décoration sur les liens afin de bien différencier les liens d'un simple texte) :

[Home](#) [News](#) [Contact](#) [About](#)

Comme vous pouvez le constater, il est assez simple de modifier la liste comme bon vous semble par la suite, si vous souhaitez espacer un petit peu plus, il vous suffit d'utiliser le `padding` comme ceci :

```
a.menu {  
    padding: 8px;  
}
```

J'ai ajouté le padding sur la balise `<a>` car c'est uniquement le texte que je veux décaler, remarquez que j'aurais très bien pu l'ajouter à la balise `<li>` mais je trouve que ça donne une « cohérence » au code ainsi.

Si vous souhaitez maintenant un arrière-plan sur toute la longueur de votre barre de navigation, rien de plus simple ! Il vous suffit juste de modifier le `background-color` de la balise `<ul>` comme ceci :

```
ul.menu {  
    list-style-type: none;  
    background-color: #333;  
    width: 100%;  
}
```

Vous devez d'ores et déjà être familier avec le `background-color`, cependant, le 100% situé sur width doit vous être légèrement inconnu. Lorsque je vous ai dit que la taille était mesurée en pixel, eh bien c'est vrai ! Mais il n'y a pas que le pixel comme unité, le pourcentage est également l'une d'entre-elles.

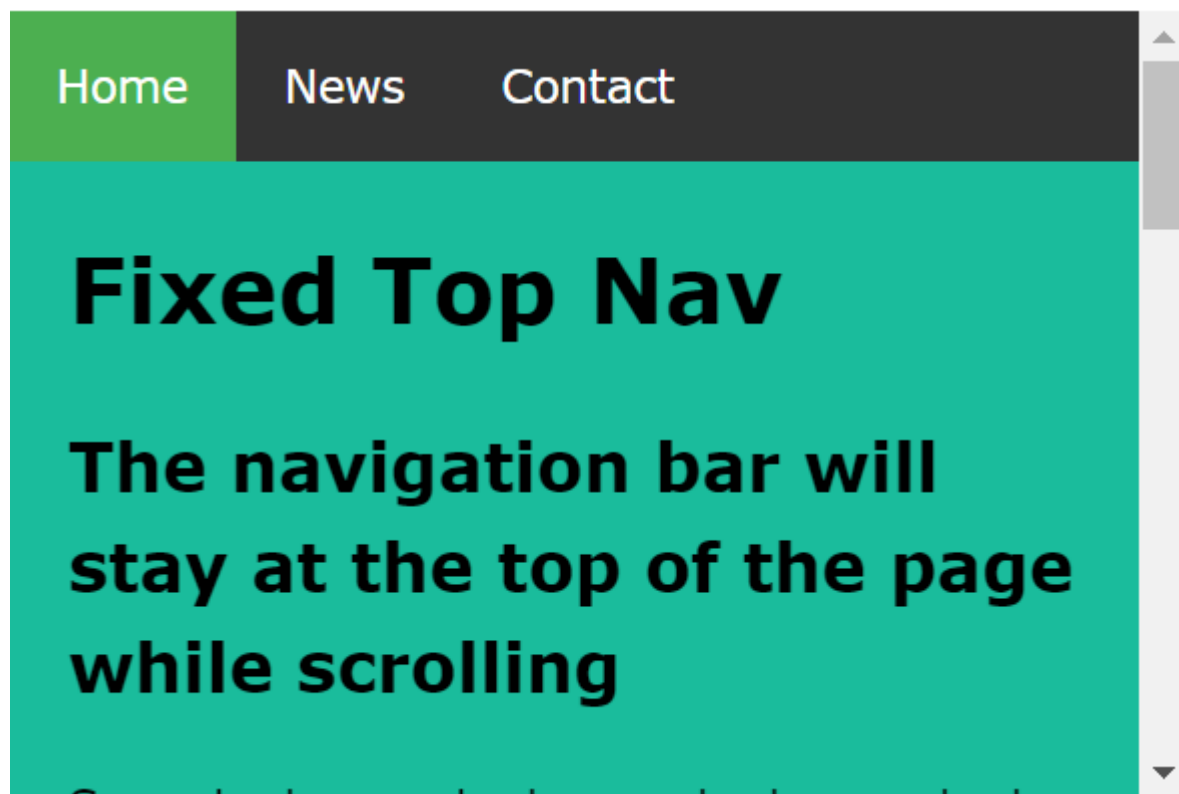
L'avantage du pourcentage est qu'il n'est pas nécessaire de connaître la taille de votre écran pour ajuster correctement ce que vous souhaitez.

Et enfin pour finir, j'aimerais introduire les barres de navigation fixe. Par exemple, lorsque vous naviguez sur un site, il se peut que la taille totale du contenu dépasse la hauteur de votre écran, vous êtes donc obligé de descendre pour tout lire, mais, lorsque vous souhaitez aller sur une autre page, vous devez remonter pour accéder à la barre de navigation ! Pas très pratique !

Fort heureusement, le CSS gère également ceci, il vous suffit de dire que la *position* de la barre de navigation est fixe, comme ceci :

```
ul.menu {  
  list-style-type: none;  
  background-color: #333;  
  width: 100%;  
  top: 0;  
  position: fixed;  
}
```

Après toutes ces modifications, voici à quoi ressemble ma barre de navigation :



## 21.2 Les barres de navigations verticales

Pour être honnête, les barres de navigations verticales utilisent le même principe que les barres de navigation horizontale. Seulement deux petits détails vont changer : la première chose qui va changer va être le `display`. Comme vous avez pu le voir tout à l'heure, le `display` permet de régler l'affichage de la liste, donc, au lieu de mettre le `display` à `inline`, il vous suffit de le mettre à `block`.

```
li.menu {  
    display: block;  
}
```

**Cependant, maintenant que vous avez modifier « l'alignement » de votre liste, vous devez aussi modifier le padding et la largeur de cette dernière.**

La deuxième chose à changer est lorsque vous souhaitez fixer votre barre de navigation, une seule chose va changer, nous n'allons pas mettre la largeur à 100%, mais la hauteur, comme ceci :

```
ul.menu {  
    list-style-type: none;  
    background-color: #333;  
    height: 100%;  
    position: fixed;  
}
```

## 22. La notion de padding

Lorsque vous allez concevoir votre première barre de navigation, vous aurez sûrement envie de déplacer les éléments avec un « padding ».

Un « padding » c'est en fait une distance entre deux éléments. Lorsque vous allez créer votre liste de lien, vous pouvez dire à votre page, « je veux que chacun de mes liens soient décalés de 30 pixels sur la droite » etc... Et tout ça passe par le padding.

Le principe est exactement le même que pour le [background-position](#) que l'on a vu au tout début, vous pouvez utiliser les pixels comme le pourcentage, les deux fonctionnent.

Il existe 4 types de padding :

- [padding-top](#) : déplace vers le bas
- [padding-bottom](#) : déplace vers le haut
- [padding-left](#) : déplace vers la droite
- [padding-right](#) : déplace vers la gauche

Cependant soyez vigilant ! N'en abusez pas, car lorsque vous utilisez un padding sur une zone de page, dites-vous bien que cette zone ne sera plus utilisable :



Les zones encadrées en rouge ici sont les zones qui sont « remplies » par le padding. Bien entendu ici ce n'est pas très dérangeant.

## 23. Pour conclure

Félicitation ! Vous avez terminé le tutoriel CSS. Si ça vous a plu, je vous invite à aller visiter le site <http://www.w3schools.com/css> qui ont des tutoriaux très bien fait sur les langages du web.