

# Margin et propriétés sur les textes

## Margin et propriétés sur les textes

Syntaxe de base du CSS

Les unités de longueur (Référence)

Les unités relatives

Le danger de la composition

Les unités absolues

Des propriétés sur le texte

Exemple HTML de cette section

La taille du texte, la propriété `font-size` (Référence)

La taille du texte, la propriété `font-weight` (Référence)

La taille du texte, la propriété `font-style` (Référence)

La décoration du texte, la propriété `text-decoration` (Référence)

Les valeurs de `text-decoration-line` (Référence)

Les valeurs de `text-decoration-color` (Référence)

Compatibilité des navigateurs

Les valeurs de `text-decoration-style` (Référence)

Compatibilité des navigateurs

Exemple de la décoration

La taille du texte, la propriété `text-align` (Référence)

Les ombres sur un texte, la propriété `text-shadow` (Référence)

Compatibilité des navigateurs

Des propriétés sur les « boîtes »

L'espace extérieur *haut* d'une boîte, la propriété `marge-top` (Référence)

L'espace extérieur *haut* d'une boîte, la propriété `marge-bottom` (Référence)

L'espace extérieur *haut* d'une boîte, la propriété `marge-left` (Référence)

L'espace extérieur *haut* d'une boîte, la propriété `marge-right` (Référence)

La super-propriété `marge` (Référence)

Exemple dans le cours

## Syntaxe de base du CSS

Le CSS se compose d'un ensemble de blocs de style. Chaque bloc commence par un « *appel* » à une balise. Par la suite, nous attribuons une série de propriétés à l'intérieur acrobate ouvrante `{` et fermante `}`.

```
balise {  
  propriété-1: valeur;  
  propriété-2: valeur;  
  propriété-3: valeur;  
  //...  
}
```

La balise peut appliquer son style sur les identifiants aussi. Il faut comprendre, alors, que le style est appliqué sur l'ensemble de la balise avec l'identifiant mentionné (voir notre [cours sur les identifiants](#)). Dans ce cas de figure, la syntaxe du CSS change légèrement. Au lieu d'« *appeler* » une balise, vous appelez votre identifiant. Toutefois, votre identifiant doit être précédé du symbole dièse `#` (exactement comme les liens) <sup>1</sup>. **Tout le reste** de la syntaxe **demeure identique**.

```
#identifiant {  
  propriété-1: valeur;  
  propriété-2: valeur;  
  propriété-3: valeur;  
  //...  
}
```

De plus, il est possible de réaliser la même chose avec toutes nos classes (que nous avons vu dans notre [cours sur les classes](#)). Dans ce cas, notre syntaxe se modifie tout comme les identifiants. Au lieu d'utiliser le dièse # pour « appeler » votre classe, vous devez utiliser le symbole <sup>1</sup> point ..

```
.classe {  
  propriété-1: valeur;  
  propriété-2: valeur;  
  propriété-3: valeur;  
  //...  
}
```

Enfin, lorsque votre bloc de style s'applique à plus d'une balise, d'un identifiant ou d'une classe, vous pouvez tous les appeler en séparant chaque élément par une virgule.

```
balise, #identifiant, .classe {  
  propriété-1: valeur;  
  propriété-2: valeur;  
  propriété-3: valeur;  
  //...  
}
```

## Les unités de longueur ([Référence](#))

Tout comme les couleurs, il existe de nombreuses unités de longueur en CSS. Chacune d'elle permet de mesurer la taille d'écriture ou d'un trait d'une manière différente. Il existe deux types d'unité les unités relatives et les unités absolues.

### Les unités relatives

Les unités relatives sont mesurées par rapport à une référence. Prenez votre écran en exemple. Je pourrais vous donner une unité qui s'appelle *écran*. Cette unité mesure la longueur des objets par rapport à votre écran. Ainsi, un objet qui a une longueur d'un écran a la même longueur que votre écran. Une longueur de deux écrans correspond à deux fois votre écran. Pour le CSS, nous avons les unités suivantes :

Nom de l'unité	Symbole (utilisé en CSS)	Signification	Équivalent
Pixel	px	Représente la taille selon le nombre de pixels. Cette unité dépend de l'écran ou du média.	
	em	Représente la taille selon la taille <code>font-size</code> du parent. Si la balise parent a une taille de <code>16px</code> alors <code>1em</code> vaut ce nombre. <b>Attention à la composition</b>	$1em \approx 16px$
	rem	Représente la taille par rapport à l'élément <code>&lt;html&gt;&lt;/html&gt;</code> . Si la taille de base du navigateur est <code>16px</code> alors <code>1rem</code> vaut ce nombre pour tous les éléments.	
Pourcentage	%	Représente la taille par rapport à l'élément parent. Par exemple, <code>1 %</code> correspond à $1\% \approx \text{taille}_{parent} / 100$	
	ex	Représente la taille par rapport à la taille de la lettre minuscule <code>x</code> de la police d'écriture utilisée.	

## Le danger de la composition

L'un des dangers de l'unité `em` est la composition. Pour faire une histoire courte, lorsque nous avons plusieurs balises imbriquées l'une dans l'autre, nous appelons l'ensemble une **composition**. Le problème de l'unité `em` est qu'elle définit la taille selon la taille d'écriture de **son** parent. Prenons par exemple le code suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Exemple</title>
    <link href="style.css" rel="stylesheet" >
  </head>
  <body>
    <!-- -->
    <h1>Un titre</h1>
    <h2>Un sous-titre</h2>
    <p>
      Ceci est un texte sans être contenu dans une balise
      <strong>span</strong>.
      <br >
      <span>
        Cette balise contient du texte :<br>
        <span>
          Qui est aussi contenu dans une autre balise.
        </span>
      </span>
    </p>
```

```
</body>
</html>
```

Les balises `<p>` et `<span>` imbriquée l'une dans l'autre est une composition. Or, si nous définissons la taille d'écriture des balises `<span>` à `2em`. Alors, le texte de la balise `<p>` sera d'une taille normale, le texte de la première balise `<span>` sera le double de la taille de `<p>` (sa balise parente) et le texte de la deuxième balise `<span>` sera le double de son parent **à lui** soit la première balise `<span>`.

### Un titre

#### Un sous-titre

Ceci est un texte sans être contenu dans une balise `span`.

Cette balise contient du texte :

# Qui est aussi contenu dans une autre balise.

À l'inverse, si nous définissons la taille des balises `<span>` à `2rem`, alors la taille de la deuxième balise sera identique à la première, car la taille est définie à partir de la balise `<html>` et elle est donc globale à toute la page.

## Les unités absolues

Les unités absolues ne dépendent pas des éléments de la page ou de l'écran. Elles seront identiques pour tous les écrans. Parmi ces unités, nous retrouvons les unités de longueur essentiellement.

- `in` représente un pouce soit environ 2.54 centimètres ;
- `cm` représente un centimètre ;
- `mm` représente un millimètre ;
- `pt` représente un point soit, par convention, `1/72` de pouce ;
- `pc` représente un pica soit 12 points.

## Des propriétés sur le texte

Depuis le début de ce cours, nous parlons de la taille de la police d'écriture (qui la même chose que la taille du texte). Bien que les unités s'appliqueront aussi aux autres propriétés CSS, nous allons commencer par voir comment modifier du texte en CSS.

### Exemple HTML de cette section

Pour la prochaine section, nous utiliserons le code HTML suivant :

```
<p>
  <span class="normal">
    Ceci est un texte en HTML modifié par la classe normal.
  </span>
  <br >
  <span class="ex-1">
    Ceci est un texte en HTML modifié par la classe ex-1.
  </span>
  <br >
  <span class="ex-2">
    Ceci est un texte en HTML modifié par la classe ex-2.
  </span>
  <br >
```

```

<span class="ex-3">
    Ceci est un texte en HTML modifié par la classe ex-3.
</span>
<br >
<span class="ex-4">
    Ceci est un texte en HTML modifié par la classe ex-4.
</span>
<br >
</p>

```

Prenez note que la classe `normal` correspond uniquement aux valeurs de base du navigateur.

## La taille du texte, la propriété `font-size` ([Référence](#))

Cette propriété permet de modifier la taille d'écriture du texte selon la taille donnée. Il existe plusieurs tailles prédéfinies dans le navigateur : *xx-small*, *x-small*, *small*, *medium*, *large*, *x-large* et *xx-large*. Il est aussi possible d'utiliser les mots clés *smaller* et *larger* qui signifie respectivement **plus petit que** et **plus grand que** la taille de l'élément parent.

Par la suite, il est possible d'utiliser l'une des unités précédentes dans le calcul de la taille de notre texte. Prenons l'exemple CSS suivant :

```

.ex {
    font-size: 1ex;
}
.ex-1 {
    font-size: 2em;
}
.ex-2 {
    font-size: 64px;
}
.ex-3 {
    font-size: 150%;
}
.ex-4 {
    font-size: 3ex;
}

```

Le résultat obtenu est alors :

Ceci est un texte en HTML modifié par la classe normal.

Ceci est un texte en HTML modifié par la classe ex-1.

Ceci est un texte en HTML modifié par la classe ex-2.

Ceci est un texte en HTML modifié par la classe ex-3.

Ceci est un texte en HTML modifié par la classe ex-4.

## La taille du texte, la propriété `font-weight` ([Référence](#))

Cette propriété modifie le « poids » de l'écriture. Le poids correspond à l'épaisseur du trait. C'est-à-dire que plus le trait est gros, plus le texte est gras. Il existe plusieurs valeurs possibles soit les tailles nommées : *normal*, *bold*, *bolder*, *lighter* ou les nombres 100, 200, 300, 400, 500, 600, 700, 800 et 900. À noter que les tailles 400 et 700 correspondent respectivement à *normal* et *bold*. Il faut aussi noter que ce ne sont pas toutes les polices d'écriture qui supportent toutes les tailles mentionnées.

Prenons l'exemple CSS suivant :

```

.ex-1 {
    font-weight: 400;
}
.ex-2 {
    font-weight: bold;
}
.ex-3 {
    font-weight: 700;
}
.ex-4 {
    font-weight: 900;
}

```

Le résultat obtenu est alors :

Ceci est un texte en HTML modifié par la classe normal.  
 Ceci est un texte en HTML modifié par la classe ex-1.  
**Ceci est un texte en HTML modifié par la classe ex-2.**  
**Ceci est un texte en HTML modifié par la classe ex-3.**  
**Ceci est un texte en HTML modifié par la classe ex-4.**

## La taille du texte, la propriété `font-style` ([Référence](#))

Cette propriété applique un effet visuel (style) au texte. Elle prend trois valeurs possible : *normal*, *italic* et *oblique*. Elle sert principalement à mettre du texte en italique. En général, la valeur *oblique* a le même impact que la valeur *italic*. Prenons l'exemple CSS suivant :

```

.ex-1 {
    font-style: italic;
}
.ex-2 {
    font-style: oblique;
}
.ex-3 {
    font-weight: 900;
    font-style: italic;
}
.ex-4 {
    font-weight: 900;
    font-style: oblique;
}

```

Le résultat obtenu est alors :

Ceci est un texte en HTML modifié par la classe normal.  
*Ceci est un texte en HTML modifié par la classe ex-1.*  
*Ceci est un texte en HTML modifié par la classe ex-2.*  
***Ceci est un texte en HTML modifié par la classe ex-3.***  
***Ceci est un texte en HTML modifié par la classe ex-4.***

## La décoration du texte, la propriété `text-decoration` ([Référence](#))

Cette propriété permet d'ajouter une *décoration* au texte. Par les décorations, nous entendons des lignes. Ces lignes peuvent être sous le texte (souligné), par-dessus le texte et dessus le texte (barré). Toutefois, cette propriété possède une syntaxe particulière. Elle est appelée **super-propriété**. Pour l'utiliser, la syntaxe est la suivante <sup>2</sup> :

```
text-decoration: (text-decoration-line)+ text-decoration-color text-decoration-style;
```

### Les valeurs de `text-decoration-line` ([Référence](#))

Cette propriété permet de définir le type de ligne. Il existe trois possibilités :

Mot clé	Effet	Exemple
<code>overline</code>	Mets une ligne <b>par-dessus</b> le texte.	<u>Cette phrase a une ligne par-dessus.</u>
<code>underline</code>	Mets une ligne <b>sous</b> le texte	<u>Cette phrase a une ligne sous elle.</u>
<code>line-through</code>	Mets une ligne <b>sur</b> le texte.	<del>Cette phrase a une ligne sur elle.</del>






De plus, il est possible de combiner plusieurs mots clés ensemble. Pour y parvenir, nous ajoutons les deux mots clés.

Mot clé	Effet	Exemple
<code>overline underline</code>	Mets une ligne <b>par-dessus</b> et <b>sous</b> le texte.	<u><u>Cette phrase a une ligne par-dessus.</u></u>
<code>underline overline</code>	Mets une ligne <b>sous</b> le texte	<u><u>Cette phrase a une ligne sous elle.</u></u>
<code>line-through underline</code>	Mets une ligne <b>sur</b> le texte.	<del><u>Cette phrase a une ligne sur elle.</u></del>

### Les valeurs de `text-decoration-color` ([Référence](#))

Cette propriété a été ajoutée avec **CSS3**. Elle permet de modifier la couleur de la décoration. Par défaut, cette valeur est la même que celle du texte. De plus, nous lui donnons les mêmes valeurs que pour la couleur du texte.

## Compatibilité des navigateurs



Navigateur					
Version minimale	57.0	79.0	36.0	12.1	44.0
Modificateur <code>-moz-</code>	N/A	N/A	6.0	N/A	N/A
Modificateur <code>-webkit-</code>	N/A	N/A	N/A	7.1	N/A

## Les valeurs de `text-decoration-style` ([Référence](#))

Cette propriété permet de modifier le style de la ligne. Elle a été ajoutée avec le CSS3 et elle possède cinq valeurs possibles. De plus, elle est toujours utilisée avec une valeur `text-decoration-line`. Par défaut, sa valeur est `solid`.

Mot clé	Effet	Exemple
<code>solid</code>	La ligne sera <b>pleine</b> .	<u>La décoration est une ligne pleine.</u>
<code>wavy</code>	La ligne sera <b>ondulée</b> .	<u>La décoration est une ligne ondulée.</u>
<code>dotted</code>	La ligne sera sous la forme de <b>points</b> .	<u>La décoration est une ligne sera sous la forme de point.</u>
<code>dashed</code>	La ligne sera sous la forme de <b>tiret</b> .	<u>La décoration est une ligne sera sous la forme de tiret.</u>
<code>double</code>	La ligne sera <b>doublée</b> .	<u><u>La décoration est une ligne doublée.</u></u>

## Compatibilité des navigateurs

Navigateur					
Version minimale	57.0	79.0	36.0	12.1	44.0
Modificateur <code>-moz-</code>	N/A	N/A	6.0	N/A	N/A
Modificateur <code>-webkit-</code>	N/A	N/A	N/A	7.1	N/A

## Exemple de la décoration

Prenons l'exemple CSS suivant :



```
.ex-1 {
  text-decoration: underline green solid;
}
.ex-2 {
  text-decoration: underline green solid;
}
.ex-3 {
  text-decoration: underline blue solid;
}
.ex-4 {
  text-decoration: underline #056A92 double;
}
```

Le résultat obtenu est alors :

Ceci est un texte en HTML modifié par la classe normal.  
 Ceci est un texte en HTML modifié par la classe ex-1.  
 Ceci est un texte en HTML modifié par la classe ex-2.  
 Ceci est un texte en HTML modifié par la classe ex-3.  
 Ceci est un texte en HTML modifié par la classe ex-4.

## La taille du texte, la propriété `text-align` ([Référence](#))

Cette propriété permet de modifier l'alignement **horizontal** du texte dans son élément parent. Par défaut, le texte est toujours aligné à **gauche** (*left*). Elle possède quatre valeurs.

Mot clé	Effet	Exemple
<code>left</code>	La ligne sera alignée sur la <b>gauche</b> .	Cette phrase est alignée à gauche.
<code>right</code>	La ligne sera alignée sur la <b>droite</b> .	Cette phrase est alignée à droite.
<code>center</code>	La ligne sera alignée au <b>centre</b> .	Cette phrase est alignée au centre.
<code>justify</code>	La ligne sera <b>justifiée</b> (comme dans les journaux) <sup>3</sup> .	Cette phrase est justifiée.

## Les ombres sur un texte, la propriété `text-shadow` ([Référence](#))

Apparue avec le CSS3, la propriété `text-shadow` permet d'afficher un effet d'ombre sous le texte. Il utilise la syntaxe suivante :

```
text-shadow: horizontal verticale blur color;
```

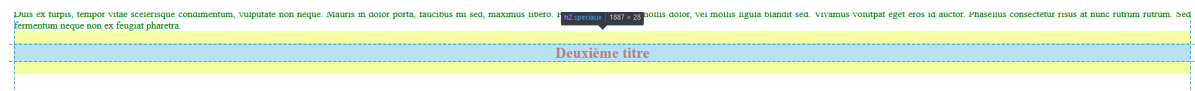
Valeur	Obligatoire	Description
<code>horizontal</code>	✓ Oui	Détermine le décalage horizontal de l'ombre. Cette valeur est numérique. La valeur négative est acceptée et elle s'effectue sur la gauche.
<code>verticale</code>	✓ Oui	Détermine le décalage vertical de l'ombre. Cette valeur est numérique. La valeur négative est acceptée et elle s'effectue vers le haut.
<code>blur</code>	✗ Non	Détermine le dégradé, le temps que la couleur s'efface, de la couleur de l'ombre.
<code>color</code>	✗ Non	La couleur de l'ombre.

## Compatibilité des navigateurs

Navigateur					
Version minimale	4.0	10.0	3.5	4.0	10.0

## Des propriétés sur les « boîtes »

Pour cette semaine, nous nous concentrerons uniquement sur les marges des boîtes. Une boîte est ni plus ni moins que le rectangle qui encadre une balise. Cette boîte est invisible, mais elle est calculée par le navigateur. Grâce aux outils de développement web, nous pouvons « voir » cette boîte. Ici, nous voyons cette boîte pour un titre `<h2></h2>`. La boîte est le trait bleu sur l'image.



## L'espace extérieur *haut* d'une boîte, la propriété `marge-top` ([Référence](#))

La propriété `marge-top` permet de modifier l'espace entre le haut d'une balise et le bas de la balise précédente. S'il s'agit de la première balise, la valeur sera calculée entre la balise et la fenêtre. De plus, la valeur donnée sera calculée selon l'une des unités que nous avons vues précédemment. Prenons l'exemple suivant :

```
p {
  margin-top: 150px;
}
```

Le résultat obtenu est alors :

Ceci est un texte en HTML modifié par la classe normal.  
Ceci est un texte en HTML modifié par la classe ex-1.  
Ceci est un texte en HTML modifié par la classe ex-2.  
Ceci est un texte en HTML modifié par la classe ex-3.  
Ceci est un texte en HTML modifié par la classe ex-4.

Ceci est un texte en HTML modifié par la classe normal.  
Ceci est un texte en HTML modifié par la classe ex-1.  
Ceci est un texte en HTML modifié par la classe ex-2.  
Ceci est un texte en HTML modifié par la classe ex-3.  
Ceci est un texte en HTML modifié par la classe ex-4.

## L'espace extérieur *haut* d'une boîte, la propriété `marge-bottom` ([Référence](#))

La propriété `marge-bottom` permet de modifier l'espace entre la base d'une balise et le haut de la balise précédente. De plus, la valeur donnée sera calculée selon l'une des unités que nous avons vu précédemment. Sur plusieurs navigateurs, la valeur par défaut est 16px. Prenons l'exemple suivant :

```
p {  
  margin-bottom: 40px;  
}
```

Le résultat obtenu est alors :

Ceci est un texte en HTML modifié par la classe normal.  
Ceci est un texte en HTML modifié par la classe ex-1.  
Ceci est un texte en HTML modifié par la classe ex-2.  
Ceci est un texte en HTML modifié par la classe ex-3.  
Ceci est un texte en HTML modifié par la classe ex-4.

Ceci est un texte en HTML modifié par la classe normal.  
Ceci est un texte en HTML modifié par la classe ex-1.  
Ceci est un texte en HTML modifié par la classe ex-2.  
Ceci est un texte en HTML modifié par la classe ex-3.  
Ceci est un texte en HTML modifié par la classe ex-4.

## L'espace extérieur *haut* d'une boîte, la propriété `marge-left` ([Référence](#))

La propriété `marge-left` permet de modifier l'espace entre la balise et ce qui se trouve à sa gauche. De plus, la valeur donnée sera calculée selon l'une des unités que nous avons vues précédemment. Prenons l'exemple suivant :

```
p {  
  margin-left: 15px;  
}
```

Le résultat obtenu est alors :

Ceci est un texte en HTML modifié par la classe normal.  
Ceci est un texte en HTML modifié par la classe ex-1.  
Ceci est un texte en HTML modifié par la classe ex-2.  
Ceci est un texte en HTML modifié par la classe ex-3.  
Ceci est un texte en HTML modifié par la classe ex-4.

## L'espace extérieur *haut* d'une boîte, la propriété `margin-right` ([Référence](#))

La propriété `margin-right` permet de modifier l'espace entre la balise et ce qui se trouve à sa gauche. De plus, la valeur donnée sera calculée selon l'une des unités que nous avons vues précédemment.

## La super-propriété `margin` ([Référence](#))

Grâce à cette propriété, vous pouvez modifier l'ensemble des marges (`top`, `bottom`, `left` et `right`) en une seule ligne. Il s'agit d'une super-propriété. C'est-à-dire qu'elle regroupe plusieurs propriétés ensemble. Pour l'utiliser, il existe différentes syntaxes.

- Pour avoir la même marge partout (en haut, à droite, en bas et à gauche), on utilise la syntaxe suivante :
- 

```
margin: longueur;
```

- Pour avoir la même marge à la verticale (en haut et en bas) et une marge différente à l'horizontale (à gauche et à droite), on utilise la syntaxe suivante :

```
margin: verticale horizontale;
```

- Pour avoir une marge en haut, une autre en bas et une à la verticale, on utilise la syntaxe suivante :

```
margin: haut horizontale bas;
```

- Pour avoir une marge différente sur les quatre côtés, on utilise la syntaxe suivante :

```
margin: haut droite bas gauche;
```

## Exemple dans le cours

Durant notre cours, nous avons travaillé sur une page. Vous trouverez l'exemple ci-bas :

- La page du groupe 1
- La page du groupe 2

---

1. Il ne doit pas y avoir d'espace entre le symbole (`#` ou `.`) et le nom de l'identifiant ou de la classe. [↗](#)

2. Le symbole (groupe)+ signifie que le groupe est présent au moins une fois ou plus. Cette notation est inspirée de la forme Extended Backus-Naur. [↗](#)

3. Selon l'INLB, il n'est pas recommandé d'utiliser cet alignement. Ce dernier ne favorise pas l'accessibilité par les personnes mal voyante. [↗](#)