

Travailler avec la Typographie

Le domaine de typographie web a considérablement progressé au fil du temps. Il y a différentes raisons à sa montée en puissance, la raison la plus largement répandue est le développement d'un système pour intégrer nos propres polices Web sur un site Web.

Dans le passé nous étions limités à un petit nombre de type de caractères que nous pouvions utiliser sur un site Web. Ces polices de caractères correspondaient aux polices les plus couramment installées sur ordinateurs, elles étaient les plus susceptibles d'être correctement affichées à l'écran. Si une police n'était pas installée sur un ordinateur, elle ne pouvait pas être affichée sur le site. Maintenant, avec la possibilité d'incorporer les polices à un site Web, nous pouvons choisir dans une plus grande palette de types de caractères.

Bien que la capacité d'incorporer des polices nous donne accès à d'innombrables nouvelles polices de caractères, il est aussi important pour nous de connaître les principes de base de la typographie.

Dans cette leçon nous allons jeter un oeil à certains de ces principes de base et comment les appliquer à nos pages Web utilisant HTML et CSS.

Police de caractères vs. Police

Les termes "*police de caractères*" et "polices" sont souvent interchangeables, provoquant la confusion. Une *police de caractères* est ce que nous voyons, c'est l'impression artistique, la façon dont le texte est rendu à l'écran.

Une *police* est un fichier qui contient une police de caractères. L'utilisation d'une police sur un ordinateur permet à l'ordinateur d'accéder à la police de caractères.

Une façon d'aider à clarifier la différence entre une police de caractères et une police est de les comparer à une chanson et un MP3. Une police de caractères est très similaire à une chanson en ce qu'elle est une œuvre d'art. Elle est créée par un artiste ou des artistes et est ouverte à l'interprétation du public. Une police, d'autre part, est très similaire à un MP3 en ce qu'il n'est pas l'impression artistique elle-même, mais seulement une méthode de présentation du contenu artistique.

Ajouter de la couleur au texte

Typiquement une des premières décisions nous allons faire lors de la construction d'un site web est de choisir les polices de caractères et les couleurs de texte principale à utiliser. Bien qu'il existe un certain nombre d'autres propriétés qui peuvent être changées -taille, poids...- la police et la couleur du texte ont généralement le plus grand impact sur l'apparence et la lisibilité d'une page. Se débarrasser des valeurs par défaut du navigateur et utiliser immédiatement notre propre police caractères et couleur de texte donne le ton de notre page.

La seule propriété que nous avons pour définir la couleur du texte est la propriété `color`. La propriété `color` accepte une valeur de couleur, mais dans de nombreux formats différents. Ces formats, comme nous l'avons déjà vu, incluent les mots clés, les valeurs hexadécimales, et les valeurs RGB, RGBA, HSL et HSLA. Les valeurs hexadécimales sont les plus répandues, car ils fournissent le plus de contrôle avec le moins d'effort.

Jetons un coup d'oeil à la CSS nécessaire pour changer la couleur de tout le texte dans l'élément `<html>` sur une page.

```
1 html {  
2   color: #555;  
3 }
```

Modification des propriétés de police

CSS offre beaucoup de propriétés différentes pour éditer le rendu du texte sur une page. Ces propriétés entrent dans deux catégories: les propriétés basées sur la police et les propriétés de texte. La plupart de ces propriétés sera précédée de `font-*` ou de `text-*`. Pour commencer nous allons discuter des propriétés sur la base des polices.

Font-family

La propriété `font-family` est utilisée pour déclarer qu'elles font -ainsi que quelle fonts de substitution- sera utilisée pour afficher le texte. La valeur de la propriété `font-family` peut contenir plusieurs noms de polices, toutes séparées par des virgules.

La première police déclarée, à partir de la gauche, est le choix de la police principale. Si la première police n'est pas disponible, les polices alternatives sont déclarées après par ordre de préférence de gauche à droite.

Le noms des polices est constitué de deux ou plusieurs mots qui doivent être délimités par des guillemets. En outre, la dernière des police doit être une valeur de mot clé, qui correspond à la police par défaut du système pour le type spécifié, le plus souvent soit `sans-serif` ou `serif`.

La propriété `font-family` en action :

```
1 body {  
2   font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
3 }
```

Dans ce cas, de `Helvetica Neue` est la police demandée par défaut pour l'affichage. Si cette police est indisponible ou non installé sur un périphérique donné, la prochaine police dans la liste -Helvetica- sera utilisée, et ainsi de suite.

Font-size

La propriété `font-size` offre la possibilité de définir la taille du texte à l'aide de valeurs de longueurs communes y compris les pixels, les unités em, les pourcentages, les points ou les mots clés

Ci dessous la CSS fixe une `font-size` de 14 pixels sur l'élément `<body>`:

```
1 body{
2   font-size: 14px;
3 }
```

Font-style

Pour mettre un texte en italique, ou pour empêcher un texte d'être en italique, nous allons utiliser la propriété `font-style`. La propriété `font-style` accepte quatre valeurs mot-clé: `normal`, `italic`, `oblique`, et `inherit`. Parmi ces quatres, les plus couramment utilisées sont `italic` (ensembles texte en italique) et `normal` (retours de texte à son style normal).

Le CSS suivant définit tous les éléments de classe `special` avec un style de police en italique :

```
1 .special{
2   font-style: italic;
3 }
```

Font-variant

Pas souvent utilisée, mais de temps en temps si le texte doit être mis en petites capitales. Pour ce cas précis nous allons utiliser la propriété `font-variant`. La propriété `font-variant` accepte trois valeurs: `normal`, `small-caps` et `inherit`. Les valeurs les plus en vue sont `normal` et `small-caps` qui sont utilisées pour commuter entre les variantes `normal` et `small-caps` des polices.

Pour commuter tous éléments avec une classe `firm`, nous allons utiliser une `font-variant` de `small-caps`.

```
1 .firm{
2   font-variant: small-caps;
3 }
```

Font-weight

A l'occasion, nous voulons mettre un texte ou une lettre en gras. Dans ce cas nous allons utiliser la propriété `font-weight`. La propriété `font-weight` accepte soit des mot-clés soit des valeurs numériques.

Les valeurs mots-clés comprennent `normal`, `bold`, `bolder`, `lighter` et `inherit`. De ces valeurs de mot clé, il est recommandé d'utiliser principalement `normal` et `bold` pour changer le texte de normal à gras et vice versa. Plutôt que d'utiliser les valeurs des mots clés `bolder` ou `lighter`, il est préférable d'utiliser une valeur numérique pour contrôler la graisse d'un élément.

Dans la pratique, voici le CSS pour définir la `font-weight` en gras pour tous les éléments avec la classe `important` :

```
1 .important {  
2   font-weight: bold;  
3 }
```

Les valeurs numériques 100, 200, 300, 400, 500, 600, 700, 800, et 900 se rapportent spécifiquement à des polices caractères qui ont plusieurs graisses possibles. L'ordre de ces poids commence avec le poids le plus mince, 100 et évolue jusqu'au poids le plus épais 900. Pour référence, la valeur clé des `normal` correspond à 400 et le mot clé `bold` à 700. Ainsi, toute valeur numérique inférieure 400 sera plus mince, et toute valeur supérieure 700 sera plus épaisse.

Si on fixe la `font-weight` à 600 pour tout les élément de la classe `important`, cela rend ce texte comme la valeur `-semi-bold-` donc pas aussi épais que le `bold`, définit plus haut.

```
1 .important {  
2   font-weight: 600;  
3 }
```

Poids des polices

Avant d'utiliser une valeur numérique, nous avons besoin de vérifier et de voir si la police de caractères que nous utilisons propose le poids que nous aimerions utiliser. Tenter d'utiliser un poids qui est indisponible pour une police de caractères donnée forcera le poids de la police à la valeur par défaut la plus proche. Par exemple, la police de caractères "Times New Roman" à deux poids possible `normal` ou 400, et `bold` ou 700, si vous tentez d'utiliser un poids de 900 la police de caractères aura le poids connexe le plus proche, 700 dans cette cas.

Hauteur de ligne

La hauteur de ligne est la distance entre deux lignes de texte est déclarée en utilisant la propriété `line-height`. La propriété `line-height` accepte toutes les valeurs de longueur générales. La meilleure pratique pour la lisibilité est de définir la `line-height` à environ une fois et demie notre valeur de propriété `font-size`. Cela pourrait être réalisé rapidement en réglant la `line-height` à 150%, ou à 1.5.

Cependant, si nous travaillons avec une grille de base, il peut être préférable d'avoir un peu plus de contrôle sur notre hauteur de ligne en utilisant des pixels.. Dans le CSS ci-après, nous établissons une `line-height` de 22 pixels dans l'élément, plaçant ainsi 22 pixels entre chaque ligne de texte:

```
1 corps{  
2   line-height: 22px;  
3 }
```

La hauteur de ligne peut également être utilisée pour centrer verticalement une seule ligne de texte dans un élément. En utilisant la même valeur pour les propriétés `line-height` et `height` le texte sera centré verticalement:

```
1 .btn {  
2   height: 22px;  
3   line-height: 22px;  
4 }
```

Cette technique peut être utilisée pour les boutons, messages d'alerte et autres blocs de texte d'une seule ligne.

Raccourci des propriétés font

Toutes les propriétés de base de `font` énumérées précédemment peuvent être combinées dans une seule propriété `font`. La propriété `font` peut accepter de multiples valeurs de propriété de base. L'ordre de ces valeurs de propriété devrait être la suivante, de gauche à droite: `font-style`, `font-variant`, `font-weight`, `font-size`, `line-height` et `font-family`.

Ces valeurs de propriété sont énumérés de gauche à droite sans l'utilisation de virgules (sauf pour les noms de police, les valeurs de propriété `font-family` sont séparées par des virgules). Une barre oblique, `/`, est nécessaire entre les valeurs de propriété `font-size` et `line-height`.

Lorsque vous utilisez cette valeur raccourcie, chaque valeur de propriété est facultative *sauf* `font-size` et `font-family`.

```
1 html {  
2   font: italic small-caps bold 14px/22px "Helvetica Neue",  
3   Helvetica, Arial, sans-serif;  
4 }
```

Les Propriétés de police

Jetons un coup d'œil à un exemple qui utilise toutes ces propriétés de base ensemble. Le code HTML et CSS suivant illustre les différentes possibilités lorsque l'on souhaite styler du texte.

HTML

```
1 <h2><a href="#">Je suis un developpeur</a></h2>
2 <p class="ligne">Publié par Dev</p>
3 <p>Chaque jour des développeurs travaillent. Ils travaillent
4 intelligemment dans le but de produire des produits exceptionnels.
5 Ils résolvent des problèmes réels et sont fiers de leur travail.
6 <a href="#">Continuer & # 8230;</a>
7 </p>
```

CSS

```
1 h2, p{
2   color: #555;
3   font: 13px/20px "Helvetica Neue", Helvetica, Arial, sans-serif;
4 }
5 a{ color: #0087cc; }
6 a:hover {
7   color: #ff7b29;}
8 h2{
9   font-size: 22px;
10  font-weight: bold;
11  margin-bottom: 6px;
12 }
13 .ligne {
14   color: #9799a7;
15   font-family: Georgia, Times, "Times New Roman", serif;
16   font-style: italic;
17   margin-bottom: 18px;
18 }
```

Je suis un developpeur

Publié par Dev

Chaque jour des développeurs travaillent. Ils travaillent intelligemment dans le but de produire des produits exceptionnels. Ils résolvent des problèmes réels et sont fiers de leur travail.
[Continuer...](#)

Figure 1
Démonstration sur les font

CSS pseudo-classes

La démonstration utilise ici `a:hover` soit la pseudo-classe CSS `hover`, quelque chose nous avons jamais vu auparavant. Pour référence, les pseudo-classes sont des mots clés qui peuvent être ajoutés à la fin d'un sélecteur de style un élément pour sélectionner un état. La pseudo-classe `hover` permet de modifier les style de l'élément sur lequel l'utilisateur va passer la souris -sans cliquer, juste passer au dessus !-. Dans notre exemple, ce style va s'appliquer quand un élément `<a>` sera "survolé" par la souris, nos éléments `<a>` vont changer de couleur quand il seront "survolé".

En pratique

Nous allons commencer à ajouter des propriétés de police sur notre projet. Nous allons commencer par mettre à jour la police sur l'ensemble de notre texte. Pour ce faire, nous allons appliquer des styles à notre élément `<body>`. Nous allons commencer par mettre une propriété `color` avec la valeur `#888`, et nous allons également ajouter une `font-weight` `300`, `font-size` de `16px`, `line-height` de `22px`, et `font-family` avec les familles suivantes "Open Sans", "Helvetica Neue", Helvetica, Arial et sans-serif, par le biais de la propriété `font` et des valeurs abrégées.

Dans une tentative de garder notre `style.css` fichier aussi organisé que possible, nous allons créer une nouvelle section pour ces styles personnalisés, plaçant juste en dessous de notre `reset.css`.

Nous devons ajouter ce qui suit:

```
1  /*
2  =====
3  styles personnalisés
4  =====
5  */
6  body {
7      color: #888;
8      font: 300 16px/22px "Open Sans", "Helvetica Neue", Helvetica,
9      Arial, sans-serif;
10 }
```

Dans le cours précédent nous avons commencé à ajouter quelques styles typographiques, ajoutant spécifiquement une marge inférieure `margin-bottom` à quelques différents niveaux de titres et paragraphes. Dans la même section du `style.css`, ajoutons une couleur `#648880` au niveau des titres.

```
1 h1, h2, h3, h4 {  
2   color: #648880;  
3 }
```

Pendant que nous y sommes, nous allons ajouter aussi une taille de police pour ces différents niveaux de titre. Nos éléments `<h1>` et `<h2>` utiliseront d'assez grandes valeurs de `font-size` et par conséquent, nous voulons également augmenter leurs valeurs `line-height` pour garder le texte à l'intérieur de ces éléments lisibles.

Pour rappel, nous allons faire leur `line-height` de valeurs 44 pixels, double de la valeur de la base `line-height` fixé dans le `<body>` pour l'ensemble des éléments.

```
1 h1 {  
2   font-size: 36px;  
3   line-height: 44px;  
4 }  
5 h2 {  
6   font-size: 24px;  
7   line-height: 44px;  
8 }  
9 h3 {  
10  font-size: 21px;  
11 }  
12 h4 {  
13  font-size: 18px;  
14 }
```

Nos éléments `<h5>` vont être un peu différents du reste de nos titres de rubriques. En conséquence, nous allons changer leurs styles un peu. Nous allons utiliser une valeur de propriété différente pour `color #a9b2b9` et une `font-size` légèrement plus petite de 14px, et nous allons changer la `font-weight` à 400, ou normal. Par défaut, les navigateurs rendent les titres avec une `font-weight` de bold. Nos rubriques, cependant, sont actuellement toutes avec un `font-weight` de 300. Notre reset au sommet de notre fichier `style.css` a changé la `font-weight` à normal puis notre `font-weight` à 300 au sein de l'élément `<body>`. La `font-weight` de 400 sur les éléments `<h5>` rendra le texte un peu plus épais que le reste de nos autres rubriques et textes.

```
1 h5 {  
2   color: #a9b2b9;  
3   font-size: 14px;  
4   font-weight: 400;  
5 }
```

Notre reset au début de notre feuille de style réinitialise également les styles des navigateur par défaut pour les éléments ``, `<cite>`, et ``, que nous

voulons ajouter. Pour nos éléments `` nous allons vouloir définir une `font-weight` de 400, ce qui équivaut en fait à `normal` et non `bold`, comme la police de caractères que nous utilisons est plus épaisse que la plupart des polices de caractères. Alors, pour nos éléments `<cite>` et `` nous définirons une `font-style` à la valeur *italic*.

```
1 strong{
2     font-weight: 400;
3 }
4 cite, em {
5     font-style: italic;
6 }
```

Continuons en ajoutant des styles à nos éléments d'ancrage. Actuellement ils ont le bleu navigateur par défaut. Donnons la même couleur à nos éléments titre de `<h1>` à `<h4>`. En outre, nous allons utiliser la pseudo-classe `hover` pour changer la propriété `color` d'un gris clair, #648880, quand un utilisateur survole un point d'ancrage.

```
1 / *
2  =====
3  Liens
4  =====
5  * /
6
7 a{
8     color:#648880;
9 }
10 a:hover{
11     color: #a9b2b9;
12 }
```

Maintenant nous allons jeter un oeil à notre `<header>` et mettre à jour nos styles. Nous allons commencer par le logo en ajoutant les propriétés `font-size` de 48px et `line-height` de 44px. En ajoutant aux règles existantes `border-top`, `float` et `padding`. Le nouveau jeu de règles devrait ressembler à ceci:

```
1 .logo{
2     border-top: 4px solid #648880;
3     float: left;
4     font-size: 48px;
5     line-height: 44px;
6     padding: 40px 0 22px 0;
7 }
```

Parce que nous avons un peu contrainte la taille du logo, nous allons ajouter une marge à l'élément `<h2>` dans le `<header>` pour l'équilibrer. Nous allons le faire en

plaçant une valeur d'attribut de classe de `slogan` sur le `<h2>` et puis en utilisant cette classe au sein de notre CSS pour lui appliquer les marges appropriées. Il ne faut pas oublier que les modifications apportées à l'élément `<h3>` doivent se reproduire sur toutes les pages.

HTML

```
1 <h2 class="slogan">Octobre 2016 &ndash; Paris</h2>
```

CSS

```
1 .slogan{
2     margin: 66px 0 22px 0;
3 }
```

Ajoutons une valeur d'attribut de classe de `nav-principale` à l'élément `<nav>` et ajoutez `font-size` de 14px et `font-weight` de 400 pour que la navigation se détache du reste de l'entête.

HTML

```
1 <nav class="nav-principale">
2     ...
3 </nav>
```

CSS

```
1 .nav-principale {
2     font-size: 14px;
3     font-weight: 400;
4 }
```

Avec l'élément `<header>` légèrement amélioré, nous allons jeter aussi un coup d'oeil à notre `<footer>`. Dans la classe `footer-principal`, nous allons changer la `color` et `font-size` pour tout le texte dans l'élément `<footer>`. De plus, nous allons changer la `font-size` de l'élément `<small>` pour 400.

Y compris les styles existants, les styles pour notre section de pied de page principale devrait ressembler ceci:

```
1 .footer-principal {
2     color: #648880;
3     font-size: 14px;
4     padding-bottom: 44px;
5     padding-top: 44px;
6 }
7 .footer-principal small{
8     float: left;
9     font-weight: 400;
10 }
```

Nous allons mettre à jour notre page d'accueil. Nous allons commencer par la section "leader", en augmentant la `line-height` de la section à 44 pixels. Nous augmenterons la `font-size` à 36 pixels pour les `<h2>` et à 24 pixels pour le `<p>`.

```
1 .leader {  
2   line-height: 44px;  
3   padding: 22px 80px 66px 80px;  
4 }  
5 .leader h2 {  
6   font-size: 36px;  
7 }  
8 .leader p {  
9   font-size: 24px;  
10 }
```

Enfin, nous avons un petit problème à résoudre sur notre page d'accueil. Auparavant nous avons tous nos éléments d'ancrage d'une valeur de couleur gris clair quand un utilisateur survole. Cela fonctionne très bien, sauf pour les trois intro sur notre page d'accueil où l'élément d'ancrage enveloppe deux `<h3>` et `<h5>` éléments. Parce que les `<h3>` et `<h5>` éléments ont leur propre définition de couleur, ils ne sont pas affectés par le style du `:hover`.

Nous pouvons résoudre ce problème, mais il va exiger un sélecteur assez compliqué. Nous allons commencer par ajouter une classe `intro` aux trois colonnes sur la page d'accueil.

```
1 <section class="grid">  
2   <section class="intro col-1-3">  
3     <a href="programme.html">  
4       <h5>Au programme...</h5>  
5       <h3>.. </h3>  
6     </a>  
7     <p>Blah blah...</p>  
8   </section>  
9   ...  
10 </section>
```

Avec une classe "intro" en place, nous sommes prêts à faire un peu de CSS et créer un sélecteur assez complexe. Nous allons commencer notre sélection avec la classe "intro" parce que nous ne voulons cibler que les éléments contenus dans la classe. De là nous voulons appliquer des styles qu'aux éléments qui se trouvent dans les éléments d'ancrage lorsque l'on passe la souris dessus; donc nous allons ajouter un sélecteur de type avec une pseudo-classe `:hover`. Enfin, nous allons ajouter le sélecteur de type `h3` pour sélectionner les éléments `<h3>` sur lesquels nous voulons appliquer ces styles.

```

1 .intro a:hover h3 {
2   color: #a9b2b9;
3 }

```

Ouf, ce fut un peu long mais la bonne nouvelle est que notre page d'accueil commence à ressembler à quelque chose et fait preuve d'un peu de personnalité.

Cours HTML/CSS

Octobre 2016 – Paris

[Accueil](#) [Programme](#) [Horaires](#) [Lieu](#) [S'inscrire](#)

Construire son site

Un cours adapté à tout les publics pour produire rapidement un site de qualité pro. Un cours adapté à tout les publics pour produire rapidement un site de qualité pro. Un cours adapté à tout les publics pour produire rapidement un site de qualité pro. Un cours adapté à tout les publics pour produire rapidement un site de qualité pro.

[S'enregistrer maintenant](#)

[Au programme...](#)

[Horaires](#)

[Y aller...](#)

[...des kilomètres de code](#)

[...des heures de code](#)

[...des kilomètres tout court](#)

Le cours HTML CSS le plus complet à l'ouest du Pecos !

Et des heures de code...

Comment venir aux cours HTML / CSS

[Accéder au programme](#)

Figure 2

Notre site a un peu plus de style grâce à une poignée de propriétés de bases...

Application des propriétés au texte

Savoir définir les famille, taille, style, variante, hauteur, poids et hauteur de ligne d'une police est seulement la moitié de la bataille. En plus nous pouvons décider de comment aligner, décorer, transformer et espacer notre texte. Commençons par l'alignement du texte de la page.

Aligner du texte

La propriété `text-align` a cinq valeurs possibles: `left`, `right`, `center`, `justify` et `inherit`. Toutes ces valeurs sont assez simples; comme on peut s'y attendre, elles alignent le texte à gauche, à droite ou au centre, ou elles justifient le texte. Le CSS suivant définit tout le texte d'un paragraphe comme centré:

```
1 p{
2   text-align: center;
3 }
```

La propriété `text-align`, cependant, ne doit pas être confondue avec la propriété `float`. Les valeurs `left` et `right` de `text-align` servent à aligner le texte dans un élément à gauche ou à droite, tandis que les valeurs `left` et `right` de `float` déplace l'élément entier. Parfois la propriété `text-align` va nous donner le résultat souhaité, et d'autres fois nous aurons besoin d'utiliser la propriété `float`.

Habillage du texte

La propriété `text-decoration` offre une poignée de façons d'embellir le texte. Il accepte les valeurs de mot clé: `none`, `underline`, `overline`, `line-through` et `inherit`. L'utilisation de la propriété `text-decoration` varie, mais l'utilisation la plus populaire est de souligner les liens, qui est le style par défaut des navigateurs .

```
1 .note{
2   text-decoration: underline;
3 }
```

Plusieurs valeurs pour la propriété `text-decoration` peuvent être appliquées à un élément à la fois en ajoutant chaque mot clé séparé par un espace.

Indentation du texte

La propriété `text-indent` peut être utilisée pour mettre en retrait de la première ligne de texte dans un élément, comme on le voit dans les publications. Toutes les valeurs de longueur communes sont disponibles pour cette propriété, y compris pixels, points, pourcentages, et ainsi de suite... Les valeurs positives font un retrait de texte vers l'intérieur, tandis que les valeurs négatives font un retrait de texte vers l'extérieur.

Ici, le CSS indente le texte pour tous les éléments `<p>` vers l'intérieur par 20 pixels:

```
1 p{
2   text-indent: 20px;
3 }
```

Ombrage du texte

La propriété `text-shadow` nous permet d'ajouter une ombre ou plusieurs ombres au texte. La propriété prend généralement quatre valeurs, toutes inscrites l'une après l'autre de gauche à droite. Les trois premières valeurs sont des longueurs, et la dernière valeur est une couleur.

Dans les trois valeurs de longueur, la première valeur détermine l'horizontal de l'ombre offset, la seconde valeur détermine l'ombre de décalage vertical, et la troisième valeur détermine le rayon de flou de l'ombre. La quatrième et dernière valeur est la couleur de l'ombre, qui peut être une des valeurs de couleurs utilisées dans la propriété couleur.

La propriété `text-shadow` ici est une ombre noire de 3 pixels vers la droite, 6 pixels vers le bas, et floue 2 pixels et de 30% opaque :

```
1 p {  
2   text-shadow: 3px 6px 2px rgba(0, 0, 0, 0.3);  
3 }
```

L'utilisation de valeurs de longueur négatives pour les décalages horizontaux et verticaux nous permet de placer des ombres vers la gauche et le haut.

Plusieurs ombres de texte peuvent également être enchaînées ensemble à l'aide valeurs séparées par des virgules, ajoutant plus d'une ombre au texte. En utilisant nombreuses ombres nous permet de les placer au dessus et au dessous du texte, ou toute variation désirée.

Box Shadow

La propriété `text-shadow` place une ombre spécifiquement sur le texte d'un élément. Si nous souhaitons placer une ombre sur l'élément dans son ensemble, nous pouvons utiliser la propriété `box-shadow`. La propriété `box-shadow` fonctionne exactement comme la `text-shadow` et accepte les mêmes valeurs de décalage horizontal et vertical, un flou, et une couleur.

La propriété `box-shadow` accepte également une quatrième valeur de longueur optionnelle, avant la valeur de couleur, pour la propagation d'une ombre. En tant que valeur de longueur positive, l'écart va étendre l'ombre plus grande que la taille de l'élément sur lequel elle est appliquée, et en tant que valeur de longueur négative, l'écart va rétrécir l'ombre pour être plus petite que la taille de l'élément sur lequel elle est appliquée.

Enfin, la propriété `box-shadow` peut inclure une valeur `inset` facultative au début de la valeur pour placer l'ombre à l'intérieur d'un élément plutôt qu'à l'extérieur de l'élément.

Text Transform

La propriété `text-transform` est similaire à la propriété `font-variant`. Bien que la propriété `font-variant` cherche une autre variante d'une police de caractères, la propriété `text-transform` va changer la ligne de texte sans avoir besoin d'une police de caractères alternative. La propriété `text-transform` accepte cinq valeurs: `none`, `capitalize`, `uppercase`, `lowercase` et `inherit`.

La valeur `capitalize` mettra en majuscule la première lettre de chaque mot, la valeur `uppercase` mettra en majuscule chaque lettre, et la valeur `lowercase` mettra

en minuscule chaque lettre. Enfin l'utilisation `inherit` ne retournera aucun de ces valeurs héritées mais fera un retour au style de texte original.
Le CSS suivant définit tout le texte de l'élément `<p>` en majuscules:

```
1 p {  
2   text-transform: uppercase;  
3 }
```

Espacement des lettres

En utilisant la propriété `letter-spacing`, nous pouvons ajuster l'espace entre les lettres sur une page. Une valeur de longueur positive va espacer les lettres les unes des autres, tandis qu'une valeur négative va rapprocher les lettres. La valeur du mot clé `none` permet de retrouver l'espace par défaut entre les lettres.

L'utilisation d'une valeur relative de longueur avec la propriété `letter-spacing` aidera à assurer que nous maintenons l'espacement correct entre lettres même si la `font-size` du texte est modifiée.

Avec le CSS ici, toutes les lettres au sein de nos éléments `<p>` apparaîtront rapprochées avec un espace de `-0.5 em`:

```
1 p {  
2   letter-spacing: -.5em;  
3 }
```

Espacement des mots

Tout comme la propriété `letter-spacing`, nous pouvons également ajuster l'espace entre les mots dans un élément utilisant la propriété `word-spacing`. La propriété `word-spacing` accepte les mêmes valeurs et mots clés longueur que la propriété `letter-spacing`. Au lieu d'espacer les lettres, la propriété applique ces valeurs entre les mots.

Ici chaque mot dans un `<p>` seront espacés de `.25 em`:

```
1 p {  
2   word-spacing: .25em;  
3 }
```

Propriétés du texte

Revisitons notre démonstration sur les polices en ajoutant cette fois quelques propriétés de texte en sus de nos propriétés de polices.

HTML

```
1 <h2><a href="#">Je suis un développeur</a></h2>  
2 <p class="ligne">Publié par Dev</p>  
3 <p>Chaque jour des développeurs travaillent. Ils travaillent  
4 intelligemment dans le but de produire des produits exceptionnels.  
   Ils résolvent des problèmes réels et sont fiers de leur travail.
```

```
5 <a href="#">Continuer & # 8230;</a>
6 </p>
7
```

CSS

```
1 h2, p{
2   color: #555;
3   font: 13px/20px "Helvetica Neue", Helvetica, Arial, sans-serif;
4 }
5 a{
6   color: #0087cc;
7 }
8 a:hover {
9   color: #ff7b29;
10 }
11 h2{
12   font-size: 22px;
13   letter-spacing: -.02em;
14   font-weight: bold;
15   margin-bottom: 6px;
16 }
17 h2 a{
18   text-decoration: none;
19   text-shadow: 2px 2px 1px rgba(0, 0, 0, .2);
20 }
21 .ligne {
22   color: #9799a7;
23   font-family: Georgia, Times, "Times New Roman", serif;
24   font-style: italic;
25   margin-bottom: 18px;
26 }
27 .intro {
28   text-indent: 15px;
29 }
30 .intro a{
31   font-size: 11px;
32   font-weight: bold;
33   text-decoration: underline;
34   text-transform: uppercase;
35 }
```


En pratique

A l'heure actuelle chaque lien sur la page est souligné, ce qui est le style par défaut pour les éléments d'ancrage. Ce style est un peu pénible alors nous allons changer un peu.

L'ajout à notre section des liens au sein de notre fichier `style.css`, nous allons commencer par supprimer le soulignement de tous les éléments d'ancrage par l'intermédiaire de la propriété `text-decoration`. Ensuite, nous allons sélectionner tous les éléments d'ancrage qui apparaissent dans un élément de paragraphe et leur donner une bordure inférieure.

Nous pourrions utiliser la propriété `text-decoration` au lieu de la propriété `border-bottom` pour souligner tous les liens. Cependant, en utilisant la propriété `border-bottom` nous avons plus de contrôle sur l'apparence du soulignement. Ici, par exemple, le soulignement sera une autre couleur que le texte lui même. Notre section de liens, qui comprend nos styles hover précédents, devrait ressembler à ceci:

```
1 a{
2   color: #648880;
3   text-decoration: none;
4 }
5 a:hover{
6   color: #a9b2b9;
7 }
8 p a{
9   border-bottom: 1px #dfe2e5 solid;
}
```

Pour revenir à nos éléments `<h5>` qui ont des styles légèrement différents que le reste des rubriques, nous allons forcer les majuscules en utilisant la propriété `text-transform`. Nos nouveaux `<h5>` devraient ressembler à ceci:

```
1 h5 {
2   color: #a9b2b9;
3   font-size: 14px;
4   font-weight: 400;
5   text-transform: uppercase;
6 }
```

Revenons à notre élément `<header>` pour appliquer des styles supplémentaires à notre menu de navigation (à laquelle nous avons ajouté précédemment la valeur d'attribut classe `nav-principale`). Après les propriétés existantes, `font-size` et `font-weight` ajoutons un léger `letter-spacing` et mettons notre texte en majuscule via la propriété `text-transform`. Nos styles pour l'élément `<nav>` avec la valeur d'attribut de classe `nav-principale` devrait être maintenant comme ceci:

```

1 .nav-principale{
2   font-size: 14px;
3   font-weight: 400;
4   letter-spacing: .5px;
5   text-transform: uppercase;
6 }

```

Auparavant, nous avons fait flotter notre logo à la gauche dans le `<header>`. Maintenant notre slogan est positionné directement à droite du logo; cependant, nous aimerions qu'il apparaisse tout le chemin à droite de l'élément `<header>`, aligné droite.

Nous avons besoin d'ajouter la propriété `text-align` avec une valeur de `right` pour les `<h3>` avec la valeur d'attribut de classe `slogan` pour que le slogan puisse être vers la droite

Ajoutée à la propriété existante, `margin` nos nouveaux styles pour les `<h3>` avec la valeur d'attribut de classe de `slogan` ressembleront à ceci:

```

1 .slogan{
2   margin: 66px 0 22px 0;
3   text-align: right;
4 }

```

Nous aimerions également que nos menus de navigation, à la fois dans les éléments `<header>` et `<footer>`, soient aligner à droite. Parce qu'à la fois dans le `<header>` et `<footer>` leurs éléments enfants flottent à la gauche, nous allons utiliser la même approche que nous avons fait avec notre classe "slogan".

Les éléments flottant du `<header>` et du `<footer>` sont retirés de l'écoulement normal de la page, ce qui provoque que les autres éléments vont s'envelopper autour d'eux. Dans ce cas précis, nos menus de navigation sont les éléments entourant ces éléments flottants.

Parce que nous devons partager les mêmes styles dans les deux menus de navigation, nous allons leur donner chaque classe `nav`. Notre élément `<header>` va maintenant ressembler à ceci:

```

1 <header class="group container">
2   <h1 class="logo">...</h1>
3   <h2 class="slogan">...</h2>
4   <nav class="nav-principal nav">
5     ...
6   </nav>
7 </header>

```

Et notre élément `<footer>` va maintenant ressemblera à ceci:

```

1 <footer class ="groupe container footer-principal">
2   <small>...</small>
3   <nav class ="nav">

```

```
4     ...
5     </ nav>
6 </ footer>
```

Il ne faut pas oublier les changements à nos éléments `<header>` et `<footer>` doivent être faits sur chaque page.

Avec la classe `nav` en place sur les deux menus de navigation, nous allons créer une nouvelle section au sein de notre fichier `style.css` pour ajouter les styles de navigation partagés. Nous allons commencer par ajouter la propriété `text-align` avec une valeur de `right` à la classe `nav`. Nous allons étendre ces styles plus tard, mais cela servira de base.

```
1 /*
2  =====
3  Navigation
4  =====
5  */
6
7 .nav {
8     text-align: right;
9 }
```

Alors que nous ajoutons la propriété `text-align` avec une valeur de `center` à notre classe `leader`. Ces styles, y compris nos propriétés existantes, `line-height` et `padding` seront situées dans la section de la section "accueil" de notre fichier `style.css`.

```
1 .leader {
2     line-height: 44px;
3     padding: 22px 80px 66px 80px;
4     text-align:center;
5 }
```

Notre site a maintenant un certain style, notre site Web progresse !

Construire son site

Un cours adapté à tout les publics pour produire rapidement un site de qualité pro. Un cours adapté à tout les publics pour produire rapidement un site de qualité pro. Un cours adapté à tout les publics pour produire rapidement un site de qualité pro. Un cours adapté à tout les publics pour produire rapidement un site de qualité pro.

S'enregistrer maintenant

AU PROGRAMME...

...des kilomètres de code

Le cours HTML CSS le plus complet à l'ouest du Pecos !

[Accéder au programme](#)

HORAIRES

...des heures de code

Et des heures de code...

Y ALLER...

...des kilomètres tout court

Comment venir aux cours HTML / CSS

© BCMA

Accueil Programme Horaires Lieu S'inscrire

Figure 3 Notre site s'embellit

Utilisation de polices de caractères Web

Par défaut il y a quelques polices qui sont pré-installés sur chaque ordinateur, tablette, téléphone intelligent -smartphone ;-)-, ou autre appareil de navigation Web compatible. Parce qu'ils ont été installés sur tous ces appareils, nous pouvons utiliser ces polices librement dans nos sites, sachant que peu importe l'appareil à partir duquel on va naviguer sur notre site, la police rendra correctement. Ces polices connues comme "Web-safe" sont seulement une poignée et les plus sûres -safe-. Elles sont répertoriées ci dessous :

- Arial
- Courier New, Courier
- Garamond
- Géorgie
- Lucida Sans, Lucida Grande, Lucida
- Palatino Linotype
- Tahoma
- Times New Roman, Times
- Trebuchet
- Verdana

Embarquer des Web Fonts

Nous avons aussi la possibilité de télécharger les polices sur un serveur et inclure sur dans un site Web via la règle CSS `@font-face`. Cette capacité a fait grandement évoluée la typographie en ligne.

L'incorporation de nos propres polices Web se fait comme suit en CSS. Tout d'abord, nous utilisons la règle `@font-face` pour donner un le nom à notre police via la propriété `font-family`, ainsi que la source de notre police (le chemin d'accès au fichier de police contenant notre police choisie) via la propriété `src`. À partir de là, nous sommes en mesure d'utiliser cette police en incluant son nom dans n'importe quelle valeur de la propriété `font-family`.

```
1 @font-face {  
2   font-family: "Lobster";  
3   src: local("Lobster"), url("lobster.woff") format("woff");  
4 }  
5 body {  
6   font-family: "Lobster", "Comic Sans", cursive;  
7 }
```

Je suis un développeur

Publié par Dev

Chaque jour des développeurs travaillent. Ils travaillent intelligemment dans le but de produire des produits exceptionnels. Ils résolvent des problèmes réels et sont fiers de leur travail.

[Continuer ...](#)

Figure 4 Démo sur les font

Avoir la capacité d'intégrer n'importe quel type de police sur un site Web ne signifie pas que nous pouvons légalement le faire. Les polices de caractères sont des œuvres d'art, et les afficher sur notre serveur peut permettre à d'autres de les voler facilement. L'autorisation d'utiliser une police de caractères dépend de la licence de celles ci.

Heureusement, la valeur de l'utilisation de nouveaux types de caractères en ligne a été reconnue, et les entreprises ont commencées à développer des licences qui permettent d'inclure de nouvelles polices sur les sites Web. Certaines de ces entreprises, comme Typekit et Fontdeck, travaillent sur un modèle d'abonnement pour les licences de polices, tandis que d'autres, comme Google Fonts, ont des licences des polices gratuites. Avant de télécharger des polices, nous allons nous assurer que nous avons la permission de le faire.

En Pratique

Pour ajouter un peu de caractère à notre site Web essayons d'utiliser une police Google sur notre site Web.

Allons-y sur le site [Google Fonts](#) et recherchez la police que nous aimerions utiliser, à savoir "Lato". Une fois que nous l'avons trouvé, nous allons procéder à l'ajout à notre collection et en suivant les étapes sur leur site Web pour utiliser la police. Lorsque le moment est venu de choisir les poids de police que nous aimerions utiliser, nous allons nous assurer de sélectionner 300 et 400, car nous avons déjà utilisé ces valeurs dans notre CSS. Ajoutons également 100 à la collection pour une autre variation possible.

Google nous fournira un élément `<link>` supplémentaire à inclure dans l'élément `<head>` de toutes nos pages. Nous placerons ce nouvel élément `<link>` directement en dessous de notre élément existant `<link>`. Le nouvel élément inclura la référence de feuille de style appropriée à Google, qui prendra soin d'inclure un nouveau fichier CSS avec la bonne règle `@font-face` nécessaire pour que nous puissions utiliser la police *Lato*.

Avec l'ajout du nouvel élément `<link>`, notre élément `<head>` ressemblera à ceci:

```
1 <head>
2   <meta charset="utf-8">
3   <title>Cours HTML/CSS</title>
4   <link rel="stylesheet" href="assets/stylesheets/style.css">
5   <link rel="stylesheet"
6     href="http://fonts.googleapis.com/css?family=Lato:100,300,400">
7 </head>
```

Une fois que nous avons ajouté le nouvel élément `<link>` à toutes nos pages, nous sommes prêts à commencer à utiliser la police Lato. Nous le ferons en l'ajoutant à notre liste de polices dans la propriété `font` dans nos styles d'élément `<body>`. Ajoutons *Lato* au début de notre liste pour avoir "Lato", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif.

Bien que Lato soit un mot unique, parce qu'il s'agit d'une police Web intégrée, nous allons l'entourer de guillemets dans n'importe quelle référence CSS. Nos nouveaux styles d'élément `<body>` ressembleront à ceci:

```
1 body{
2   color: #888;
3   font: 300 16px/22px "Lato", "Open Sans", "Helvetica
4     Neue", Helvetica, Arial, sans-serif;}
```

Lato devrait maintenant être opérationnel, visible dans tout notre texte sur le site. Jetons maintenant un coup d'oeil à notre logo et mettons le à jour.

Dans notre jeu de règles de sélection de classe "logo", nous allons commencer par ajouter la propriété `font-weight` avec une valeur de 100 pour rendre le texte assez mince. Nous utiliserons également la propriété `text-transform` avec une valeur de majuscule pour que toutes les lettres soient affichées en majuscules, ainsi que la propriété d'espacement des lettres avec une valeur de .5 pixels pour ajouter un peu d'espace entre chaque lettre dans le logo.

Au total, les styles de notre logo ressembleront à ceci:

```
1 .logo {  
2   border-top: 4px solid #648880;  
3   float: left;  
4   font-size: 48px;  
5   font-weight: 100;  
6   letter-spacing: .5px;  
7   line-height: 44px;  
8   padding: 40px 0 22px 0;  
9   text-transform: uppercase;  
10 }
```

Parce que nous avons une valeur de 100 pour la `font-size`, nous allons également définir l'élément paragraphe dans notre section "leader" à ce poids. Nous pouvons utiliser le sélecteur existant pour le faire, et le nouveau jeu de règles ressemblera à ceci:

```
1 .leader p {  
2   font-size: 24px;  
3   font-weight: 100;  
4 }
```

Notre site Web a bien avancé, et l'apparence de notre site Web commence à vraiment être sympathique !

Construire son site

Un cours adapté à tout les publics pour produire rapidement un site de qualité pro. Un cours adapté à tout les publics pour produire rapidement un site de qualité pro. Un cours adapté à tout les publics pour produire rapidement un site de qualité pro. Un cours adapté à tout les publics pour produire rapidement un site de qualité pro.

S'enregistrer maintenant

AU PROGRAMME...

...des kilomètres de code

Le cours HTML CSS le plus complet à l'ouest du Pecos !

[Accéder au programme](#)

HORAIRES

...des heures de code

Et des heures de code...

Y ALLER...

...des kilomètres tout court

Comment venir aux cours HTML / CSS

Fig 5
Notre site avec la police Google Lato Google

Les citations et les sources

La rédaction en ligne implique parfois de citer différentes sources ou citations. Tous les différents cas de citations peuvent être couverts sémantiquement en HTML à l'aide des éléments `<cite>`, `<q>` et `<blockquote>`. Parce qu'ils sont généralement distingués du texte régulier en apparence, nous les présentons ici dans la leçon de typographie.

Savoir quand utiliser l'élément et l'attribut pour marquer correctement les citations nécessite un peu de pratique. En général, suivez ces règles:

- `<cite>`: Utilisé pour faire référence à une œuvre de création, à un auteur ou à une ressource
- `<q>`: Utilisé pour les citations courtes en ligne
- `<blockquote>`: Utilisé pour des citations externes plus longues

Citer un ouvrage créatif

L'élément `<cite>` inline est utilisé en HTML pour citer spécifiquement un travail de création; L'élément doit inclure le titre de l'œuvre, le nom de l'auteur ou une référence URL de l'œuvre. Par défaut, le contenu contenu dans l'élément `<cite>` apparaît en italique dans le navigateur.

Pour plus de référence, il est utile d'inclure un lien hypertexte vers la source originale de la citation, le cas échéant.

Ici, le livre Steve Jobs, de Walter Isaacson, est référencé dans l'élément `<cite>`. A l'intérieur de la citation est également un lien hypertexte vers le livre.

```
1 <p>Le livre <cite> <a  
2 href="https://www.amazon.com/Responsive-Design-Principles-Ethan-Ma  
3 rcotte/dp/1937557332/"> d'Ethan Marcotte </a> </cite> est la  
4 référence du responsive design.</p>
```

Mise en forme de dialogue

Très souvent, un dialogue ou une phrase peut être citée sur en ligne, à l'intérieur d'un autre texte. A cet effet, l'élément `<q>` (ou quote) inline doit être appliqué.

L'élément `<q>` indique sémantiquement un dialogue ou de la prose et ne doit pas être utilisé à d'autres fins.

Par défaut, le navigateur insère les guillemets appropriés pour nous et va même modifier les guillemets en fonction de la langue identifiée dans l'attribut global `lang`. Voici un exemple:

```
1 <p>Ethan Marcote a dit <q>Evitez le Divitisme et le Classitisme en  
2 abandonnant les tables pour un balisage plus léger...</q></p>  
3
```

Ajouter une citation à un dialogue

Un attribut facultatif à inclure sur l'élément `<q>` est l'attribut `"cite"`. L'attribut `"cite"` agit comme une référence de citation à la citation sous la forme d'une URL. Cet attribut ne modifie pas l'apparence de l'élément; Il ajoute simplement de la valeur pour les lecteurs d'écran et autres périphériques. Parce que l'attribut n'est pas visible dans le navigateur, il est également utile de fournir un lien hypertexte vers cette source à côté de la citation réelle.

Voici un exemple:

```
1 <p>  
2 <a href="http://ethanmarcotte.com/">Ethan Marcote</a> a dit  
3 <q cite="https://en.wiktionary.org/wiki/Citations:divitis">Evitez  
4 le Divitisme et le Classitisme en abandonnant les tables pour un  
5 balisage plus léger...</q>  
6 </p>
```

Citation externe

Pour citer un gros bloc de texte provenant d'une source externe et couvrant plusieurs lignes, nous utiliserons l'élément `<blockquote>`. Le `<blockquote>` est un élément au niveau du bloc qui peut avoir d'autres éléments au niveau du bloc imbriqués à l'intérieur, y compris les en-têtes et les paragraphes.

Voici un exemple qui utilise l'élément `<blockquote>`:

```
1 <blockquote>
2   <p>Je pense qu'il y a 3 mots sur le web en ce moment qui
3   demandent du travail pour beaucoup d'entre nous : Responsive,
4   Web, Design.</p>
5 </blockquote>
```

Citation externe

Les citations plus longues utilisées dans l'élément `<blockquote>` incluront souvent une citation. Cette citation peut comprendre à la fois l'attribut `cite` et l'élément `<cite>`.

L'attribut `cite` peut être inclus sur l'élément `<blockquote>` - de la même façon qu'il a été utilisé précédemment sur l'élément `<q>` - pour fournir une référence de citation à la citation sous la forme d'une URL. L'élément `<cite>` peut alors être après la citation réelle elle-même pour spécifier la source originale de la citation, le cas échéant.

Le HTML ici décrit une citation étendue d'Ethan Marcotte tirée de son livre "Responsive Web Design". La citation est marquée à l'aide de l'élément `<blockquote>` avec l'attribut `cite` pour spécifier l'origine de la citation. Dans l'élément `<blockquote>`, l'élément `<cite>`, avec un élément `<a>`, fournit une citation et une référence supplémentaires pour la citation qui est visible par les utilisateurs.

```
1 <blockquote cite="http://ethanmarcotte.com/#writing">
2   <p>Je pense qu'il y a 3 mots sur le web en ce moment qui
3   demandent du travail pour beaucoup d'entre nous : Responsive,
4   Web, Design.</p> <p><cite>&#8212; Ethan Marcotte dans <a
5   href="https://abookapart.com/products/responsive-web-design">
6   Responsive Web Design</a></cite></p>
7 </blockquote>
```


