

8. Le layout.

Il y a trois façons d'annoncer CSS au HTML. Le premier est appelé style en ligne. Cela signifie que le CSS est mis "en ligne" avec les éléments HTML en utilisant l'attribut style. Bien que cette méthode puisse sembler simple, elle rend également difficile de maintenir et de mettre à jour le CSS. Vous devez aussi répéter le style pour chaque élément unique auquel il s'applique. Pour ces raisons, le style en ligne est rarement utilisé.

```
<p style="color:red;">
  Ce paragraphe est rouge.
</p>
```

La deuxième façon est appelée style interne. Dans ce cas, le CSS est toujours à l'intérieur du HTML, mais est mis dans la tête en haut du fichier avec la balise de style. Cela vous permet de sélectionner plusieurs éléments à la fois, mais est toujours difficile à mettre à jour car des modifications devraient être apportées à chaque page HTML individuellement. Cette méthode est utile si une seule page a un style différent du reste de votre site Web.

```
<head>
<style>
p {color:red;}
</style>
</head>

<p>Ce paragraphe est rouge.</p>
```

C'est la méthode la plus populaire. Dans ce cas, il y a un fichier CSS séparé du HTML. Plusieurs pages HTML peuvent être liées au même fichier CSS, vous permettant de garder tous les styles dans un seul endroit. Cela permet d'appliquer facilement le même style sur l'ensemble d'un site Web ainsi que de revenir en arrière et de modifier le CSS. Vous utilisez la balise de lien dans la tête du HTML pour lui indiquer où la feuille de style CSS est enregistrée.

```
<head>
<link rel="stylesheet" type="text/css" ref="filename.css">
</head>

<p>Ce paragraphe est rouge.</p>
```

Parfois, les règles CSS entrent en conflit pour un élément donné si plusieurs styles s'y appliquent. Dans ce cas, la règle la plus spécifique prévaudra. Cela signifie que les règles spécifiques d'identification l'emportent sur les règles de classe qui l'emportent sur les règles de balise d'élément. Si des règles contradictoires ont la même spécificité, alors la dernière de la liste aura préséance.

Si nous voulons appliquer un style à un groupe d'éléments ensemble, comme mettre une frontière autour d'un paragraphe et son titre, nous pouvons les regrouper avec un élément **div**. Les divs agissent comme des conteneurs pour d'autres éléments HTML, et en appliquant un style à l'élément div, nous pouvons menacer tous les éléments à l'intérieur comme un seul. Si on veut appliquer un style à une partie seulement d'un élément, disons un seul mot dans un paragraphe, on peut utiliser un élément **span**.

```
<div class="maingroup">
  <h1>Mon <span class="important">Titre</span></h1>
  <p>C'est un paragraphe.</p>
</div>
```

CSS nous permet non seulement de changer l'apparence d'un élément, mais aussi son emplacement sur la page. Nous pouvons le faire avec l'attribut **position**, qui a quatre options : statique (la valeur par défaut), relative, fixe et absolue. Cela indique ce que nous utilisons comme référence pour le poste. **Relative** nous permet de placer l'élément par rapport à sa position habituelle, **fixed** est par rapport à la fenêtre du navigateur, et **absolute** est par rapport à son élément parent. Nous utilisons ensuite les attributs en haut, en bas, à gauche et à droite pour dire où l'élément devrait être.

```
#footer {
    position:fixed;
    bottom: 0;
    left: 0;
}
```

Normalement, les éléments occupent toute la largeur de la fenêtre. Définir un attribut **width** nous permet de corriger cela, mais les éléments ne seront toujours pas dans la même ligne. Nous pouvons corriger cela avec **float**, qui prend deux valeurs : gauche ou droite. La valeur indique peut être utilisée pour placer une colonne sur le côté gauche de votre page, ou pour enrouler du texte autour d'une image.

```
.picture {  
    max-width: 300px;  
    float: left;  
}
```

L'attribut `clear` nous permet de remplacer le paramètre **float** de l'élément précédent. Il garantit qu'un élément efface d'autres éléments avant d'être placé. Cet attribut a trois valeurs : gauche, droite ou les deux, indiquant de quel côté l'élément doit être clair.

```
#footer {  
    clear: both;  
}
```