

Technologie du Web

Prof. Mohamed NABIL

Département Informatique

Année Universitaire 2020/2021

Langage CSS

1. Introduction

CSS signifie feuilles de style en cascade. CSS (Cascading Style Sheets) est un langage de feuille de style standard utilisé pour décrire la présentation (c'est-à-dire la mise en page et le formatage) des pages Web.

Avant CSS, presque tous les attributs de présentation des documents HTML étaient contenus dans le balisage HTML (dans les balises HTML); toutes les couleurs de police, les styles d'arrière-plan, les alignements d'éléments, les bordures et les tailles devaient être explicitement décrits dans le HTML.

En conséquence, le développement des grands sites Web est devenu un processus long et coûteux, car les informations de style étaient ajoutées à plusieurs reprises à chaque page du site Web.

Pour résoudre ce problème, CSS a été introduit en 1996 par le W3C (World Wide Web Consortium). CSS a été conçu pour permettre la séparation de la présentation et du contenu. Désormais, les concepteurs Web peuvent déplacer les informations de mise en forme des pages Web vers une feuille de style distincte, ce qui se traduit par un balisage HTML considérablement plus simple et une meilleure maintenabilité. CSS3 est la dernière version de la spécification CSS. CSS3 ajoute plusieurs nouvelles fonctionnalités de style et améliorations pour améliorer les capacités de présentation Web.

Il y a beaucoup de choses que vous pouvez faire avec CSS. Vous pouvez :

- ✓ Facilement appliquer les mêmes règles de style sur plusieurs éléments.
- ✓ Contrôler la présentation de plusieurs pages d'un site Web avec une seule feuille de style.
- ✓ Présenter la même page différemment sur différents appareils.
- ✓ Styliser les états dynamiques d'éléments tels que le survol, le focus, etc., ce qui n'est pas possible autrement.
- ✓ Modifier la position d'un élément sur une page Web sans modifier le balisage.
- ✓ Modifier l'affichage des éléments HTML existants.
- ✓ Transformer des éléments tels que l'échelle, la rotation, l'inclinaison, etc. dans un espace 2D ou 3D.
- ✓ Créer des animations et des effets de transitions sans utiliser de JavaScript.
- ✓ Créer une version imprimable de vos pages Web.
- ✓ Etc.

Le plus grand avantage du CSS est qu'il permet de séparer le style et la mise en page du contenu du document. Il y a d'autres avantages de l'utilisation de CSS :

CSS Gagnez beaucoup de temps: CSS donne beaucoup de flexibilité pour définir les propriétés de style d'un élément. Vous pouvez écrire CSS une fois; puis le même code peut être appliqué aux groupes d'éléments HTML, et peut également être réutilisé dans plusieurs pages HTML.

Maintenance facile: CSS fournit un moyen simple de mettre à jour la mise en forme des documents et de maintenir la cohérence entre plusieurs documents. Parce que le contenu de l'ensemble des pages Web peut être facilement contrôlé à l'aide d'une ou plusieurs feuilles de style.

Chargement plus rapide des pages : CSS permet à plusieurs pages de partager les informations de mise en forme, ce qui réduit la complexité et la répétition du contenu structurel des documents. Cela réduit considérablement la taille du transfert de fichiers, ce qui accélère le chargement de la page.

Styles supérieurs au HTML : CSS a des capacités de présentation beaucoup plus larges que HTML et offre un bien meilleur contrôle sur la mise en page de vos pages Web. Ainsi, vous pouvez donner un meilleur aspect à vos pages Web par rapport aux éléments et attributs de présentation HTML.

Compatibilité de plusieurs appareils : CSS permet également d'optimiser les pages Web pour plusieurs types d'appareils ou de supports. En utilisant CSS, le même document HTML peut être présenté dans différents styles d'affichage pour différents périphériques de rendu tels que les ordinateurs de bureau, les téléphones portables, etc.

2. Inclure le CSS dans les documents HTML

Le CSS peut être joint en tant que document séparé ou incorporé dans le document HTML lui-même. Il existe trois méthodes pour inclure du CSS dans un document HTML:

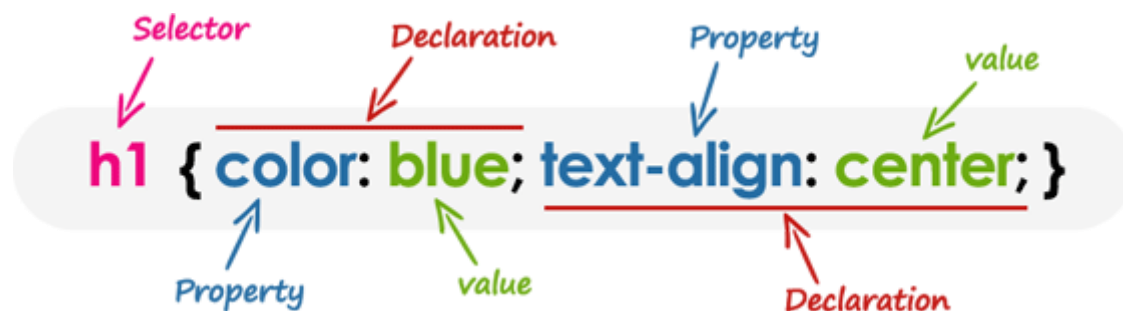
Styles en ligne - Utilisation de l'attribut style dans la balise de début HTML.

Styles incorporés - Utilisation de l'élément `<style>` dans la section d'en-tête d'un document.

Feuilles de style externes - Utilisation de l'élément `<link>`, pointant vers un fichier CSS externe.

3. Syntaxe CSS

Une feuille de style CSS se compose d'un ensemble de règles qui sont interprétées par le navigateur Web puis appliquées aux éléments correspondants tels que les paragraphes, les en-têtes, etc. dans le document. Une règle CSS comprend deux parties principales, un sélecteur et une ou plusieurs déclarations:



Le sélecteur spécifie le ou les éléments de la page HTML auxquels la règle CSS s'applique. Tandis que, les déclarations dans le bloc déterminent comment les éléments sont formatés sur une page Web. Chaque déclaration se compose d'une propriété et d'une valeur séparées par un deux-points (:) et se terminant par un point-virgule (;), et les groupes de déclaration sont entourés d'accolades {}. La propriété est l'attribut de style que vous souhaitez modifier; ils peuvent être la police, la couleur, l'arrière-plan, etc. Chaque propriété a une valeur, par exemple la propriété de couleur peut avoir la valeur bleue ou # 0000FF etc.

```
h1 {color:blue; text-align:center;}
```

Pour rendre le CSS plus lisible, vous pouvez mettre une déclaration sur chaque ligne, comme ceci:

```
h1 {  
    color: blue;  
    text-align: center;  
}
```

Dans l'exemple ci-dessus, h1 est un sélecteur, la couleur et l'alignement du texte sont les propriétés CSS, les données bleu et center sont les valeurs correspondantes de ces propriétés.

- **Commentaire en CSS**

Les commentaires sont généralement ajoutés dans le but de rendre le code source plus facile à comprendre. Cela peut aider un autre développeur (ou vous-même à l'avenir lorsque vous éditez le code source) à comprendre ce que vous essayez de faire avec le CSS. Les commentaires sont importants pour les programmeurs mais ignorés par les navigateurs. Un commentaire CSS commence par /* et se termine par */ , comme indiqué dans l'exemple ci-dessous:

```
/* This is a CSS comment */  
  
h1 {  
    color: blue;  
    text-align: center;  
}  
  
/* This is a multi-line CSS comment that spans across more than one line */  
  
p {  
    font-size: 18px;  
    text-transform: uppercase;  
}
```

- **Sensibilité à la casse en CSS**

Les noms de propriétés CSS et de nombreuses valeurs ne sont pas sensibles à la casse. Alors que les sélecteurs CSS sont généralement sensibles à la casse.

4. Sélecteurs CSS

Un sélecteur CSS est un modèle pour faire correspondre les éléments d'une page Web. Les règles de style associées à ce sélecteur seront appliquées aux éléments qui correspondent au modèle du sélecteur. Les sélecteurs sont l'un des aspects les plus importants du CSS car ils vous permettent de cibler des éléments spécifiques sur votre page Web de différentes manières afin qu'ils puissent être stylisés.

Plusieurs types de sélecteurs sont disponibles en CSS, parmi eux:

- **Sélecteur universel**

Le sélecteur universel, indiqué par un astérisque (*), correspond à chaque élément de la page. Le sélecteur universel peut être omis si d'autres conditions existent sur l'élément. Ce sélecteur est souvent utilisé pour supprimer les marges et les rembourrages par défaut des éléments à des fins de test rapide. Essayons l'exemple suivant pour comprendre son fonctionnement:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example of CSS universal selector</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }
  </style>
</head>
<body>
  <h1>This is heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

Les règles de style à l'intérieur du sélecteur `*` seront appliquées à chaque élément d'un document. Il est recommandé de ne pas utiliser trop souvent le sélecteur universel (`*`) dans un environnement de production, car ce sélecteur correspond à chaque élément d'une page Web qui met trop de pression inutile sur les navigateurs. Utilisez plutôt le type d'élément ou le sélecteur de classe.

- **Sélecteurs de type d'élément**

Un sélecteur de type d'élément correspond à toutes les instances de l'élément dans le document avec le nom de type d'élément correspondant. Essayons un exemple pour voir comment cela fonctionne réellement:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example of CSS element type selector</title>
  <style>
    h1 {
      color: red;
    }
    p {
      color: blue;
    }
  </style>
</head>
<body>
  <h1>This is heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

Les règles de style à l'intérieur du sélecteur `p` seront appliquées à chaque élément `<p>` (ou paragraphe) du document et le coloreront en bleu, quelle que soit leur position dans l'arborescence du document. Et la même chose pour le sélecteur `h1`.

• Sélecteurs d'identifiants

Le sélecteur d'identifiant est utilisé pour définir des règles de style pour un élément unique. Le sélecteur d'**id** est défini avec un signe de hachage (#) immédiatement suivi de la valeur d'id.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example of CSS id selector</title>
  <style>
    #error {
      color: #ff0000;
    }
  </style>
</head>
<body>
  <p id="error">This is a warning!</p>
</body>
</html>
```

Cette règle de style rend le texte d'un élément en rouge, dont l'attribut id est défini sur error. La valeur d'un attribut **id** doit être unique dans un document donné- ce qui signifie qu'aucun élément de votre document HTML ne peut partager la même valeur d'**id**.

• Sélecteurs de classe

Les sélecteurs de classe peuvent être utilisés pour sélectionner tout élément HTML ayant un attribut de classe. Tous les éléments ayant cette classe seront formatés selon la règle définie.

Le sélecteur de classe est défini par un signe de période (.) Immédiatement suivi de la valeur de la classe.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example of CSS class selector</title>
  <style>
    .blue {
      color: #0000ff;
    }
  </style>
</head>
<body>
  <h1 class="blue">This is a heading</h1>
  <p class="blue">This is a paragraph.</p>
  <p>This is another paragraph.</p>
</body>
</html>
```

Les règles de style ci-dessus rendent le texte en bleu de chaque élément du document dont l'attribut de classe est défini sur bleu. Vous pouvez le rendre un peu plus particulier. Par exemple:

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example of CSS class selector</title>
  <style>
    p.blue {
      color: #0000ff;
    }
  </style>
</head>
<body>
  <h1 class="blue">This is a heading</h1>
  <p class="blue">This is a paragraph.</p>
  <p>This is another paragraph.</p>
</body>
</html>

```

La règle de style à l'intérieur du sélecteur **p.blue** rend le texte en bleu uniquement des éléments **<p>** dont l'attribut de classe est défini sur bleu et n'a aucun effet sur les autres paragraphes.

• Sélecteurs descendants

Vous pouvez utiliser ces sélecteurs lorsque vous devez sélectionner un élément qui est le descendant d'un autre élément, par exemple, si vous souhaitez cibler uniquement les ancres contenues dans une liste non ordonnée, plutôt que de cibler tous les éléments d'ancrage. Voyons voir comment ça fonctionne:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example of CSS Descendant Selectors</title>
  <style>
    h1 em {
      color: green;
    }
    ul.menu {
      padding: 0;
      list-style: none;
    }
    ul.menu li{
      display: inline;
    }
    ul.menu li a {
      margin: 10px;
      text-decoration: none;
    }
  </style>
</head>
<body>
  <h1>This is a <em>heading</em></h1>
  <ul class="menu">

```

```

        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Contact</a></li>
    </ul>
</body>
</html>

```

Les règles de style à l'intérieur du sélecteur **ul.menu li a** s'appliquent uniquement aux éléments **<a>** contenus dans un élément **** ayant la classe **.menu**, et n'ont aucun effet sur les autres liens à l'intérieur du document. De même, les règles de style à l'intérieur du sélecteur **h1 em** seront appliquées uniquement aux éléments **** contenus dans l'élément **<h1>** et n'ont aucun effet sur les autres éléments ****.

• Sélecteurs enfants

Un sélecteur d'enfant est utilisé pour sélectionner uniquement les éléments qui sont les enfants directs d'un élément. Un sélecteur enfant est composé de deux sélecteurs ou plus séparés par un symbole supérieur à (>). Vous pouvez utiliser ce sélecteur, par exemple, pour sélectionner le premier niveau d'éléments de liste à l'intérieur d'une liste imbriquée qui a plus d'un niveau. Voyons un exemple pour comprendre comment cela fonctionne:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of CSS Child Selectors</title>
<style>
    ul > li {
        list-style: square;
    }
    ul > li ol {
        list-style: none;
    }
</style>
</head>
<body>
    <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li>
            <a href="#">Services</a>
            <ol>
                <li><a href="#">Design</a></li>
                <li><a href="#">Development</a></li>
            </ol>
        </li>
        <li><a href="#">Contact</a></li>
    </ul>
</body>
</html>

```


La règle de style à l'intérieur du sélecteur **ul> li** s'applique uniquement aux éléments **** qui sont des enfants directs des éléments ****, et n'a aucun effet sur les autres éléments de la liste.

• Sélecteurs de frères adjacents

Les sélecteurs frères adjacents peuvent être utilisés pour sélectionner des éléments frères (c'est-à-dire des éléments au même niveau). Ce sélecteur a la syntaxe comme: **E1 + E2**, où **E2** est la cible du sélecteur.

Le sélecteur **h1 + p** de l'exemple suivant sélectionnera les éléments **<p>** uniquement si les éléments **<h1>** et **<p>** partagent le même parent dans l'arborescence du document et **<h1>** précède immédiatement l'élément **<p>**. Cela signifie que seuls les paragraphes qui viennent immédiatement après chaque titre **<h1>** auront les règles de style associées. Voyons comment ce sélecteur fonctionne réellement:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of CSS Adjacent Sibling Selectors</title>
<style>
    h1 + p {
        color: blue;
        font-size: 18px;
    }
    ul.task + p {
        color: #f0f;
        text-indent: 30px;
    }
</style>
</head>
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
    <ul class="task">
        <li>Task 1</li>
        <li>Task 2</li>
        <li>Task 3</li>
    </ul>
    <p>This is one more paragraph.</p>
    <p>This is also a paragraph.</p>
</body>
</html>
```

• Sélecteurs de frères généraux

Le sélecteur général des frères est similaire au sélecteur des frères adjacent (**E1 + E2**), mais il est moins strict. Un sélecteur frère général est composé de deux sélecteurs simples séparés par le caractère tilde (~). Il peut s'écrire comme: **E1 ~ E2**, où **E2** est la cible du sélecteur.

Le sélecteur **h1 ~ p** de l'exemple ci-dessous sélectionnera tous les éléments **<p>** précédés de l'élément **<h1>**, où tous les éléments partagent le même parent dans l'arborescence du document.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of CSS General Sibling Selectors</title>
<style>
    h1 ~ p {
        color: blue;
        font-size: 18px;
    }
    ul.task ~ p {
        color: #f0f;
        text-indent: 30px;
    }
</style>
</head>
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
    <ul class="task">
        <li>Task 1</li>
        <li>Task 2</li>
        <li>Task 3</li>
    </ul>
    <p>This is one more paragraph.</p>
    <p>This is also a paragraph.</p>
</body>
</html>
```

• Sélecteurs de regroupement

Souvent, plusieurs sélecteurs dans une feuille de style partagent les mêmes déclarations de règles de style. Vous pouvez les regrouper dans une liste séparée par des virgules pour réduire le code dans votre feuille de style. Cela vous empêche également de répéter les mêmes règles de style encore et encore. Nous allons jeter un coup d'œil:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of CSS Selectors without Grouping</title>
<style>
    h1 {
        font-size: 36px;
        font-weight: normal;
    }
    h2 {
```

```

        font-size: 28px;
        font-weight: normal;
    }
    h3 {
        font-size: 22px;
        font-weight: normal;
    }
</style>
</head>
<body>
    <h1>This is a heading of level 1</h1>
    <h2>This is a heading of level 2</h2>
    <h3>This is a heading of level 3</h3>
</body>
</html>

```

Comme vous pouvez le voir dans l'exemple ci-dessus, la même règle de style **font-weight: normal;** est partagé par les sélecteurs **h1**, **h2** et **h3**, il peut donc être regroupé dans une liste séparée par des virgules, comme ceci:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of CSS Grouping Selectors</title>
<style>
    h1, h2, h3 {
        font-weight: normal;
    }
    h1 {
        font-size: 36px;
    }
    h2 {
        font-size: 28px;
    }
    h3 {
        font-size: 22px;
    }
</style>
</head>
<body>
    <h1>This is a heading of level 1</h1>
    <h2>This is a heading of level 2</h2>
    <h3>This is a heading of level 3</h3>
</body>
</html>

```

5. Couleur CSS

La propriété **color** définit la couleur du texte (couleur de premier plan en général) d'un élément. Par exemple, la propriété **color** spécifiée dans le sélecteur de corps définit la couleur

du texte par défaut pour toute la page. Essayons l'exemple suivant pour voir comment cela fonctionne:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting Text Color in CSS</title>
<style>
  body {
    color: #ff5722;
  }
  .text-green {
    color: #008000;
  }
</style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
  <div class="text-green">This is a simple piece of text.</div>
</body>
</html>
```

• Définition des valeurs de couleur

Couleurs en CSS le plus souvent spécifiées dans les formats suivants:

- ✓ La couleur un mot-clé (keyword) - comme "red", "green", "blue", "transparent", etc..
- ✓ Une valeur HEX - comme "# ff0000", "# 00ff00", etc.
- ✓ Une valeur RGB - comme "rgb (255, 0, 0)"

CSS3 a introduit plusieurs autres formats de couleur tels que HSL, HSLA et RGBA qui prennent également en charge la transparence alpha.

Mots-clés de couleur

Ces mots-clés de couleur de base sont: **aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow**. Les noms de couleur sont insensibles à la casse.

Cependant, les navigateurs Web modernes prennent pratiquement en charge beaucoup plus de noms de couleurs que ceux définis dans la norme CSS, mais pour être plus sûr, vous devez utiliser des valeurs de couleur hexadécimales. Pour une liste complète des noms de couleurs possibles, voir la référence suivante <https://www.tutorialrepublic.com/css-reference/css-color-names.php>

• Effet de la propriété de couleur sur les bordures et les contours

La propriété **color** n'est pas seulement pour le contenu textuel, mais pour tout ce qui au premier plan prend une valeur de couleur. Par exemple, si la valeur de **border-color** ou **outline-color** n'a pas été définie explicitement pour l'élément, la valeur de **color** sera utilisée à la place. Voyons un exemple:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Setting Foreground Color for Elements</title>
<style>
  p.one {
    color: #0000ff;
    border: 2px solid;
  }
  p.two {
    color: #00ff00;
    outline: 2px solid;
  }
</style>
</head>
<body>
  <p class="one">The border color of this paragraph is same as the element's text color.</p>
  <p class="two">The outline color of this paragraph is same as the element's text color.</p>
</body>
</html>

```

6. Arrière-plan CSS

L'arrière-plan joue un rôle important dans la présentation visuelle d'une page Web. CSS fournit plusieurs propriétés pour styliser l'arrière-plan d'un élément, y compris la coloration de l'arrière-plan, le placement des images en arrière-plan et la gestion de leur positionnement, etc. Les propriétés d'arrière-plan sont **background-color**, **background-image**, **background-repeat**, **background-attachment** et **background-position**.

• Couleur de l'arrière plan

La propriété **background-color** est utilisée pour définir la couleur d'arrière-plan d'un élément. L'exemple suivant montre comment définir la couleur d'arrière-plan de la page entière.

```

<head>
<meta charset="utf-8">
<title>Setting background-color of an Element in CSS</title>
<style>
  body {
    background-color: #f0e68c;
  }
  h1 {
    background-color: #46c79c;
  }
</style>
</head>
<body>
  <h1>Background Color Demo</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc neque. In hac habitasse platea dictumst. Pellentesque ultrices. Donec nunc metus, aliquam sit amet, aliquam in, luctus ac, odio. Aenean orci velit, elementum ac, egestas pellentesque, facilisis non, turpis. Quisque dolor. Vestibulum in massa in lectus hendrerit condimentum.
</p>
</body>
</html>

```

- **Image d'arrière-plan**

La propriété **background-image** définit une image comme arrière-plan d'un élément HTML. L'exemple suivant définit une image d'arrière-plan pour toute la page.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting background-image in CSS</title>
<style>
  body {
    background-image: url("/examples/images/tile.png");
  }
</style>
</head>
<body>
  <h1>Background Image Demo</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc neque. In hac habitasse platea dictumst. Pellentesque ultrices. Donec nunc metus, aliquam sit amet, aliquam in, luctus ac, odio. Aenean orci velit, elementum ac, egestas pellentesque, facilisis non, turpis. Quisque dolor. Vestibulum in massa in lectus hendrerit condimentum.
</p>
</body>
</html>
```

- **Répétition du fond**

La propriété **background-repeat** vous permet de contrôler la façon dont une image d'arrière-plan est répétée ou mise en mosaïque à l'arrière-plan d'un élément. Vous pouvez définir une image d'arrière-plan pour qu'elle se répète verticalement (axe y), horizontalement (axe x), dans les deux directions ou dans aucune direction. L'exemple suivant montre comment définir l'arrière-plan dégradé d'une page Web en répétant l'image découpée horizontalement le long de l'axe des x.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting Horizontal background-repeat in CSS</title>
<style>
  body {
    background-image: url("/examples/images/gradient.png");
    background-repeat: repeat-x;
  }
</style>
</head>
<body>
  <h1>Background Repeat Demo</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc neque. In hac habitasse platea dictumst. Pellentesque ultrices. Donec nunc metus, aliquam sit amet, aliquam in, luctus ac, odio. Aenean orci velit, elementum ac, egestas pellentesque, facilisis non, turpis. Quisque dolor. Vestibulum in massa in lectus hendrerit condimentum.
</p>
</body>
</html>
```

De même, vous pouvez utiliser la valeur **repeat-y** pour répéter l'image d'arrière-plan verticalement le long de l'axe y ou la valeur **no-repeat** pour empêcher complètement la répétition.

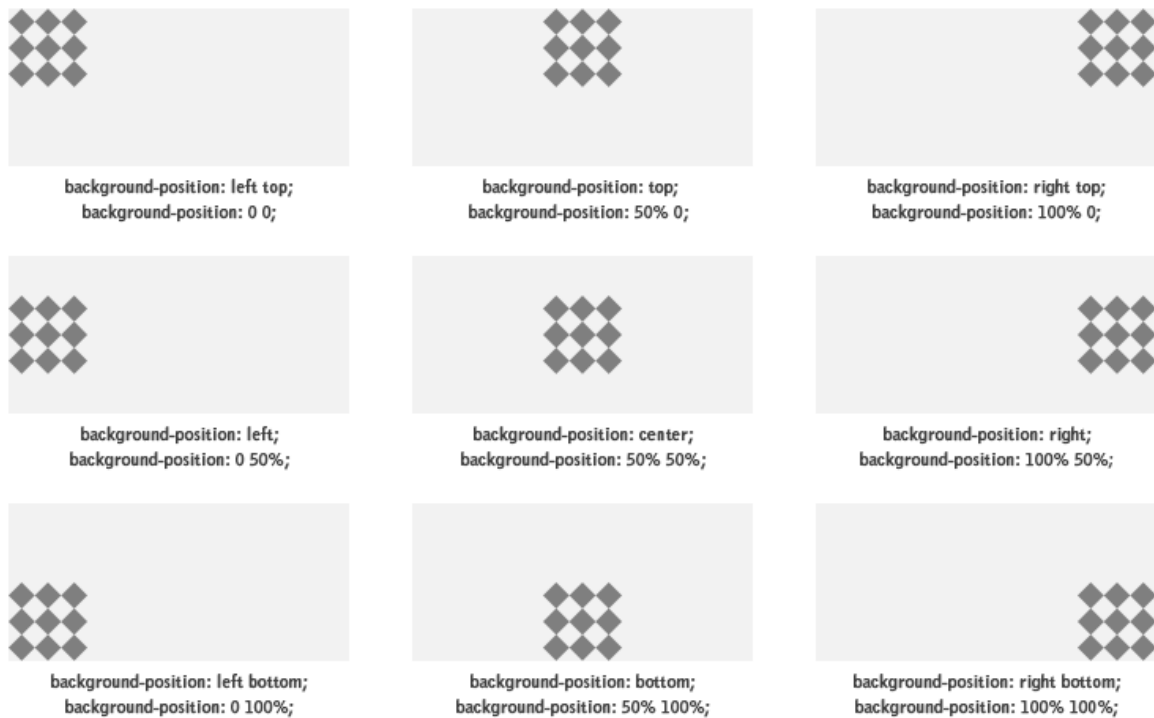
- **Position d'arrière-plan**

La propriété **background-position** est utilisée pour contrôler la position de l'image d'arrière-plan. Si aucune position d'arrière-plan n'a été spécifiée, l'image d'arrière-plan est placée à la position par défaut en haut à gauche de l'élément, c'est-à-dire à (0,0), essayons l'exemple suivant:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting Custom background-position of Images in CSS</title>
<style>
  body {
    background-image: url("/examples/images/robot.png");
    background-repeat: no-repeat;
    background-position: 100% top;
  }
  h1, p {
    margin-right: 200px;
  }
</style>
</head>
<body>
  <h1>Background Position Demo</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc neque. In hac habitasse
platea dictumst. Pellentesque ultrices. Donec nunc metus, aliquam sit amet, aliquam in,
luctus ac, odio. Aenean orci velit, elementum ac, egestas pellentesque, facilisis non,
turpis. Quisque dolor. Vestibulum in massa in lectus hendrerit condimentum. Duis suscipit,
ectus varius fermentum fermentum, est lectus facilisis metus, ac ornare risus ipsum at
sem.</p>
</body>
</html>
```

Si deux valeurs sont spécifiées pour la propriété background-position, la première valeur représente la position horizontale et la seconde la position verticale. Si une seule valeur est spécifiée, la deuxième valeur est supposée être centrale.

Outre les mots-clés, vous pouvez également utiliser des valeurs de pourcentage ou de longueur, telles que **px** ou **em** pour cette propriété.



Faisons un coup d'œil à l'illustration suivante pour comprendre comment cette propriété fonctionne réellement.

- **Attachement en arrière-plan**

La propriété **background-attachment** détermine si l'image d'arrière-plan est fixe par rapport à la fenêtre ou défile avec le bloc conteneur. Essayons l'exemple suivant pour comprendre son fonctionnement:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Fixing background-attachment in CSS</title>
<style>
  body {
    background-image: url("/examples/images/bell.png");
    background-repeat: no-repeat;
    background-position: center;
    background-attachment: fixed;
  }
</style>
</head>
<body>
  <h1>Background Attachment Demo</h1>
  <p><strong>Tip:</strong> Scroll down the page to see how background-attachment
property works.</p>
  <br>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc neque. In hac habitasse
platea dictumst. Pellentesque ultrices. Donec nunc metus, aliquam sit amet, aliquam in,
luctus ac, odio. Aenean orci velit, elementum ac, egestas pellentesque, facilisis non,
turpis. Quisque dolor. Vestibulum in massa in lectus hendrerit condimentum.</p>
```



```

<br><br><br><br><br><br><br><br><br><br><br><br>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc neque. In hac habitasse
platea dictumst. Pellentesque ultrices. Donec nunc metus, aliquam sit amet, aliquam in,
luctus ac, odio. Aenean orci velit, elementum ac, egestas pellentesque, facilisis non,
turpis. Quisque dolor. Vestibulum in massa in lectus hendrerit condimentum.</p>
<br><br><br><br><br><br><br><br><br><br><br><br>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc neque. In hac habitasse
platea dictumst. Pellentesque ultrices. Donec nunc metus, aliquam sit amet, aliquam in,
luctus ac, odio. Aenean orci velit, elementum ac, egestas pellentesque, facilisis non,
turpis. Quisque dolor. Vestibulum in massa in lectus hendrerit condimentum.</p>
<br><br><br><br><br><br><br><br><br><br><br><br>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc neque. In hac habitasse
platea dictumst. Pellentesque ultrices. Donec nunc metus, aliquam sit amet, aliquam in,
luctus ac, odio. Aenean orci velit, elementum ac, egestas pellentesque, facilisis non,
turpis. Quisque dolor. Vestibulum in massa in lectus hendrerit condimentum.</p>
</body>
</html>

```

• La propriété de raccourci d'arrière-plan

Comme vous pouvez le voir dans les exemples ci-dessus, il existe de nombreuses propriétés à prendre en compte lors du traitement des arrière-plans. Cependant, il est également possible de spécifier toutes ces propriétés dans une seule propriété pour raccourcir le code ou éviter une saisie supplémentaire. C'est ce qu'on appelle une propriété Shorthand.

La propriété **background** est une propriété abrégée permettant de définir toutes les propriétés d'arrière-plan individuelles, c'est-à-dire **background-color**, **background-image**, **background-repeat**, **background-attachment** et la propriété **background-position** à la fois. Voyons comment cela fonctionne:

<pre>background-color: #f0e68c; background-image: url("images/smiley.png"); background-repeat: no-repeat; background-attachment: fixed; background-position: 250px 25px;</pre>	<pre>background: #f0e68c url("images/smiley.png") no-repeat fixed 250px 25px;</pre>
--	---

Lors de l'utilisation de la propriété **shorthand** d'arrière-plan, l'ordre des valeurs de propriété doit être: **background: color image repeat attachment position**;

7. Styliser les polices avec CSS

Le choix de la bonne police et du bon style est très important pour la lisibilité du texte sur une page. CSS fournit plusieurs propriétés pour styliser la police du texte. Les propriétés de police sont les suivantes: **font-family**, **font-style**, **font-weight**, **font-size** et **font-variant**.

• Famille de polices

La propriété **font-family** est utilisée pour spécifier la police à utiliser pour rendre le texte. Cette propriété peut contenir plusieurs noms de polices séparés par des virgules en tant que système de secours, de sorte que si la première police n'est pas disponible sur le système de l'utilisateur, le navigateur essaie d'utiliser la seconde, et ainsi de suite.

Vous devriez terminer la liste avec une famille de polices générique composée de cinq - **serif**, **sans-serif**, **monospace**, **cursive** et **fantasy**.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting font-family in CSS</title>
<style>
  body {
    font-family: Arial, Helvetica, sans-serif;
  }
</style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

Si le nom d'une famille de polices contient plus d'un mot, il doit être placé entre guillemets, comme **"Times New Roman"**, **"Courier New"**, **"Segoe UI"**, etc.

- **Le style de police**

La propriété **font-style** est utilisée pour définir le style de police pour le contenu textuel d'un élément. Le style de police peut être **normal**, **italic** ou **oblique**. La valeur par défaut est **normal**. Essayons l'exemple suivant pour comprendre son fonctionnement:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting font-style in CSS</title>
<style>
  p.normal {
    font-style: normal;
  }
  p.italic {
    font-style: italic;
  }
  p.oblique {
    font-style: oblique;
  }
</style>
</head>
<body>
  <p class="normal">This is a normal paragraph.</p>
  <p class="italic">This is a paragraph with italic font style.</p>
  <p class="oblique">This is a paragraph with oblique font style.</p>
</body>
</html>
```

- **Taille de police**

La propriété **font-size** est utilisée pour définir la taille de la police pour le contenu textuel d'un élément. Il existe plusieurs façons de spécifier les valeurs de taille de police, par exemple avec mots-clés, pourcentage, pixels, ems, etc.

CSS fournit plusieurs mots-clés que vous pouvez utiliser pour définir les tailles de police. Une taille de police absolue peut être spécifiée à l'aide de l'un des mots clés suivants: **xx-small**, **x-small**, **small**, **medium**, **large**, **x-large**, **xx-large**. Alors qu'une taille de police relative peut être spécifiée à l'aide des mots-clés: **smaller** ou **larger**.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Defining Font Size Using Keywords</title>
<style>
  body {
    font-size: large;
  }
  h1 {
    font-size: larger;
  }
  p {
    font-size: smaller;
  }
</style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

Les valeurs de taille de police pour les autres sont comme suit :

font-size: 24px;

font-size: 2em;

font-size: 62.5%;

Le mot-clé **medium** équivaut à la taille de police par défaut du navigateur, qui est normalement de 16 px. De même, **xx-small** équivaut à 9 pixels, **x-small** à 10 pixels, **small** à 13 pixels, **large** à 18 pixels, **x-large** à 24 pixels et **xx-large** à 32 pixels. (Pour plus des détails voir <https://www.tutorialrepublic.com/css-tutorial/css-fonts.php>)

• Poids de la police

La propriété **font-weight** spécifie l'épaisseur ou le caractère gras de la police. Cette propriété peut prendre l'une des valeurs suivantes: **normal**, **bold**, **bolder**, **lighter**.

```
p { font-weight: bold; }
```

• Variante de police

La propriété **font-variant** permet d'afficher le texte dans une variante spéciale en petites majuscules.

```
p { font-variant: small-caps; }
```

8. Formatage du texte avec CSS

CSS fournit plusieurs propriétés qui vous permettent de définir très facilement et efficacement divers styles de texte tels que la couleur, l'alignement, l'espacement, la décoration, la transformation, etc. Les propriétés de texte couramment utilisées sont: **text-align**, **text-**

decoration, text-transform, text-indent, line-height, letter-spacing, word-spacing, etc. Ces propriétés vous donnent un contrôle précis sur l'apparence visuelle des caractères, des mots, des espaces, etc.

• Alignement du texte

La propriété **text-align** est utilisée pour définir l'alignement horizontal du texte. Le texte peut être aligné de quatre manières: à gauche, à droite, au centre ou justifié (marges droites gauche et droite). Faisons un coup d'œil à un exemple pour comprendre le fonctionnement de cette propriété.

<pre> <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Setting Text Alignment using CSS</title> <style> h1 { text-align: center; } p { width: 300px; text-align: justify; } </style> </head> <body> <h1>This is a heading</h1> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis.</p> </body> </html> </pre>			
<p>Alice opened the door and found that it led into a small passage, not much larger than a rat hole: she knelt down and looked along the passage into the loveliest garden you ever saw.</p>	<p>Alice opened the door and found that it led into a small passage, not much larger than a rat hole: she knelt down and looked along the passage into the loveliest garden you ever saw.</p>	<p>Alice opened the door and found that it led into a small passage, not much larger than a rat hole: she knelt down and looked along the passage into the loveliest garden you ever saw.</p>	<p>Alice opened the door and found that it led into a small passage, not much larger than a rat hole: she knelt down and looked along the passage into the loveliest garden you ever saw.</p>
left	center	right	justify

• Décoration de texte

La propriété **text-decoration** est utilisée pour définir ou supprimer des décorations du texte. Cette propriété accepte généralement l'une des valeurs suivantes: **underline**, **overline**, **line-through**, et **none**. Vous devez éviter de souligner le texte qui n'est pas un lien, car cela pourrait dérouter le visiteur.

<pre> <!DOCTYPE html> <html lang="en"> <head> </pre>
--

```

<meta charset="utf-8">
<title>Setting Text Decoration using CSS</title>
<style>
  h1 {
    text-decoration:overline;
  }
  h2 {
    text-decoration:line-through;
  }
  h3 {
    text-decoration:underline;
  }
</style>
</head>
<body>
  <h1>This is heading 1</h1>
  <h2>This is heading 2</h2>
  <h3>This is heading 3</h3>
</body>
</html>

```

La propriété **text-decoration** est largement utilisée pour supprimer le soulignement par défaut des liens hypertexte HTML. Vous pouvez également fournir d'autres indices visuels pour le faire ressortir du texte normal, par exemple, en utilisant une bordure en pointillé au lieu d'un soulignement solide.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Removing the Default Underline from HTML Links</title>
<style>
  a {
    text-decoration: none;
    border-bottom: 1px dotted;
  }
  a:hover {
    border-bottom: none;
  }
</style>
</head>
<body>
  <p>Place your mouse pointer <a href="#">over me!</a></p>
</body>
</html>

```

• Transformation de texte

La propriété **text-transform** est utilisée pour définir les cas d'un texte. En utilisant cette propriété, vous pouvez modifier le contenu textuel d'un élément en majuscules ou minuscules, ou mettre en majuscule la première lettre de chaque mot sans modifier le texte d'origine.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">

```

```

<title>Setting Text Transformation using CSS</title>
<style>
  h1 {
    text-transform: uppercase;
  }
  h2 {
    text-transform: capitalize;
  }
  h3 {
    text-transform: lowercase;
  }
</style>
</head>
<body>
  <h1>This is heading 1</h1>
  <h2>This is heading 2</h2>
  <h3>This is heading 3</h3>
</body>
</html>

```

- **Indentation du texte**

La propriété **text-indent** est utilisée pour définir l'indentation de la première ligne de texte dans un bloc de texte. Cela se fait généralement en insérant l'espace vide avant la première ligne de texte. La taille de l'indentation peut être spécifiée à l'aide du pourcentage (%), des valeurs de longueur en pixels, ems, etc.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting Text Indentation using CSS</title>
<style>
  p {
    text-indent: 100px;
  }
</style>
</head>
<body>
  <h1>CSS Text Indentation</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius
quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante.
Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non
aliquet sagittis.</p>
</body>
</html>

```

- **Espacement des lettres**

La propriété **letter-spacing** est utilisée pour définir un espacement supplémentaire entre les caractères du texte. Cette propriété peut prendre une valeur de longueur en pixels, ems, etc.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting Letter Spacing using CSS</title>

```

```

<style>
  h1 {
    letter-spacing: -3px;
  }
  p {
    letter-spacing: 10px;
  }
</style>
</head>
<body>
  <h1>This is a heading.</h1>
  <p>This is a simple paragraph of text.</p>
</body>
</html>

```

• Espacement des mots

La propriété **word-spacing** est utilisée pour spécifier un espacement supplémentaire entre les mots. Cette propriété peut accepter une valeur de longueur en pixels, ems, etc. Les valeurs négatives sont également autorisées.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting word spacing using CSS</title>
<style>
  p.one {
    word-spacing: 20px;
  }
  p.two {
    width: 150px;
    word-spacing: 20px;
    text-align: justify;
  }
  p.three {
    word-spacing: 20px;
    white-space: pre;
  }
</style>
</head>
<body>
  <p class="one">This is a normal paragraph.</p>
  <hr>
  <p class="two">Note that spacing between the words of this paragraph are varying in order
to justify the text even if the value of word-spacing property is set to 20px.</p>
  <hr>
  <p class="three">Note that spacing between the words
    of this paragraph are higher than the normal spacing
    even if whitespace are preserved.</p>
</body>
</html>

```

• Interligne

La propriété **line-height** est utilisée pour définir la hauteur de la ligne de texte. Il est également appelé interligne et couramment utilisé pour définir la distance entre les lignes de texte. La valeur de cette propriété peut être un nombre, un pourcentage (%) ou une longueur en pixels, ems, etc.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting Line Height using CSS</title>
<style>
  p {
    line-height: 1.2;
    border: 1px solid #00ff00;
  }
  div {
    line-height: 200%;
    border: 1px solid #ff0000;
  }
</style>
</head>
<body>
  <h1>Change the values to see how it works</h1>
  <p>The <code>line-height</code> property sets the height between lines of text.<br> The
line height of this paragraph is specified using number.</p>
  <div>The <code>line-height</code> property sets the height between lines of text.<br> The
line height of this paragraph is specified using percentage.</div>
</body>
</html>
```

9. Styler les hyperliens avec CSS

Un lien ou hyperlien a quatre états différents: lien, visité, actif et survolé. Ces quatre états d'un lien peuvent être stylisés différemment en utilisant les sélecteurs de pseudo-classes d'ancrage suivants.

a:link - définissez des styles pour les liens normaux ou non consultés.

a:visited - définir des styles pour les liens que l'utilisateur a déjà visités.

a:hover - définissez les styles d'un lien lorsque l'utilisateur place le pointeur de la souris dessus.

a:active - définissez les styles des liens lorsque l'utilisateur clique dessus.

Vous pouvez spécifier n'importe quelle propriété CSS que vous souhaitez, par exemple couleur, police, arrière-plan, bordure, etc. à chacun de ces sélecteurs pour personnaliser le style des liens, tout comme vous le faites avec le texte normal.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Styling Different Link States using CSS</title>
<style>
  /* unvisited link */
  a:link {
    color: #ff0000;
```



```

    text-decoration: none;
    border-bottom: 1px solid;
}
/* visited link */
a:visited {
    color: #ff00ff;
}
/* mouse over link */
a:hover {
    color: #00ff00;
    border-bottom: none;
}
/* active link */
a:active {
    color: #00ffff;
}
</style>
</head>
<body>
  <p><a href="https://www.tutorialrepublic.com/" target="_top">Visit Tutorial
  Republic</a></p>
</body>
</html>

```

l'ordre des pseudo-classes doit être le suivant -: **:link**, **:visited**, **:hover**, **:active**, **:focus** pour que celles-ci fonctionnent correctement.

• Modification des styles de lien standard

Par défaut, les liens texte apparaîtront comme suit dans la plupart des navigateurs:

Un lien non visité sous forme de texte bleu souligné.

Un lien visité sous forme de texte violet souligné.

Un lien actif sous forme de texte rouge souligné.

Voyons maintenant comment personnaliser les liens en remplaçant leur style par défaut.

Utilisez simplement la propriété **color** pour définir la couleur de votre choix pour différents états d'un lien.

```

a:link { color: #1ebba3; }
a:visited { color: #ff00f4; }
a:hover { color: #a766ff; }
a:active { color: #ff9800; }

```

• Suppression du soulignement par défaut des liens

Si vous n'aimez pas le soulignement par défaut sur les liens, vous pouvez simplement utiliser la propriété **text-decoration** pour vous en débarrasser. Vous pouvez également appliquer d'autres styles aux liens tels que la couleur d'arrière-plan, la bordure inférieure, la police en gras, etc. pour le faire ressortir un peu mieux du texte normal.

```

a:link, a:visited { text-decoration: none; }
a:hover, a:active { text-decoration: underline; }

```

- **Faire ressembler les liens de texte à des boutons**

Vous pouvez également faire en sorte que vos liens de texte ordinaires ressemblent à un bouton en utilisant CSS. Pour ce faire, nous devons utiliser quelques propriétés CSS supplémentaires telles que la couleur d'arrière-plan, la bordure, l'affichage, le remplissage, etc.

```
a:link, a:visited {
    color: white;
    background-color: #1ebba3;
    display: inline-block;
    padding: 10px 20px;
    border: 2px solid #099983;
    text-decoration: none;
    text-align: center;
    font: 14px Arial, sans-serif;
}
a:hover, a:active {
    background-color: #9c6ae1;
    border-color: #7443b6;
}
```

10. Styliser les listes avec CSS

- **Modification du type de marqueur des listes**

Par défaut, les éléments d'une liste ordonnée sont numérotés avec des chiffres arabes (1, 2, 3, 5, etc.), tandis que dans une liste non ordonnée, les éléments sont marqués par des puces rondes (•). Mais, vous pouvez changer ce type de marqueur de liste par défaut en tout autre type tel que les chiffres romains, les lettres latines, le cercle, le carré, etc. en utilisant la propriété **list-style-type**.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting Numbering or Bullet Point Style of Lists</title>
<style>
    ul {
        list-style-type: square;
    }
    ol {
        list-style-type: upper-roman;
    }
</style>
</head>
<body>
    <h2>Unordered List</h2>
    <ul>
        <li>List Item 1</li>
        <li>List Item 2</li>
```

```

    <li>List Item 3</li>
  </ul>
  <h2>Ordered List</h2>
  <ol>
    <li>List Item 1</li>
    <li>List Item 2</li>
    <li>List Item 3</li>
  </ol>
</body>
</html>

```

• Modification de la position des marqueurs de liste

Par défaut, les marqueurs de chaque élément de liste sont positionnés en dehors de leurs boîtes d'affichage. Cependant, vous pouvez également positionner ces marqueurs ou puces à l'intérieur des zones d'affichage de l'élément de liste à l'aide de la propriété **list-style-position** avec la valeur à l'intérieur. Dans ce cas, les lignes s'enrouleront sous le marqueur au lieu d'être indentées. Voici un exemple:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting the Position of List Marker</title>
<style>
  body{
    font-size: 14px;
    font-family: Arial,sans-serif;
  }
  ol li {
    background: #ddd;
    padding: 5px;
    margin: 5px;
  }
  ol.in li {
    list-style-position: inside;
  }
  ol.out li {
    list-style-position: outside;
  }
</style>
</head>
<body>
  <h2>List Marker Position - Inside</h2>
  <ol class="in">
    <li>Fasten your seatbelt</li>
    <li>Start the car's engine and take a closer look the instrument cluster for any warning sign</li>
    <li>Look around carefully and go</li>
  </ol>
  <h2>List Marker Position - Outside</h2>
  <ol class="out">
    <li>Fasten your seatbelt</li>
    <li>Start the car's engine and take a closer look the instrument cluster for any warning sign</li>
    <li>Look around carefully and go</li>
  </ol>
</body>
</html>

```

• Utilisation d'images comme marqueurs de liste

Vous pouvez également définir une image comme marqueur de liste à l'aide de la propriété **list-style-image**. La règle de style de l'exemple suivant affecte une image PNG transparente "arrow.png" comme marqueur de liste pour tous les éléments de la liste non ordonnée. Essayons-le et voyons comment cela fonctionne:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting an Image as List Marker with CSS</title>
<style>
  ul li {
    list-style-image: url("/examples/images/bullet.png");
    margin: 5px;
  }
</style>
</head>
<body>
  <h2>Using Image as Bullet Point</h2>
  <ul>
    <li>List Item 1</li>
    <li>List Item 2</li>
    <li>List Item 3</li>
  </ul>
</body>
</html>
```

• Définition de toutes les propriétés de la liste à la fois

La propriété **list-style** est une propriété abrégée permettant de définir les trois propriétés **list-style-type**, **list-style-image** et **list-style-position** d'une liste en un seul endroit. La règle de style suivante définit toutes les propriétés de la liste dans une seule déclaration.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>CSS list-style Shorthand Property</title>
<style>
  ul {
    list-style: square inside url("/examples/images/bullet.png");
    background: #9dddef;
  }
  ul li {
    background: #ededed;
    margin: 5px 0;
    padding: 5px;
  }
</style>
</head>
<body>
  <h2>The <code>list-style</code> Shorthand Property</h2>
  <ul>
    <li>List Item 1</li>
```

```

    <li>List Item 2</li>
    <li>List Item 3</li>
  </ul>
</body>
</html>

```

Lors de l'utilisation de la propriété abrégée de style liste, l'ordre des valeurs est : **list-style-type | list-style-position | list-style-image**. Peu importe si l'une des valeurs ci-dessus est manquante tant que les autres sont dans l'ordre spécifié.

• Création de menus de navigation à l'aide de listes

Les listes HTML sont fréquemment utilisées pour créer une barre de navigation horizontale ou un menu qui apparaît généralement en haut d'un site Web. Mais comme les éléments de la liste sont des éléments de bloc, nous devons donc utiliser la propriété d'affichage CSS pour les afficher en ligne. L'exemple montre comment cela fonctionne:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Building Navigation Bar with HTML List and CSS</title>
<style>
  body{
    font-size: 14px;
    font-family: Arial,sans-serif;
  }
  ul {
    padding: 0;
    list-style: none;
    background: #f2f2f2;
  }
  ul li {
    display: inline-block;
  }
  ul li a {
    display: block;
    padding: 10px 25px;
    color: #333;
    text-decoration: none;
  }
  ul li a:hover {
    color: #fff;
    background: #939393;
  }
</style>
</head>
<body>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About Us</a></li>
      <li><a href="#">Products</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Portfolio</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
</body>

```

```

    </ul>
  </nav>
  <p><strong>Note:</strong> Place mouse pointer over the menu link to see the hover
effect.</p>
</body>

```

11. Styliser les tableaux avec CSS

Les tableaux sont généralement utilisés pour afficher des données tabulaires. Mais lorsque vous créez un tableau HTML sans style ni attribut, les navigateurs les affichent sans aucune bordure. Avec CSS, vous pouvez grandement améliorer l'apparence de vos tableaux. CSS fournit plusieurs propriétés qui vous permettent de contrôler la mise en page et la présentation des éléments du tableau. Dans la section suivante, vous verrez comment utiliser CSS pour créer des tableaux élégants et cohérents.

- **Ajout de bordures aux tableaux**

La propriété **border** est le meilleur moyen de définir les bordures des tables. L'exemple suivant définira une bordure noire pour les éléments **<table>**, **<th>** et **<td>**.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of Setting Table Borders</title>
<style>
  table, th, td {
    border: 1px solid black;
  }
</style>
</head>
<body>
  <table>
    <tr>
      <th>ID</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
    <tr>
      <td>1</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Rambo</td>
      <td>johnrambo@mail.com</td>
    </tr>
    <tr>
      <td>4</td>
      <td>Harry</td>
      <td>Potter</td>
      <td>harrypotter@mail.com</td>
    </tr>
  </table>
</body>
</html>

```

- **Réduire les bordures du tableau**

Il existe deux modèles distincts pour définir les bordures des cellules d'un tableau en CSS: séparer et réduire.

Dans le modèle de bordure séparée, qui est la valeur par défaut, chaque cellule de tableau a ses propres bordures distinctes, tandis que dans le modèle de bordure réduite, les cellules de tableau adjacentes partagent une bordure commune. Vous pouvez définir le modèle de bordure d'un tableau HTML à l'aide de la propriété **border-collapse**. Les règles de style suivantes réduiront les bordures des cellules du tableau et appliqueront une bordure noire d'un pixel.

```
table { border-collapse: collapse; }
th, td { border: 1px solid black; }
```

Vous pouvez également supprimer l'espace entre les bordures des cellules du tableau en définissant la valeur de la propriété **border-spacing** sur 0. Cependant, cela supprime uniquement l'espace mais ne fusionne pas les bordures comme lorsque vous définissez le border-collapse sur réduire.

- **Ajustement de l'espace à l'intérieur des tableaux**

Par défaut, le navigateur crée les cellules du tableau juste assez grandes pour contenir les données dans les cellules. Pour ajouter plus d'espace entre le contenu de la cellule du tableau et les bordures de cellule, vous pouvez simplement utiliser la propriété **padding**. Comme dans l'exemple suivant:

```
table { border-collapse: collapse; }
th, td { border: 1px solid black; }
th, td { padding: 15px; }
```

Vous pouvez également ajuster l'espacement entre les bordures des cellules à l'aide de la propriété **border-spacing**, si les bordures de votre tableau sont séparées (ce qui est par défaut).

```
table { border-spacing: 10px; }
th, td { border: 1px solid black; }
th, td { padding: 15px; }
```

- **Définition de la largeur et de la hauteur de la table**

Par défaut, un tableau sera rendu juste assez large et haut pour contenir tout son contenu. Cependant, vous pouvez également définir la largeur et la hauteur du tableau ainsi que ses cellules explicitement à l'aide de la propriété **width** et **height**. Les règles de style de l'exemple suivant définissent la largeur du tableau sur 100% et la hauteur des cellules d'en-tête du tableau sur 40 pixels.

```
table { width: 100%; }
th { height: 40px; }
```

- **Contrôle de la disposition du tableau**

Une table se développe et se contracte pour accueillir les données qu'elle contient. Ceci est le comportement par défaut. Au fur et à mesure que les données se remplissent à l'intérieur du tableau, elles continuent à se développer tant qu'il y a de l'espace. Parfois, cependant, il est

nécessaire de définir une largeur fixe pour le tableau afin de gérer la mise en page. Vous pouvez le faire à l'aide de la propriété **table-layout**. Cette propriété définit l'algorithme à utiliser pour mettre en page les cellules, les lignes et les colonnes du tableau. Cette propriété prend l'une des deux valeurs suivantes:

auto - les largeurs du tableau et de ses cellules sont ajustées pour s'adapter au contenu. Ceci est la valeur par défaut.

fixed - la disposition horizontale du tableau ne dépend pas du contenu des cellules; cela ne dépend que de la largeur du tableau, de la largeur des colonnes et des bordures ou de l'espacement des cellules.

Les règles de style de l'exemple suivant spécifient que le tableau HTML est mis en page à l'aide de la valeur fixe et a une largeur fixe de 300 pixels.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of CSS table-layout property</title>
<style>
  table {
    width: 250px;
    border-collapse: separate;
  }
  table, tr, th, td{
    border: 1px solid #000000;
  }
  .auto {
    table-layout: auto;
  }
  .fixed {
    table-layout: fixed;
  }
  td{
    width: 50%;
  }
</style>
</head>
<body>
  <table class="auto">
    <caption>Example 1. Auto</caption>
    <tr>
      <th>Name</th>
      <td>John Carter</td>
    </tr>
    <tr>
      <th>Email</th>
      <td>johncarter@mail.com</td>
    </tr>
  </table>
  <br>
  <table class="fixed">
    <caption>Example 2. Fixed</caption>
    <tr>
      <th>Name</th>
      <td>Peter Parker</td>
    </tr>
    <tr>
      <th>Email</th>
```



```

        <td>peterparker@mail.com</td>
    </tr>
</table>
    <p><strong>Note:</strong> You can see the width of table cell does not change to accommodate
the content in fixed table-layout.</p>
</body>
</html>

```

• Alignement du texte à l'intérieur des cellules du tableau

Vous pouvez aligner le contenu du texte à l'intérieur des cellules du tableau horizontalement ou verticalement.

Pour l'alignement horizontal du texte à l'intérieur des cellules du tableau, vous pouvez utiliser la propriété **text-align** de la même manière que vous utilisez avec d'autres éléments. Vous alignez le texte à gauche, à droite, au centre ou à justifier (**left**, **right**, **center**, **justify**). Les règles de style suivantes aligneront à gauche le texte à l'intérieur des éléments **<th>**.

```

table { width: 100%;
        border-collapse: collapse;
}
th, td { border: 1px solid black; }
th, td { padding: 15px;
        border: 1px solid black;
}
th { text-align: left; }

```

Le texte à l'intérieur des éléments **<td>** est aligné à gauche par défaut, tandis que le texte à l'intérieur des éléments **<th>** est aligné au centre et rendu en gras par défaut.

De même, vous pouvez aligner verticalement le contenu à l'intérieur des éléments **<th>** et **<td>** vers le haut, le bas ou le milieu (**top**, **bottom**, **middle**) à l'aide de la propriété **vertical-align**. L'alignement vertical par défaut est au milieu. Les règles de style suivantes aligneront verticalement le texte en bas à l'intérieur des éléments **<th>**.

```

table { width: 100%;
        border-collapse: collapse;
}
th, td { border: 1px solid black; }
th, td { padding: 15px;
        border: 1px solid black;
}
th { height: 40px;
        vertical-align: bottom;
}

```

- **Contrôle de la position de la légende du tableau**

Vous pouvez définir la position verticale d'une légende de tableau à l'aide de la propriété **caption-side**. La légende peut être placée en haut (**top**) ou en bas (**bottom**) du tableau. La position par défaut est **top**.

```
table, td, th {  
    border: 1px solid gray;  
}  
  
caption {  
    caption-side: bottom;  
}
```

- **Gestion des cellules vides**

Dans les tableaux qui utilisent un modèle de bordure distinct, qui est par défaut, vous pouvez également contrôler le rendu des cellules qui n'ont pas de contenu visible à l'aide de la propriété **empty-cells**. Cette propriété accepte une valeur de **show** ou **hide**. La valeur par défaut est **show**, qui rend les cellules vides comme les cellules normales, mais si la valeur **hide** est spécifiée, aucune bordure ou arrière-plan n'est dessiné autour des cellules vides. Essayons un exemple pour comprendre comment cela fonctionne vraiment:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="utf-8">  
<title>Example of CSS empty-cells property</title>  
<style>  
    table {  
        width: 300px;  
        border-collapse: separate;  
    }  
    table, th, td {  
        border: 1px solid #000000;  
    }  
    table.empty-show {  
        empty-cells: show;  
    }  
    table.empty-hide {  
        empty-cells: hide;  
    }  
</style>  
</head>  
<body>  
    <h2>Table with Empty-cells</h2>  
    <table class="empty-show">  
        <tr>  
            <th>Name</th>  
            <td>John Carter</td>  
        </tr>  
        <tr>  
            <th>Email</th>  
            <td></td>
```

```

    </tr>
  </table>
  <br>
  <h2>Table with Hidden Empty-cells</h2>
  <table class="empty-hide">
    <tr>
      <th>Name</th>
      <td>Peter Parker</td>
    </tr>
    <tr>
      <th>Email</th>
      <td></td>
    </tr>
  </table>
  <p><strong>Note:</strong> You can see the empty cell in the second example table is
  completely hidden. Non-breaking space (i.e. &#amp;nbsp;) is considered as content.</p>
</body>
</html>

```

• Création de tableaux zébrés

La définition de différentes couleurs d'arrière-plan pour les lignes alternatives est une technique courante pour améliorer la lisibilité des tables contenant une grande quantité de données. C'est ce qu'on appelle communément le zébrage d'une table. Vous pouvez simplement obtenir cet effet en utilisant le sélecteur de pseudo-classe : **nth-child ()**. Les règles de style suivantes mettront en évidence toutes les lignes impaires dans le corps du tableau.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of Creating Zebra-striped Tables</title>
<style>
  table {
    width: 100%;
    font-family: arial, sans-serif;
    border-collapse: collapse;
  }
  th, td {
    padding: 8px;
    text-align: left;
    border-top: 1px solid #dee2e6;
  }
  tbody tr:nth-child(odd) {
    background-color: #f2f2f2;
  }
</style>
</head>
<body>
  <table>
    <thead>
      <tr>
        <th>Row</th>
        <th>First Name</th>

```

```

        <th>Last Name</th>
        <th>Email</th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
    </tr>
    <tr>
        <td>2</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
    <tr>
        <td>3</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
    </tr>
</tbody>
</table>
</body>
</html>

```

La pseudo-classe: `nth-child()` sélectionne des éléments en fonction de leur position dans un groupe de frères et sœurs. Il peut prendre un nombre, un mot-clé pair ou impair, ou une expression de la forme $xn + y$ où x et y sont des entiers (par exemple $1n$, $2n$, $2n + 1$, ...) comme argument.

• Rendre une table réactive

Les tableaux ne sont pas de nature réactive. Cependant, pour prendre en charge les appareils mobiles, vous pouvez ajouter de la réactivité à vos tableaux en activant le défilement horizontal sur les petits écrans. Pour ce faire, enveloppez simplement votre tableau avec un élément `<div>` et appliquez le style **overflow-x: auto**; comme indiqué ci-dessous:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of Creating Responsive Tables</title>
<style>
    table {
        width: 100%;
        border-collapse: collapse;
    }
    th, td {
        padding: 8px;
        text-align: left;
        border: 1px solid #dee2e6;
        white-space: nowrap; /* to prevent text wrapping */
    }
    .responsive-table {
        overflow-x: auto;
    }
</style>
</head>
<body>

```

```

<div class="responsive-table">
  <table>
    <thead>
      <tr>
        <th>ID</th>
        <th>Supplier</th>
        <th>Contact Name</th>
        <th>Address</th>
        <th>City</th>
        <th>Postal Code</th>
        <th>Country</th>
        <th>Phone</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Exotic Liquids</td>
        <td>Charlotte Cooper</td>
        <td>49 Gilbert St.</td>
        <td>London</td>
        <td>EC1 4SD</td>
        <td>UK</td>
        <td>(171) 555-2222</td>
      </tr>
      <tr>
        <td>2</td>
        <td>New Orleans Cajun Delights</td>
        <td>Shelley Burke</td>
        <td>P.O. Box 78934</td>
        <td>New Orleans</td>
        <td>70117</td>
        <td>USA</td>
        <td>(100) 555-4822</td>
      </tr>
      <tr>
        <td>3</td>
        <td>Grandma Kellys Homestead</td>
        <td>Regina Murphy</td>
        <td>707 Oxford Rd.</td>
        <td>Ann Arbor</td>
        <td>48104</td>
        <td>USA</td>
        <td>(313) 555-5735</td>
      </tr>
      <tr>
        <td>4</td>
        <td>Tokyo Traders</td>
        <td>Yoshi Nagase</td>
        <td>9-8 Sekimai Musashino-shi</td>
        <td>Tokyo</td>
        <td>100</td>
        <td>Japan</td>
        <td>(03) 3555-5011</td>
      </tr>
      <tr>
        <td>5</td>
        <td>Mayumis</td>
        <td>Mayumi Ohno</td>
        <td>92 Setsuko Chuo-ku</td>
        <td>Osaka</td>
        <td>545</td>
        <td>Japan</td>
        <td>(06) 431-7877</td>
      </tr>
    </tbody>
  </table>
</div>
<p><strong>Note:</strong> Toggle the editor layout or open the output in a blank window and resize it to understand how responsive table works.</p>
</body>
</html>

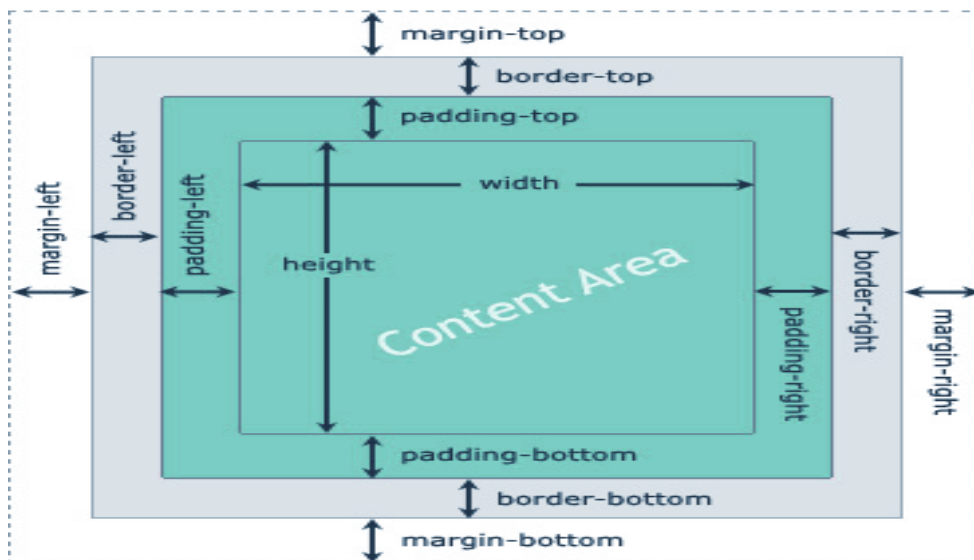
```

12. Modèle de boîte CSS

• Qu'est-ce que Box Model?

Chaque élément qui peut être affiché sur une page Web est composé d'une ou plusieurs boîtes rectangulaires. Le modèle de boîte CSS décrit généralement comment ces boîtes rectangulaires sont disposées sur une page Web. Ces boîtes peuvent avoir des propriétés différentes et peuvent interagir les unes avec les autres de différentes manières, mais chaque boîte a une **zone de contenu** et des **zones de rembourrage (padding)**, de **bordure** et de **marge** facultatives.

Le diagramme suivant montre comment les propriétés CSS **width**, **height**, **padding**, **border** et **margin** déterminent l'espace qu'un élément peut occuper sur une page Web.



Le **padding** est l'espace transparent entre le contenu de l'élément et sa bordure, tandis que **margin** est l'espace transparent autour de la bordure. De plus, si un élément a la couleur d'arrière-plan, il sera visible à travers sa zone de **padding**. La zone de **marge** reste toujours transparente, elle n'est pas affectée par la couleur d'arrière-plan de l'élément, cependant, la couleur d'arrière-plan de l'élément parent est visible à travers elle.

• Largeur et hauteur des éléments

Habituellement, lorsque vous définissez la largeur et la hauteur d'un élément à l'aide des propriétés CSS **width** et **height**, vous ne définissez en réalité que la largeur et la hauteur de la zone de contenu de cet élément. La largeur et la hauteur réelles de la boîte de l'élément dépendent de plusieurs facteurs. L'espace réel que la boîte d'un élément peut prendre sur une page Web est calculé comme suit:

Box Size	CSS Properties
Total Width	<code>width + padding-left + padding-right + border-left + border-right + margin-left + margin-right</code>
Total Height	<code>height + padding-top + padding-bottom + border-top + border-bottom + margin-top + margin-bottom</code>

Essayons maintenant l'exemple suivant pour comprendre comment le modèle de boîte fonctionne réellement:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>CSS Box Formatting Model</title>
<style>
  div {
    width: 300px;
    height: 200px;
    padding: 15px; /* set padding for all four sides */
    border: 10px solid black; /* set border for all four sides */
    /*
    margin: 20px auto; /* set top and bottom margin to 20
    pixels, and left and right margin to auto */
  }
</style>
</head>
<body>
  <div> Habituellement, lorsque vous définissez la largeur et la hauteur d'un
  élément à l'aide des propriétés CSS width et height, vous ne définissez en réalité
  que la largeur et la hauteur de la zone de contenu de cet élément. La largeur et la
  hauteur réelles de la boîte de l'élément dépendent de plusieurs facteurs. L'espace
  réel que la boîte d'un élément peut prendre sur une page Web est calculé comme
  suit:</div>
</body>
</html>
```

• Définition des dimensions des éléments

CSS a plusieurs propriétés de dimension, telles que **width**, **height**, **max-width**, **min-width**, **max-height** et **min-height** qui vous permettent de contrôler la largeur et la hauteur d'un élément. Les sections suivantes décrivent comment utiliser ces propriétés pour créer une meilleure mise en page Web.

• Réglage de la largeur et de la hauteur

La propriété **width** and **height** définit la largeur et la hauteur de la zone de contenu d'un élément. Cette largeur et cette hauteur n'incluent pas les **rembourrages**, les **bordures** ou les **marges**. Essayons l'exemple suivant et voyons comment cela fonctionne réellement:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>CSS Set Width and Height of an Element</title>
<style>
  div {
    width: 300px;
    height: 200px;
```

```
        background: #eee8aa;
    }
</style>
</head>
<body>
    <div>Play with the values to see how it works!</div>
</body>
</html>
```

Les règles de style ci-dessus attribuent une largeur fixe de 300 pixels et une hauteur de 200 pixels à l'élément <div>. Les propriétés de largeur et de hauteur peuvent prendre les valeurs suivantes:

- **length** - spécifie une largeur en px, em, rem, pt, cm, etc.
- **%** - spécifie une largeur en pourcentage (%) de la largeur de l'élément contenant.
- **auto** - le navigateur calcule une largeur appropriée pour l'élément.
- **initial** - Définit la largeur et la hauteur sur leur valeur par défaut, qui est auto.
- **inherit** - spécifie que la largeur doit être héritée de l'élément parent.

Généralement, lorsque vous créez un élément de bloc, tel que <div>, <p>, etc., le navigateur définit automatiquement sa largeur à 100% de la largeur disponible et sa hauteur à ce qui est nécessaire pour afficher tout le contenu. Vous devez éviter de définir une largeur et une hauteur fixes, sauf si cela est nécessaire.

• Définition de la largeur et de la hauteur maximales

Vous pouvez utiliser les propriétés **max-width** et **max-height** pour spécifier la largeur et la hauteur maximales de la zone de contenu. Cette largeur et hauteur maximales n'inclut pas les **rembourrages**, les **bordures** ou les **marges**.

Un élément ne peut pas être plus large que la valeur **max-width**, même si la valeur de la propriété **width** est définie sur quelque chose de plus grand. Par exemple, si la largeur est définie sur 300px et la largeur maximale sur 200px, la largeur réelle de l'élément sera de 200px. Voyons un exemple:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>CSS Set Maximum Width of an Element</title>
<style>
    div {
        width: 300px;
        max-width: 200px;
        background: #bbb3ff;
    }
    p {
```



```
    max-width: 400px;
    background: #eee8aa;
}
</style>
</head>
<body>
    <div>The maximum width of this div element is set to 200px, so it can't be
    wider than that.</div>
    <p>Enter some text to see how it works.</p>
</body>
</html>
```

De même, un élément auquel la hauteur maximale est appliquée ne sera jamais plus grand que la valeur spécifiée, même si la propriété **height** est définie sur quelque chose de plus grand. Par exemple, si la hauteur est définie sur 200px et la hauteur maximale définie sur 100px, la hauteur réelle de l'élément sera de 100px.

De la même façon, on définit la largeur et la hauteur minimales en utilisant les propriétés **min-width** et **min-height**.

- **Définition d'une plage de largeur et de hauteur**

Les propriétés **min-width** et **min-height** sont souvent utilisées en combinaison avec les propriétés **max-width** et **max-height** pour produire une plage de largeur et de hauteur pour un élément.

Cela peut être très utile pour créer une conception flexible. Dans l'exemple suivant, la largeur minimale de l'élément **<div>** serait de 300 pixels et il peut s'étendre horizontalement jusqu'à un maximum de 500 pixels.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>CSS Set Width Range for an Element</title>
<style>
    div {
        float: left;
        min-width: 300px;
        max-width: 500px;
        height: 100px;
        background: #eee8aa;
```

```

    }
</style>
</head>
<body>
    <p><strong>Note:</strong> The minimum width of the following div
    element will be 300px, and it can stretches horizontally up to 500px. Enter few
    lines of text inside the div element to understand how it works.</p>
    <div></div>
</body>
</html>

```

De même, vous pouvez définir une plage de hauteur pour un élément. Dans l'exemple ci-dessous, la hauteur minimale de l'élément `<div>` serait de 300 pixels et il peut s'étendre verticalement jusqu'à un maximum de 500 pixels.

```

div {
    min-height: 300px;
    max-height: 500px;
}

```

• Définir les rembourrages pour les côtés individuels

Vous pouvez spécifier les rembourrages pour les côtés individuels d'un élément tels que les côtés supérieur, droit, inférieur et gauche à l'aide des propriétés CSS **padding-top**, **padding-right**, **padding-bottom** et **padding-left**, respectivement. Essayons un exemple pour comprendre comment cela fonctionne:

```

h1 {
    padding-top: 50px;
    padding-bottom: 100px;
}
p {
    padding-left: 75px;
    padding-right: 75px;
}

```

• La propriété Shorthand Padding

La propriété **padding** est une propriété abrégée pour éviter de définir le padding de chaque côté séparément, c'est-à-dire **padding-top**, **padding-right**, **padding-bottom** et **padding-left**. Jetons un coup d'œil à l'exemple suivant pour comprendre comment cela fonctionne essentiellement:

```

h1 {
    padding: 50px; /* apply to all four sides */
}

```

```

}
p {
padding: 25px 75px; /* vertical | horizontal */
}
div {
padding: 25px 50px 75px; /* top | horizontal | bottom */
}
pre {
padding: 25px 50px 75px 100px; /* top | right | bottom | left
*/
}

```

Cette notation abrégée peut prendre une, deux, trois ou quatre valeurs séparées par des espaces.

- Si une valeur est spécifiée, elle est appliquée aux quatre côtés.
- Si deux valeurs sont spécifiées, la première valeur est appliquée aux côtés supérieur et inférieur, et la deuxième valeur est appliquée aux côtés droit et gauche de la boîte de l'élément.
- Si trois valeurs sont spécifiées, la première valeur est appliquée au haut, la deuxième valeur est appliquée aux côtés droit et gauche et la dernière valeur est appliquée au bas.
- Si quatre valeurs sont spécifiées, elles sont appliquées respectivement en haut, à droite, en bas et à gauche de la boîte de l'élément dans l'ordre spécifié.

Il est recommandé d'utiliser les propriétés abrégées, cela vous aidera à gagner du temps en évitant la saisie supplémentaire et à rendre votre code CSS plus facile à suivre et à maintenir.

• Effet du rembourrage et de la bordure sur la mise en page

Lors de la création de mises en page d'une page Web, l'ajout d'un padding ou d'une bordure aux éléments produit parfois un résultat inattendu, car le padding et la bordure sont ajoutés à la largeur et à la hauteur de la boîte générée par l'élément.

Par exemple, si vous définissez la largeur d'un élément `<div>` sur 100% et que vous lui appliquez également un padding ou une bordure gauche et droite, la barre de défilement horizontale apparaîtra. Voyons un exemple:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Effect of CSS Padding on Element Box Size</title>
<style>
div {
width: 100%;

```

```
padding: 25px;
background: violet;
}
</style>
</head>
<body>
  <div>
    <h1>This is a DIV Box</h1>
  </div>
  <p><strong>Notice</strong>, the scrollbar at the bottom of the
viewport.</p>
</body>
</html>
```

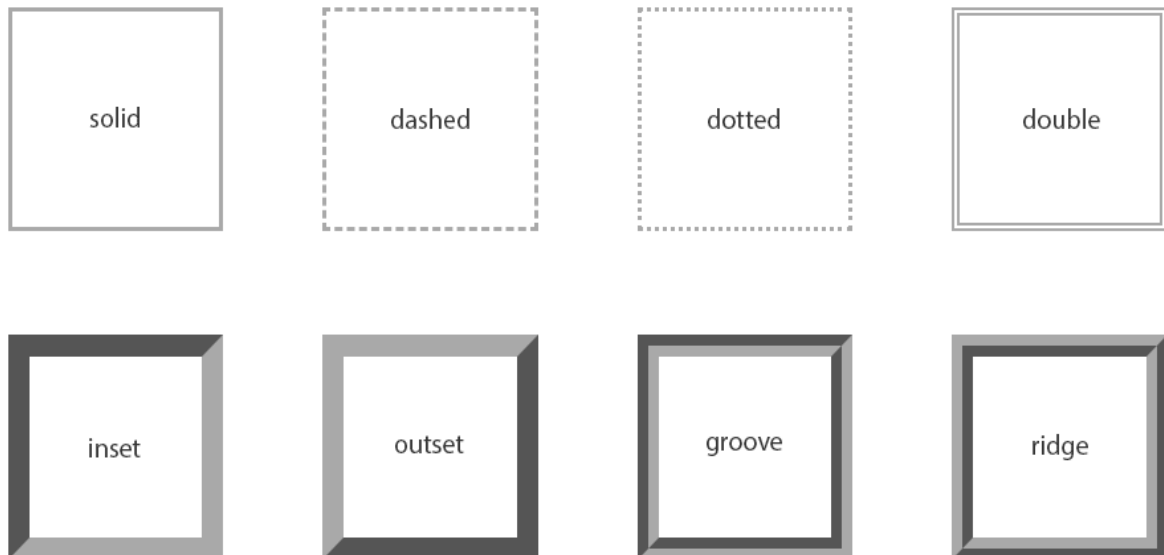
Pour empêcher le rembourrage et la bordure de modifier la largeur et la hauteur de la boîte de l'élément, vous pouvez utiliser la propriété CSS **box-sizing**. Dans l'exemple suivant, la largeur et la hauteur de la zone **<div>** resteront inchangées, cependant, sa zone de contenu diminuera avec l'augmentation du rembourrage ou de la bordure.

```
div {
  width: 100%;
  padding: 25px;
  box-sizing: border-box;
}
```

• Propriétés de bordure CSS

Les propriétés de bordure CSS vous permettent de définir la zone de bordure de la boîte d'un élément. Les bordures apparaissent directement entre la marge et le rembourrage d'un élément. La bordure peut être soit un style prédéfini comme une ligne continue, une ligne pointillée, une double ligne, etc. ou une image.

La propriété **border-style** définit le style de la bordure d'une boîte, par exemple: **solid**, **dotted**, etc. C'est une propriété abrégée pour définir le style de ligne pour les quatre côtés de la bordure des éléments. La propriété **border-style** peut avoir les valeurs suivantes: **none**, **hidden**, **solid**, **dashed**, **dotted**, **double**, **inset**, **outset**, **groove**, et **ridge**. Jetons maintenant un œil à l'illustration suivante, elle vous donne une idée des différences entre les types de style de bordure.



```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>CSS border-style Property</title>
<style>
  p {
    border-width: 15px;
    background: yellow;
    padding: 20px;
    margin: 20px;
  }
  p.none {
    border-style: none;
  }
  p.hidden {
    border-style: hidden;
  }
  p.dotted {
    border-style: dotted;
  }
  p.dashed {
    border-style: dashed;
  }
  p.solid {
    border-style: solid;
  }
  p.double {
    border-style: double;
  }
  p.groove {
    border-style: groove;
  }
  p.ridge {
    border-style: ridge;
  }

```

```

    }
    p.inset {
        border-style: inset;
    }
    p.outset {
        border-style: outset;
    }
</style>
</head>
<body>
    <p class="none">No border.</p>
    <p class="hidden">Hidden border.</p>
    <p class="dotted">A dotted border.</p>
    <p class="dashed">A dashed border.</p>
    <p class="solid">A solid border.</p>
    <p class="double">A double border.</p>
    <p class="inset">An inset border.</p>
    <p class="outset">An outset border.</p>
    <p class="groove">A groove border.</p>
    <p class="ridge">A ridge border.</p>
</body>
</html>

```

• Définition de la largeur de la bordure

La propriété **border-width** spécifie la largeur de la zone de bordure. Il s'agit d'une propriété abrégée permettant de définir simultanément l'épaisseur des quatre côtés de la bordure d'un élément. Essayons l'exemple suivant pour comprendre comment cela fonctionne:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>CSS border-width Property</title>
<style>
    p {
        border-style: solid;
        padding: 20px;
        margin: 20px;
    }
    p.one {
        border-width: 5px;
    }
    p.two {
        border-width: 5px 10px;
    }
    p.three {
        border-width: 5px 10px 15px;
    }
    p.four {
        border-width: medium 10px thick 15px;
    }

```

```

</style>
</head>
<body>
  <p class="one"><strong>one-value syntax:</strong> the single value sets the width
  of all four border sides.</p>
  <p class="two"><strong>two-value syntax:</strong> the first value sets the width
  of the top and bottom border, while the second value sets the width of the right and
  left sides border.</p>
  <p class="three"><strong>three-value syntax:</strong> the first value sets the
  width of the top border, the second value sets the width of the right and left border,
  and the third value sets the width of the the bottom border.</p>
  <p class="four"><strong>four-value syntax:</strong> each value sets the width of
  the border individually in the order top, right, bottom, and left.</p>
</body>
</html>

```

La largeur de la bordure peut également être spécifiée à l'aide de l'un des trois mots-clés suivants: **thin**, **medium** et **thick**.

• Spécification de la couleur de la bordure

La propriété **border-color** spécifie la couleur de la zone de bordure. Il s'agit également d'une propriété abrégée permettant de définir la couleur des quatre côtés de la bordure d'un élément.

Les règles de style suivantes ajoutent une bordure unie de couleur rouge autour des paragraphes.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>CSS border-color Property</title>
<style>
  p {
    border-width: 15px;
    padding: 20px;
    margin: 20px;
  }
  p.one {
    border-style: solid;
    border-color: #ff0000;
  }
  p.two {
    border-style: solid;
    border-color: #ff0000 #00ff00;
  }
  p.three {
    border-style: solid;
    border-color: #ff0000 #00ff00 #0000ff;
  }
  p.four {
    border-style: solid;
    border-color: #ff0000 #00ff00 #0000ff #ff00ff;
  }

```

```

    }
  </style>
</head>
<body>
  <p class="one"><strong>one-value syntax:</strong> the single value sets the color
  of all four border sides.</p>
  <p class="two"><strong>two-value syntax:</strong> the first value sets the color
  of the top and bottom border, while the second value sets the color of the right and
  left sides border.</p>
  <p class="three"><strong>three-value syntax:</strong> the first value sets the
  color of the top border, the second value sets the color of the right and left border, and
  the third value sets the color of the the bottom border.</p>
  <p class="four"><strong>four-value syntax:</strong> each value sets the color of
  the border individually in the order top, right, bottom, and left.</p>
</body>
</html>

```

La propriété CSS **border-width** ou **border-color** ne fonctionne pas si elle est utilisée seule. Utilisez la propriété **border-style** pour définir d'abord le style de la bordure.

• La propriété Border Shorthand

La propriété CSS **border** est une propriété abrégée permettant de définir une ou plusieurs propriétés de bordure individuelles **border-width**, **border-style** et **border-color** dans une seule règle. L'exemple suivant nous permet de comprendre comment cela fonctionne:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>CSS border Shorthand Property</title>
<style>
  p {
    border: 5px solid #00ff00;
        background: yellow;
    padding: 20px;
    margin: 20px;
  }
</style>
</head>
<body>
  <p>This is a paragraph with border.</p>
</body>
</html>

```

Si la valeur d'une propriété de bordure individuelle est omise ou non spécifiée lors de la définition de la propriété abrégée de bordure, la valeur par défaut de cette propriété sera utilisée à la place, le cas échéant. Par exemple, si la valeur de la propriété **border-color** est manquante ou non spécifiée lors de la définition de la bordure, la propriété **color** de l'élément sera utilisée comme valeur pour la couleur de la bordure. Dans l'exemple ci-dessous, la bordure sera une ligne rouge pleine de 5 pixels de largeur:


```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Omitting border-color Property</title>
<style>
  p {
    color: red;
        background: yellow;
    border: 5px solid;
    padding: 20px;
    margin: 20px;
  }
</style>
</head>
<body>
  <p>The border color of this paragraph is same as the text color.</p>
</body>
</html>
```

Mais, dans le cas du style de bordure, l'omission de la valeur ne fera apparaître aucune bordure, car la valeur par défaut de cette propriété est aucune. Dans l'exemple suivant, il n'y aura pas de bordure:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Ignoring border-style Property</title>
<style>
  p {
    border: 5px #00ff00;
    background: yellow;
    padding: 20px;
    margin: 20px;
  }
</style>
</head>
<body>
  <p>This paragraph has no border.</p>
</body>
</html>
```

- **Propriétés de la marge CSS**

Les propriétés de marge CSS vous permettent de définir l'espacement autour de la bordure de la boîte d'un élément. La marge d'un élément n'est pas affectée par sa couleur de fond, elle est toujours transparente. Cependant, si l'élément parent a la couleur d'arrière-plan, il sera visible à travers sa zone de marge.

- **Définition des marges pour les côtés individuels**

Vous pouvez spécifier les marges pour les côtés individuels d'un élément tels que les côtés supérieur, droit, inférieur et gauche à l'aide des propriétés CSS **margin-top**, **margin-right**, **margin-bottom** et **margin-left**, respectivement. Essayons l'exemple suivant pour comprendre comment cela fonctionne:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>CSS Margin for Individual Sides</title>
<style>
  h1 {
    margin-top: 50px;
    margin-bottom: 100px;
    background: yellow;
  }
  p {
    margin-left: 75px;
    margin-right: 75px;
    background: yellow;
  }
</style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a simple paragraph of text.</p>
  <p><strong>Note:</strong> Play with the margin property value to see how it
works.</p>
</body>
</html>
```

Les propriétés de marge peuvent être spécifiées à l'aide des valeurs suivantes:

length - spécifie une marge en px, em, rem, pt, cm, etc.

% - spécifie une marge en pourcentage (%) de la largeur de l'élément contenant.

auto - le navigateur calcule une marge appropriée à utiliser.

inherit - spécifie que la marge doit être héritée de l'élément parent.

Vous pouvez également spécifier des marges négatives sur un élément, par exemple, **margin: -10px ;**, **margin: -5% ;**, etc.

- **La propriété de raccourci de marge**

La propriété **margin** est une propriété abrégée pour éviter de définir la marge de chaque côté séparément, c'est-à-dire **margin-top**, **margin-right**, **margin-bottom** et **margin-left**.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Set Margin for All Sides At Once</title>
<style>
  h1 {
```

```

    margin: 50px; /* apply to all four sides */
  }
  p {
    margin: 25px 75px; /* vertical | horizontal */
  }
  div {
    margin: 25px 50px 75px; /* top | horizontal | bottom */
  }
  hr {
    margin: 25px 50px 75px 100px; /* top | right | bottom | left */
  }
  h1, p, div {
    background: yellow;
  }
</style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a simple paragraph of text.</p>
  <div>This is a DIV element.</div>
  <hr>
  <p><strong>Note:</strong> Play with the margin property value to see how it
works.</p>
</body>
</html>

```

Cette notation abrégée peut prendre une, deux, trois ou quatre valeurs séparées par des espaces.

- Si une valeur est spécifiée, elle est appliquée aux quatre côtés.
- Si deux valeurs sont spécifiées, la première valeur est appliquée aux côtés supérieur et inférieur, et la deuxième valeur est appliquée aux côtés droit et gauche de la boîte de l'élément.
- Si trois valeurs sont spécifiées, la première valeur est appliquée au haut, la deuxième valeur est appliquée aux côtés droit et gauche et la dernière valeur est appliquée au bas.
- Si quatre valeurs sont spécifiées, elles sont appliquées respectivement en haut, à droite, en bas et à gauche de la boîte de l'élément dans l'ordre spécifié.

Il est recommandé d'utiliser les propriétés abrégées, cela vous aidera à gagner du temps en évitant la saisie supplémentaire et à rendre votre code CSS plus facile à suivre et à maintenir.

• Centrage horizontal avec marges automatiques

La valeur **auto** de la propriété **margin** indique au navigateur Web de calculer automatiquement la marge. Ceci est couramment utilisé pour centrer un élément horizontalement dans un conteneur plus grand.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Centering an Element with CSS Auto Margin</title>

```

```
<style>
.container {
  width: 300px;
  height: 200px;
  background: #ddd;
  margin: 0 auto;
}
</style>
</head>
<body>
  <div class="container">
    <h1>This is a heading</h1>
    <p>This is a paragraph of text.</p>
  </div>
</body>
</html>
```

Les règles de style ci-dessus permettent à l'élément **<div>** d'occuper 300 pixels de tout l'espace horizontal disponible, et l'espace restant sera également divisé entre les marges gauche et droite.

Référence:

1. <https://www.tutorialrepublic.com/html-tutorial/>
2. <https://developer.mozilla.org/fr/docs/Web/Guide/HTML/HTML5>
3. **Jean-Pierre Vincent et Jonathan Verrecchia**, « HTML5 : De la page web à l'application web », **publié par Dunod.**