



HTML & CSS

Cours HTML & CSS

CSS APPROFONDIE

Cascading Style Sheets, est un langage de conception simple destiné à simplifier le processus de création de pages web.

CSS gère l'apparence d'une page Web. En utilisant CSS, vous pouvez contrôler la couleur du texte, le style des polices, l'espacement entre les paragraphes, la taille et la disposition des colonnes, les images de fond ou les couleurs utilisées, les schémas de disposition, les variations d'affichage et les tailles des éléments ainsi qu'une variété d'autres effets.

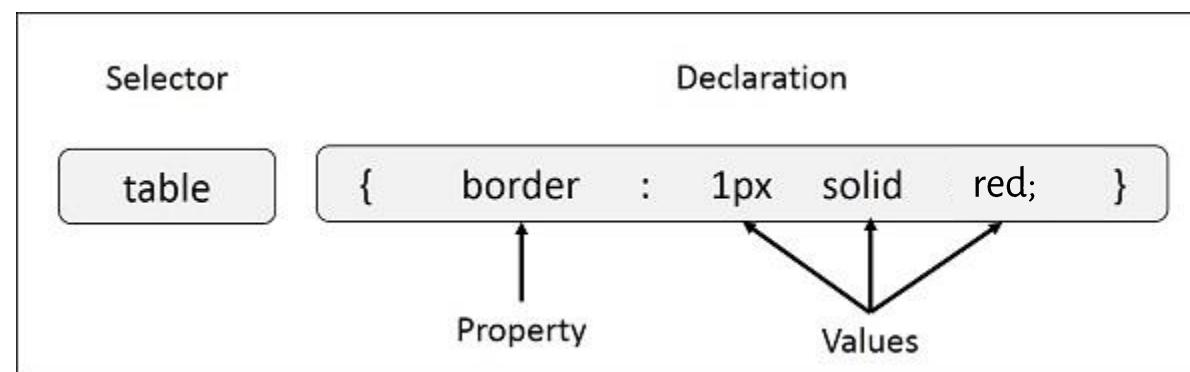
CSS est facile à apprendre et à comprendre, et il offre un contrôle puissant sur la présentation d'un document HTML.

CSS - Syntaxe

CSS comprend des règles de style qui sont interprétées par le navigateur, puis appliquées aux éléments correspondants dans votre document. Une règle de style est faite de trois parties :

- **Sélecteur** : Un sélecteur est une balise HTML à laquelle un style sera appliqué. Cela pourrait être n'importe quelle balise comme `<h1>` ou `<table>` etc.
- **Propriété** : Une propriété est un type d'attribut de balise HTML. En d'autres termes, tous les attributs HTML sont convertis en propriétés CSS. Ils pourraient être la couleur, la bordure etc.
- **Valeur** : Les valeurs sont affectées aux propriétés. Par exemple, la propriété `color` peut avoir une valeur `red` etc.

Vous pouvez mettre la syntaxe de la règle de style CSS comme suit :



CSS - Syntaxe

Exemple: Vous pouvez définir une largeur de table comme suit :

```
table {  
    width: 500px;  
}
```

Ici la **table** est un sélecteur et **width** est une propriété et la valeur donnée **500px** est la valeur de cette propriété.

La première chose que vous devriez apprendre en CSS est de savoir comment cibler les éléments. Vous pouvez définir des sélecteurs de diverses manières simples en fonction de votre confort.

Les sélecteurs par type

C'est le même sélecteur que nous avons vu ci-dessus. Encore une fois, un exemple de plus pour donner une couleur à tous les titres de niveau 1:

```
h1 {  
    color: green;  
}
```

CSS - Syntaxe

Les sélecteurs universels

Plutôt que de sélectionner des éléments d'un type spécifique, le sélecteur universel correspond tout simplement au nom de tout type d'élément :

```
* {  
    color: black;  
}
```

Cette règle rend le contenu de chaque élément de notre document en noir.

Les sélecteurs descendant

Supposons que vous souhaitez appliquer une règle de style à un élément particulier uniquement lorsqu'il se trouve à l'intérieur d'un élément particulier. Comme indiqué dans l'exemple suivant, la règle de style ne s'applique à l'élément *<i>* que lorsqu'il se trouve dans la balise *<h1>*.

```
h1 i {  
    color: black;  
}
```

CSS - Syntaxe

Les sélecteurs de classe

Vous pouvez définir des règles de style en fonction de l'attribut de classe des éléments. Tous les éléments ayant cette classe seront formatés selon la règle définie.

```
.class1 {  
    color: black;  
}
```

Cette règle rend le contenu en noir pour chaque élément avec l'attribut class défini sur **class1** (`class="class1"`) dans notre document. Vous pouvez le rendre un peu plus particulier. Par exemple:

```
h1.class1 {  
    color: black;  
}
```

Cette règle rend le contenu en noir uniquement pour les éléments `<h1>` avec l'attribut class défini sur **class1**. Vous pouvez appliquer plusieurs sélecteurs de classe à un élément donné. Considérez l'exemple suivant:

```
<address class="tetouan">  
    Cette adresse sera stylé par les classes tetouan.  
</address>
```

CSS - Syntaxe

Les sélecteurs d'ID

Vous pouvez définir des règles de style en fonction de l'attribut **id** des éléments. Tous les éléments ayant cet identifiant seront formatés selon la règle définie.

```
#id1 {  
    color: black;  
}
```

Cette règle rend le contenu en noir pour chaque élément avec l'attribut id défini sur **id1** dans notre document HTML.

Vous pouvez le rendre un peu plus particulier. Par exemple :

```
h1#id1 {  
    color: black;  
}
```

Cette règle rend le contenu en noir uniquement pour les éléments <h1> avec l'attribut id défini sur **id1**.

La vraie puissance des sélecteurs d'identifiants est quand ils sont utilisés comme base pour les sélecteurs descendants, par exemple :

```
#id1 h2 {  
    color: black;  
}
```

Dans cet exemple, tous les en-têtes de niveau 2 seront affichés en couleur noire lorsque ces en-têtes se trouveront dans des balises dont l'attribut id est défini sur **id1**.

CSS - Syntaxe

Les sélecteurs d'enfants

Vous avez vu les sélecteurs descendants. Il y a un autre type de sélecteur, qui ressemble beaucoup aux descendants mais qui a des fonctionnalités différentes. Considérez l'exemple suivant:

```
article > p {  
    color: black;  
}
```

Cette règle rendra tous les paragraphes en noir s'ils sont enfants directs de l'élément `<article>`. Les autres paragraphes placés à l'intérieur d'autres éléments comme `<div>` ou `<td>` n'auraient aucun effet sur cette règle.

Les sélecteurs d'attribut

Vous pouvez également appliquer des styles aux éléments HTML avec des attributs particuliers. La règle de style ci-dessous correspondra à tous les éléments d'entrée ayant un attribut de type avec une valeur de texte :

```
input[type="text"] {  
    color: black;  
}
```

L'avantage de cette méthode est que l'élément `<input type = "number" />` n'est pas affecté et que la couleur s'applique uniquement aux champs de texte souhaités.

CSS - Syntaxe

Règles de style multiples

Vous devrez peut-être définir plusieurs règles de style pour un seul élément. Vous pouvez définir ces règles pour combiner plusieurs propriétés et valeurs correspondantes dans un même bloc, comme indiqué dans l'exemple suivant:

```
h1 {  
    color: brown;  
    letter-spacing: 10px;  
    text-transform: uppercase;  
    text-decoration-line: underline;  
    text-decoration-color: blue;  
    text-decoration-style: double;  
}
```

Ici, toutes les paires de propriétés et de valeurs sont séparées par un point-virgule (:).

Grouper les sélecteurs

Vous pouvez appliquer un style à plusieurs sélecteurs si vous le souhaitez. Il suffit de séparer les sélecteurs avec une virgule, comme indiqué dans l'exemple suivant-

Cette règle de style de définition sera également applicable aux éléments h1, h2 et h3. L'ordre de la liste n'est pas pertinent. Tous les éléments du sélecteur recevront les déclarations correspondantes.

```
h1, h2, h3 {  
    font-weight: normal;  
    letter-spacing: .4em;  
    text-transform: lowercase;  
}  
  
table, #footer, .contenu {  
    width: 200px;  
    height: 100px;  
}
```

CSS - Inclusion

Il existe trois façons d'associer des styles à votre document HTML. Les méthodes les plus couramment utilisées sont CSS en ligne et CSS externe.

CSS intégré - L'élément <style>

Vous pouvez mettre vos règles CSS dans un document HTML en utilisant l'élément `<style>`. Cette balise est placée à l'intérieur des balises `<head> ... </ head>`. Les règles définies à l'aide de cette syntaxe seront appliquées à tous les éléments disponibles dans le document. Voici la syntaxe générique -

Voici l'exemple de l'intégration CSS basé sur la syntaxe ci-dessus -

Ceci est une rubrique

Ceci est un paragraphe.

```
<!DOCTYPE html>
<html>
  <head>
    <style type = "text/css">
      body {
        background-color: linen;
      }
      h1 {
        color: maroon;
        margin-left: 40px;
      }
    </style>
  </head>
  <body>
    <h1>Ceci est une rubrique</h1>
    <p>Ceci est un paragraphe.</p>
  </body>
</html>
```

CSS - Inclusion

Inline CSS - L'attribut de style

Vous pouvez utiliser l' attribut style d'un élément HTML pour définir des règles de style. Ces règles seront appliquées à cet élément seulement. Voici la syntaxe générique :

```
<html>
  <head></head>
  <body>
    <h1 style = "color:#36C;"> Ceci est ligne CSS </h1>
  </body>
</html>
```

```
<element style = "...style rules....">
```

Ceci est 'CSS en-ligne'

CSS externe - L'élément <link>

L'élément `<link>` peut être utilisé pour inclure un fichier de feuille de style externe dans votre document HTML.

Une feuille de style externe est un fichier texte distinct avec l' extension `.css` . Vous définissez toutes les règles de style dans ce fichier texte, puis vous pouvez inclure ce fichier dans n'importe quel document HTML en utilisant l'élément `<link>`.

```
<head>
  <link rel="stylesheet" href="style.css" />
</head>
```

CSS importé - Règle @import

@import est utilisé pour importer une feuille de style externe d'une manière similaire à l'élément <link>. Voici la syntaxe générique de la règle @import. | **@import "mystyle.css";**

Règles CSS surchargées – Priorité des règles

Nous avons discuté de quatre façons d'inclure des règles de feuille de style dans un document HTML. Voici la règle pour remplacer toute règle de feuille de style.

- Toute feuille de style en ligne (**style=""**) prend la plus haute priorité. Ainsi, il remplacera toute règle définie dans les balises **<style> ... </style>** ou les règles définies dans tout fichier de feuille de style externe.
- Toute règle définie dans les balises **<style> ... </style>** remplacera les règles définies dans tout fichier de feuille de style externe.
- Toute règle définie dans un fichier de feuille de style externe (.css) prend la priorité la plus faible et les règles définies dans ce fichier ne seront appliquées que si deux règles au-dessus ne s'appliquent pas.



CSS - Unités de mesure

CSS prend en charge un certain nombre de mesures, y compris les unités absolues telles que les pouces, les centimètres, les points, etc., ainsi que les mesures relatives telles que les pourcentages et les unités em. Vous avez besoin de ces valeurs lorsque vous spécifiez diverses mesures dans vos règles de style, par exemple **border = 1px solid red**.

Unité	Description	Exemple
px	Définit une mesure des pixels de l'écran.	p {padding: 25px;}
mm	Définit une mesure en millimètres.	p {word-spacing: 15mm;}
in	Définit une mesure en pouces.	p {word-spacing: .15in;}
%	Définit une mesure en pourcentage par rapport à une autre valeur, généralement un élément englobant.	p {font-size: 16pt; line-height: 125%;}
em	Une mesure relative de la hauteur d'une police dans les espaces em. Parce qu'une unité em est équivalente à la taille d'une police donnée.	p {font-size: 20px; line-height: 1.2em;}
vw	1% de la largeur de la fenêtre	h1 { font-size: 5.9vw; }
vh	1% de la hauteur de la fenêtre.	h2 { font-size: 3.0vh; }

Les couleurs sont très importantes pour donner un bon aspect à votre site Web. Vous pouvez spécifier des couleurs au niveau de la page à l'aide de la balise `<body>` ou vous pouvez définir des couleurs pour des balises individuelles à l'aide de la règle `color`.

Méthodes de codage de couleur en CSS et HTML :

Il existe trois méthodes différentes pour définir les couleurs dans votre page Web :

- Noms de couleur - Vous pouvez spécifier des noms de couleur directement comme le vert, le bleu ou le rouge.
- Codes hexadécimaux - Un code à six chiffres représentant la quantité de rouge, vert et bleu qui compose la couleur.
- Valeur décimale ou pourcentage en couleur - Cette valeur est spécifiée à l'aide de la propriété `rgb()`.

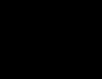
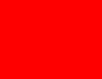
CSS - Couleurs

Couleurs HTML - Noms de couleurs

Vous pouvez spécifier un nom de couleur direct pour définir le texte ou la couleur d'arrière-plan. Le W3C a répertorié 16 noms de couleurs de base qui seront validés avec un validateur HTML, mais il existe plus de 200 noms de couleurs différents pris en charge par les principaux navigateurs.

W3C Standard 16 Couleurs

Voici la liste des noms W3C Standard 16 Couleurs et il est recommandé de les utiliser.

	Black		Gray		Silver		White
	Yellow		Lime		Aqua		Fuchsia
	Red		Green		Blue		Purple
	Maroon		Olive		Navy		Teal

CSS - Couleurs

Voici les exemples pour définir l'arrière-plan d'une balise HTML par nom de couleur :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Couleurs HTML par nom</title>
  </head>
  <body style="background-color: green;">
    <p>Utilisez des noms de couleurs différents pour le corps et la table et voyez le résultat.</p>
  </body>
</html>
```

CSS - Couleurs

Couleurs HTML - Codes hexadécimaux

Un hexadécimal est une représentation à 6 chiffres d'une couleur. Les deux premiers chiffres (RR) représentent une valeur rouge, les deux suivants sont une valeur verte (GG) et les derniers sont la valeur bleue (BB).

Chaque code hexadécimal sera précédé d'un dièse ou d'un dièse #. Voici une liste de quelques couleurs utilisant la notation hexadécimale.

Couleur	Couleur HEX
Black	# 000000
Red	# FF0000
Green	# 00FF00
Blue	# 0000FF
Yellow	# FFFF00
Cyan	# 00FFFF
Magenta	# FF00FF
Grey	# C0C0C0
White	#FFFFFF

CSS - Couleurs

Voici les exemples pour définir l'arrière-plan d'une balise HTML par code de couleur en hexadécimal :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Couleurs HTML par nom</title>
  </head>
  <body style="background-color: #008000;">
    <p>Utilisez des noms de couleurs différents pour le corps et la table et voyez le résultat.</p>
  </body>
</html>
```

Couleurs HTML - Valeurs RGB

Cette valeur de couleur est spécifiée à l'aide de la propriété `rgb()`. Cette propriété prend trois valeurs, une pour le rouge, le vert et le bleu. La valeur peut être un nombre entier compris entre 0 et 255 ou un pourcentage.

Remarque – Pas tous les navigateurs supportent la propriété `rgb()` de couleur, il est donc recommandé de ne pas les utiliser.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Couleurs HTML par nom</title>
  </head>
  <body style="background-color: rgb(0,128,1);">
    <p>Utilisez des noms de couleurs différents pour le corps et la table et voyez le résultat.</p>
  </body>
</html>
```

CSS – Arrière-plan

Ce chapitre vous apprend à définir les arrière-plans de divers éléments HTML. Vous pouvez définir les propriétés d'arrière-plan suivantes d'un élément:

- La propriété **background-color** est utilisée pour définir la couleur d'arrière-plan d'un élément.
- La propriété **background-image** est utilisée pour définir l'image d'arrière-plan d'un élément.
- La propriété **background-size** est utilisée pour redimensionner l'image d'arrière-plan.
- La propriété **background-repeat** est utilisée pour contrôler la répétition d'une image en arrière-plan.
- La propriété **background-position** est utilisée pour contrôler la position d'une image en arrière-plan.
- La propriété **background** est utilisée comme raccourci pour spécifier un certain nombre de propriétés d'arrière-plan.

Définir la couleur d'arrière-plan

```
<html>
  <head></head>
  <body>
    <p style = "background-color:yellow;">
      Ce texte a une couleur de fond jaune.</p>
  </body>
</html>
```

Ce texte a une couleur de fond jaune.

CSS – Arrière-plan

Définir l'image de fond

Nous pouvons définir l'image de fond en appelant des images stockées locales comme indiqué ci-dessous

```
<html>
<head>
  <style>
    body {
      background-image: url("css/images/css.jpg");
      background-color: #cccccc;
    }
  </style>
</head>
<body>
  <h1>Hello World!</h1>
</body>
</html>
```

Répétez l'image d'arrière-plan

L'exemple suivant montre comment répéter l'image d'arrière-plan si une image est petite. Vous pouvez utiliser la valeur **no-repeat** pour la propriété **background-repeat** si vous ne voulez pas répéter une image, dans ce cas l'image ne s'affichera qu'une seule fois.

```
<html>
<head>
  <style>
    body {
      background-image: url("css/images/css.jpg");
      background-repeat: repeat;
    }
  </style>
</head>
<body>
  <p>Paragraph</p>
</body>
</html>
```

CSS – Arrière-plan

Définir la position de l'image de fond

L'exemple suivant montre comment définir la position de l'image d'arrière-plan à 100 pixels du côté gauche et à 200 pixels du haut.

```
<html>
<head>
  <style>
    body {
      background-image: url("css/images/css.jpg");
      background-position: 100px 200px;
    }
  </style>
</head>
<body>
  <p>Paragraphe</p>
</body>
</html>
```

Propriété raccourcie

Vous pouvez utiliser la propriété `background` pour définir toutes les propriétés d'arrière-plan à la fois.
Par exemple :

```
<p style="background: url(images/pattern1.gif) repeat;">
  Ce paragraphe a corrigé l'image de fond répétée.
</p>
```

CSS - Polices

Cette partie vous apprend à définir les polices d'un contenu, disponible dans un élément HTML. Vous pouvez définir les propriétés de police suivantes d'un élément :

- La propriété **font-family** est utilisée pour changer le type d'une police.
- La propriété **font-style** est utilisée pour créer une police italique.
- La propriété **font-variant** est utilisée pour créer un effet small-caps.
- La propriété **font-weight** est utilisée pour augmenter ou diminuer la façon dont une police en gras ou claire apparaît.
- La propriété **font-size** est utilisée pour augmenter ou diminuer la taille d'une police.
- La propriété **font** est utilisée comme raccourci pour spécifier un certain nombre d'autres propriétés de police.

Définir la famille de polices

```
<html>
  <head></head>
  <body>
    <p style="font-family:georgia,garamond,serif;">
      Ce texte est affiché dans georgia, garamond ou la police serif par défaut en fonction
      en fonction de la police que vous avez sur votre système.
    </p>
  </body>
</html>
```

Ce texte est affiché dans georgia, garamond ou la police serif par défaut en fonction de la police que vous avez sur votre système.

CSS - Polices

Définir le style de police

Voici l'exemple, qui montre comment définir le style de police d'un élément.

Ce texte sera rendu en italique

```
<html>
  <head></head>
  <body>
    <p style="font-style:italic;">
      Ce texte sera rendu en italique
    </p>
  </body>
</html>
```

Définir la variante de police

L'exemple suivant montre comment définir la variante de police d'un élément. Les valeurs possibles sont normal et small-caps.

```
<html>
  <head></head>
  <body>
    <p style="font-variant:small-caps;">
      Ce texte sera rendu en petites majuscules
    </p>
  </body>
</html>
```

CE TEXTE SERA RENDU EN PETITES MAJUSCULES

CSS - Polices

Définir le poids de la police

L'exemple suivant montre comment définir le poids de la police d'un élément. La propriété font-weight fournit la fonctionnalité permettant de spécifier la taille d'une police. Les valeurs possibles peuvent être normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900.

```
<html>
  <head></head>
  <body>
    <p style="font-weight:bold;">Cette police est en gras.</p>
    <p style="font-weight:bolder;">Cette police est plus gras.</p>
    <p style="font-weight:500;">Cette police est de 500 poids.</p>
  </body>
</html>
```

Définir la taille de la police

L'exemple suivant montre comment définir la taille de police d'un élément. La propriété font-size est utilisée pour contrôler la taille des polices.

```
<html>
  <head></head>
  <body>
    <p style="font-size:20px;">Cette taille de police est de 20 pixels</p>
    <p style="font-size:small;">Cette taille de police est petite</p>
    <p style="font-size:large;">Cette taille de police est grande</p>
  </body>
</html>
```

CSS - Polices

Polices Web (Web Fonts)

Les polices Web sont une fonctionnalité CSS qui vous permet de spécifier les fichiers de polices à télécharger avec votre site Web au fur et à mesure de son accès, ce qui signifie que tout navigateur prenant en charge les polices Web peut avoir exactement les polices que vous spécifiez. La syntaxe requise ressemble à ceci:

Tout d'abord, vous avez un bloc `@font-face` au début du CSS, qui spécifie le (s) fichier (s) de police à télécharger:

```
@font-face {  
    font-family: "myFont";  
    src: url("myFont.ttf");  
}
```

En dessous, vous pouvez utiliser le nom de famille de police spécifié dans `@ font-face` pour appliquer votre police personnalisée à tout ce que vous voulez, comme d'habitude:

```
html {  
    font-family: "myFont", "Bitstream Vera Serif", serif;  
}
```

CSS - Liens

Vous pouvez changer le style de votre lien dans différents états en utilisant des pseudo-classes, les états possibles sont: **hover**, **active**, **visited** et **link**.

```
a:link {  
    color: #000000;  
}
```

signifie des liens hypertextes non visités.

```
a:visited {  
    color: #006600;  
}
```

signifie les hyperliens visités.

```
a:hover {  
    color: #FFCC00;  
}
```

indique un élément sur lequel le pointeur de la souris de l'utilisateur est actuellement placé au-dessus.

```
a:active {  
    color: #FF00CC;  
}
```

signifie un élément sur lequel l'utilisateur clique actuellement.

CSS - Bordures

Les propriétés **border** CSS vous permettent de spécifier le style, la largeur et la couleur de la bordure d'un élément.

J'ai des frontières de tous les côtés.

J'ai une bordure inférieure rouge.

J'ai arrondi les frontières.

J'ai une bordure gauche bleue.

CSS - Bordures

Style de bordure

La propriété **border-style** spécifie le type de bordure à afficher.

Les valeurs suivantes sont autorisées:

- **dotted** - Définit une bordure en pointillés.
- **dashed** - Définit une bordure en pointillés.
- **solid** - Définit une bordure solide.
- **double** - Définit une double bordure.
- **none** - Définit pas de bordure.

La propriété **border-style** peut avoir de une à quatre valeurs (pour la bordure supérieure, la bordure droite, la bordure inférieure et la bordure gauche).

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p {
      border-style: dotted;
    }
  </style>
</head>
<body>
  <p>Une bordure en pointillés.</p>
</body>
</html>
```

Une bordure en pointillés.

CSS - Bordures

Largeur de la bordure

La propriété **border-width** spécifie la largeur des quatre bordures.

La largeur peut être définie comme une taille spécifique (en px, pt, cm, em, etc).

La propriété **border-width** peut avoir de une à quatre valeurs (pour la bordure supérieure, la bordure droite, la bordure inférieure et la bordure gauche).

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p {
      border-style: dashed;
      border-width: 5px;
      border-color: cornflowerblue;
    }
  </style>
</head>
<body>
  <p>Une bordure en pointillés.</p>
</body>
</html>
```



Une bordure en pointillés.

CSS - Bordures

La propriété border-radius

La propriété **border-radius** définit le rayon des coins de l'élément.

Cette propriété vous permet d'ajouter des bordures arrondies aux éléments!

```
<html>
  <head>
    <style>
      p {
        border-style: solid;
        border-color: darkmagenta;
        border-width: 5px;
        border-top-left-radius: 10px;
        border-bottom-right-radius: 10px;
      }
    </style>
  </head>

  <body>
    <p>Ceci est un paragraphe!</p>
  </body>

</html>
```

Ceci est un paragraphe!

CSS - Bordures

Bordures - Côtés individuels

À partir des exemples ci-dessus, vous avez vu qu'il est possible de spécifier une bordure différente pour chaque côté.

En CSS, il y a aussi des propriétés pour spécifier chacune des bordures (top, right, bottom et left):

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p {
      border-top-style: dashed;
      border-top-color: cornflowerblue;
      border-bottom-style: double;
      border-bottom-color: darkviolet;
    }
  </style>
</head>
<body>
  <p>Une bordure de paragraphe.</p>
</body>
</html>
```

Une bordure de paragraphe.



CSS - Listes

Les propriétés de la liste CSS vous permettent de:

- Définir différents marqueurs d'élément de liste pour les listes ordonnées
- Définir différents marqueurs d'élément de liste pour les listes non ordonnées
- Changer la position des marqueurs.

La propriété **list-style-type** spécifie le type de marqueur d'élément de liste. Si vous souhaitez supprimer les marqueurs, utilisez la valeur none.

```
<p>Exemple de listes non ordonnées:</p>
<ul style="list-style-type: circle;">
    <li>choix 1</li>
    <li>choix 2</li>
    <li>choix 3</li>
</ul>
```

Exemple de listes non ordonnées:

- choix 1
- choix 2
- choix 3

```
<p>Exemple de listes ordonnées:</p>
<ol style="list-style-type: lower-alpha;">
    <li>choix 1</li>
    <li>choix 2</li>
    <li>choix 3</li>
</ol>
```

Exemple de listes ordonnées:

- a. choix 1
- b. choix 2
- c. choix 3

CSS - Listes

Les propriétés de la liste CSS vous permettent de:

- Définir différents marqueurs d'élément de liste pour les listes ordonnées ``
- Définir différents marqueurs d'élément de liste pour les listes non ordonnées ``
- Changer la position des marqueurs.

La propriété **list-style-position** spécifie la position des marqueurs d'élément de liste.

`<p>Exemple de listes non ordonnées:</p>`

```
<ul style="list-style-position: outside;">
    <li style="border: solid;">choix 1</li>
    <li>choix 2</li>
    <li>choix 3</li>
</ul>
```

`<p>Exemple de listes ordonnées:</p>`

```
<ol style="list-style-position: inside;">
    <li style="border: solid;">choix 1</li>
    <li>choix 2</li>
    <li>choix 3</li>
</ol>
```

Exemple de listes non ordonnées:

- choix 1
- choix 2
- choix 3

Exemple de listes ordonnées:

1. choix 1
2. choix 2
3. choix 3

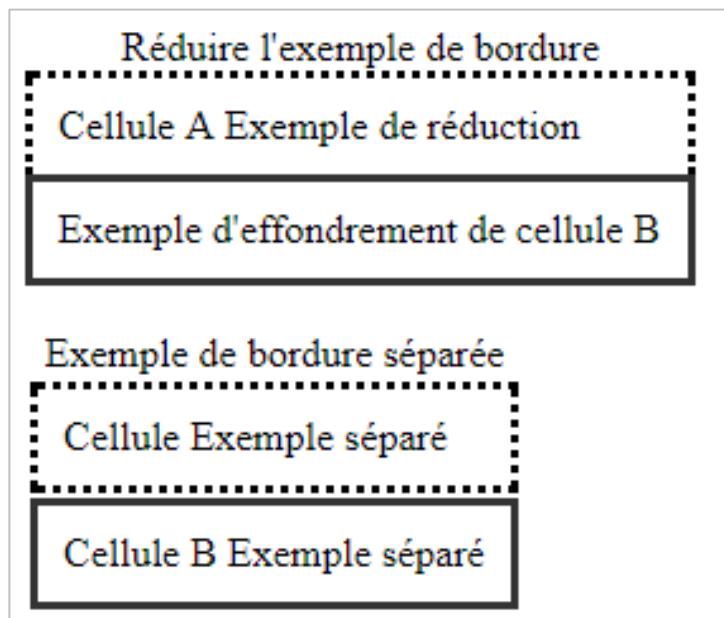
Cette partie va vous apprendre à définir différentes propriétés d'un tableau HTML en utilisant CSS. Vous pouvez définir les propriétés suivantes d'une table :

- La propriété d'effondrement de la bordure **border-collapse** indique si le navigateur doit contrôler l'apparence des bordures adjacentes qui se touchent ou si chaque cellule doit maintenir son style.
- Les règles **border-width**, **border-style** et **border-radius** définissent le style de la bordure de la table.
- Le côté de la légende **caption-side** est présentées dans l'élément **<caption>**. Par défaut, ils sont rendus au-dessus de la table dans le document. Vous utilisez la propriété **caption-side** pour contrôler l'emplacement de la légende de la table.

Maintenant, nous allons voir comment utiliser ces propriétés avec des exemples.

La propriété border-collapse

Cette propriété peut avoir deux valeurs **collapse** et **separate**. L'exemple suivant utilise les deux valeurs:



```
<html>
  <head>
    <style type="text/css">
      table.table1 {border-collapse:collapse;}
      table.table2 {border-collapse:separate;}
      td.a {
        border-style:dotted;
        border-width:3px;
        border-color:#000000;
      }
      td.b {
        border-style:solid;
        border-width:3px;
        border-color:#333333;
      }
    </style>
  </head>
  <body>
    <table class="table1">
      <caption>Collapse-Border Exemple</caption>
      <tr> <td class="a"> Cellule A Collapse Exemple</td> </tr>
      <tr> <td class="b"> Cellule B Collapse Exemple</td> </tr>
    </table>
    <br />
    <table class="table2">
      <caption>Border Séparé Example</caption>
      <tr> <td class="a"> Cellule A Separate Exemple</td> </tr>
      <tr> <td class="b"> Cellule B Separate Exemple</td> </tr>
    </table>
  </body>
</html>
```

CSS - Tables

La propriété **caption-side**:

La propriété **caption-side** vous permet de spécifier où le contenu d'un élément **<caption>** doit être placé en relation avec la table. Le tableau suivant répertorie les valeurs possibles.

Cette propriété peut avoir l'une des valeurs **top**, **bottom**. L'exemple suivant utilise chaque valeur.

REMARQUE: Ces propriétés peuvent ne pas fonctionner avec votre navigateur IE7.

Cette légende apparaîtra en haut	
Cellule A	Cellule B
Cellule A	Cellule B
Cette légende apparaîtra au bas	

```
<html>
  <head>
    <style type="text/css">
      caption.top {caption-side:top}
      caption.bottom {caption-side:bottom}
    </style>
  </head>
  <body>
    <table style="width:400px; border:1px solid black;">
      <caption class="top">
        Cette légende apparaîtra en haut
      </caption>
      <tr><td> Cellule A</td></tr>
      <tr><td> Cellule B</td></tr>
    </table>
    <br />

    <table style="width:400px; border:1px solid black;">
      <caption class="bottom">
        Cette légende apparaîtra au bas
      </caption>
      <tr><td> Cellule A</td></tr>
      <tr><td> Cellule B</td></tr>
    </table>
    <br />
  </body>
</html>
```

CSS - Tables

L'alignement du texte. La propriété vertical-align + text-align

La propriété **vertical-align** définit l'alignement vertical d'un élément.

La propriété **text-align** définit l'alignement horizontal d'un élément.

```
<!DOCTYPE html>
<html>

<body>
  <table border="1" height="100px" width="100%">
    <tbody>
      <tr>
        <td>Aligné normalement</td>
        <td style="vertical-align: top;">Aligné vers le haut</td>
        <td style="vertical-align: top; text-align: center;">Aligné vers le haut+centré</td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

Aligné normalement	Aligné vers le haut	Aligné vers le haut+centré
--------------------	---------------------	----------------------------

CSS - Texte

Cette partie vous apprend à manipuler le texte à l'aide des propriétés CSS. Vous pouvez définir les propriétés de texte suivantes d'un élément (par exemple, un paragraphe)

- La propriété **color** est utilisée pour définir la couleur d'un texte.
- La propriété **letter-spacing** est utilisée pour ajouter ou soustraire de l'espace entre les lettres qui composent un mot.
- La propriété **word-spacing** est utilisée pour ajouter ou soustraire de l'espace entre les mots d'une phrase.
- La propriété **text-indent** est utilisée pour mettre en retrait le texte d'un paragraphe.
- La propriété **text-decoration** est utilisée pour souligner, surligner et barrer du texte.
- La propriété **text-transform** permet de convertir du texte en majuscules ou en minuscules.
- La propriété **line-height** spécifie la hauteur d'une ligne (l'espace entre les lignes d'un paragraphe).

Ce texte sera écrit en rouge.

Ce texte a de l'espace entre les lettres.

Ce texte a de l'espace entre les lettres.

Ce texte aura la première ligne en retrait de 1cm et cette ligne restera à sa position actuelle.

Cela sera souligné

CE SERA EN MAJUSCULE

Ceci est un paragraphe avec une plus petite hauteur de ligne.
La hauteur de la ligne est ici réglée sur 10px.

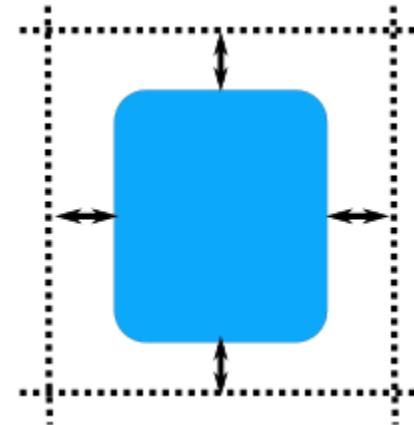
CSS - Margin

La propriété **margin** définit l'espace autour d'un élément HTML. Il est possible d'utiliser des valeurs négatives pour la propriété **margin**.

Nous avons les propriétés suivantes pour définir une marge d'élément :

- La valeur **margin** spécifie une propriété abrégée pour définir les propriétés de marge dans une déclaration.
- La marge inférieure **margin-bottom** indique la marge inférieure d'un élément.
- Le **margin-top** spécifie la marge supérieure.
- La **margin-left** spécifie la marge gauche d'un élément.
- La **margin-right** spécifie la marge droite d'un élément.

```
<html>
  <head></head>
  <body>
    <p style="margin: 150px; border:1px solid black;">
      les quatre marges seront 150px
    </p>
  </body>
</html>
```



La valeur auto

Vous pouvez définir la propriété margin sur **auto** pour centrer horizontalement l'élément de bloc dans son conteneur. L'élément prendra alors la largeur spécifiée, et l'espace restant sera partagé également entre les marges gauche et droite.



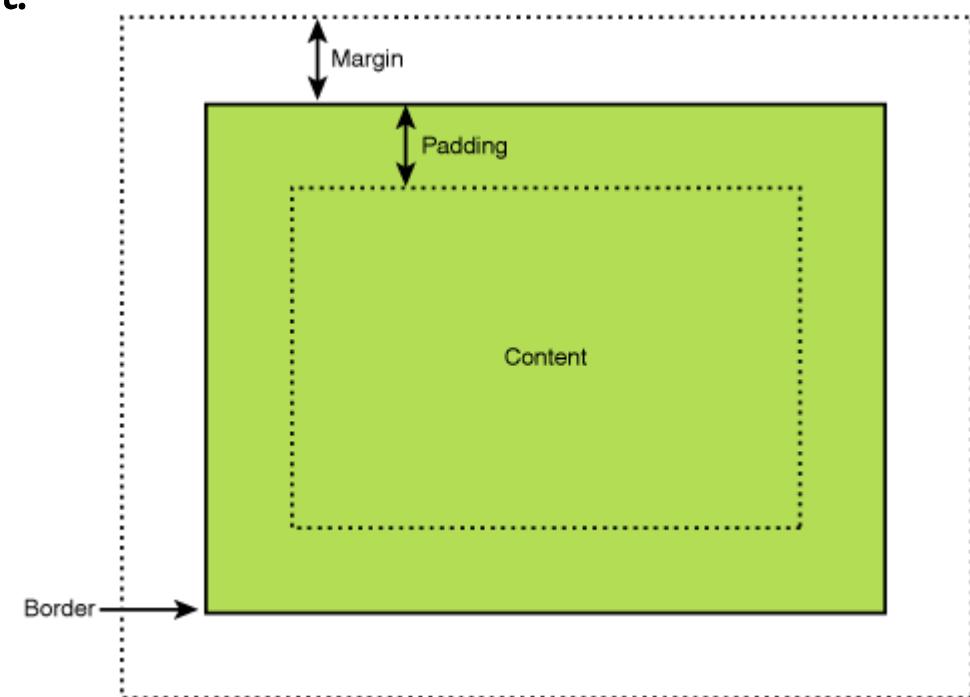
CSS - Padding

La propriété **padding** vous permet de spécifier l'espace qui doit apparaître entre le contenu d'un élément et sa bordure.

Les propriétés CSS suivantes peuvent être utilisées pour contrôler la valeur **padding**. Vous pouvez également définir des valeurs différentes pour le remplissage (padding) de chaque côté de la boîte en utilisant les propriétés suivantes :

- **padding-bottom** spécifie le remplissage de fond d'un élément.
- **padding-top** spécifie le remplissage supérieur d'un élément.
- **padding-left** spécifie le remplissage à gauche d'un élément.
- **padding-right** spécifie le bon remplissage d'un élément.
- **padding** sert de raccourci pour les propriétés précédentes.

```
<html>
  <head></head>
  <body>
    <p style="padding: 150px; border:1px solid black;">
      les quatre marges seront 150px
    </p>
  </body>
</html>
```



CSS - Positionnement

CSS vous aide à positionner votre élément HTML. Vous pouvez mettre n'importe quel élément HTML à l'endroit que vous voulez. Vous pouvez spécifier si vous souhaitez que l'élément soit positionné par rapport à sa position naturelle dans la page ou absolu en fonction de son élément parent.

Maintenant, nous allons voir toutes les propriétés liées au positionnement CSS avec des exemples :

Relative Positioning

Le positionnement relatif modifie la position de l'élément HTML par rapport à l'endroit où il apparaît normalement. Donc "left: 20" ajoute 20 pixels à la position GAUCHE de l'élément.

Vous pouvez utiliser deux valeurs **top** et **left** avec la propriété position pour déplacer un élément HTML n'importe où dans le document HTML.

- Déplacer vers la gauche - Utilisez une valeur négative pour **left**.
- Déplacer vers la droite - Utilisez une valeur positive pour **left**.
- Déplacer vers le haut - Utilisez une valeur négative pour **top**.
- Déplacer vers le bas - Utilisez une valeur positive pour **top**.

NOTE - Vous pouvez également utiliser les valeurs **bottom** ou **right** de la même manière qu'en **top** et **left**.

CSS - Positionnement

Relative Positioning (suite)

```
<html>
  <head></head>
  <body>
    <div style="position:relative;left:80px;top:2px;background-color:yellow;">
      Cette div a un positionnement relatif.
    </div>
  </body>
</html>
```

Absolute Positioning

Le positionnement absolu change le positionnement d'un élément par rapport au premier élément parent positionné!

Vous pouvez utiliser deux valeurs en haut et à gauche avec la propriété position pour déplacer un élément HTML n'importe où dans le document HTML par rapport à l'élément parent positionné.

```
<html>
  <body>
    <div style="position:absolute; left:80px; top:20px; background-color:yellow;">
      Cette div a un positionnement absolu.
    </div>
  </body>
</html>
```

Fixed Positioning

Le positionnement fixe vous permet de fixer la position d'un élément à un endroit particulier de la page, indépendamment du défilement. Les coordonnées spécifiées seront relatives à la fenêtre du navigateur. Vous pouvez utiliser deux valeurs **top** et **left** avec la propriété **position** pour déplacer un élément HTML n'importe où dans le document HTML.

- Déplacer vers la gauche - Utilisez une valeur négative pour **left**.
- Déplacer vers la droite - Utilisez une valeur positive pour **left**.
- Déplacer vers le haut - Utilisez une valeur négative pour **top**.
- Déplacer vers le bas - Utilisez une valeur positive pour **top**.

```
<html>
  <head></head>
  <body>
    <div style="position:fixed; left:80px; top:20px; background-color:yellow;">
      Cette div a un positionnement fixe.
    </div>
  </body>
</html>
```

Tous les éléments HTML peuvent être classés en deux catégories (a) Éléments de niveau de bloc (b) Éléments en ligne.

Block Elements (Éléments des blocs)

Les éléments de bloc apparaissent sur l'écran comme s'ils avaient un saut de ligne avant et après eux. Par exemple, les `<p>`, `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`, ``, ``, `<dl>`, `<pre>`, `<hr>`, `<blockquote>` et `<address>` sont tous des éléments de niveau bloc. Ils commencent tous sur leur propre ligne, et tout ce qui les suit apparaît sur sa propre ligne.

Inline Elements (Éléments en ligne)

D'autre part, les éléments en ligne peuvent apparaître dans les phrases et ne doivent pas apparaître sur une nouvelle ligne. Les ``, `<i>`, `<u>`, ``, ``, `<sup>`, `<sub>`, `<big>`, `<small>`, `<ins>`, ``, `<code>`, `<cite>`, `<dfn>`, `<kbd>` et `<var>` sont tous des éléments en ligne.

Grouper des éléments HTML

Il y a deux balises importantes que nous utilisons très fréquemment pour grouper d'autres balises HTML : la balise `<div>` et la balise ``.

La balise `<div>`

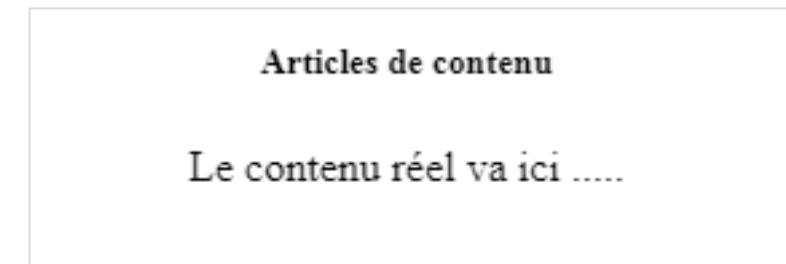
C'est la balise de niveau de bloc très importante qui joue un grand rôle dans le regroupement de divers autres balises HTML et l'application de CSS sur le groupe d'éléments. La balise `<div>` peut être utilisée pour créer une mise en page de page Web où nous définissons différentes parties (Gauche, Droite, Haut, etc.) de la page en utilisant la balise `<div>`. Cette balise ne fournit aucun changement visuel sur le bloc mais cela a plus de sens quand il est utilisé avec CSS.

HTML Blocs - Regrouper le contenu

Les éléments `<div>` et `` vous permettent de regrouper plusieurs éléments pour créer des sections ou des sous-sections d'une page.

Par exemple, vous pouvez placer toutes les notes de bas de page sur une page dans un élément `<div>` pour indiquer que tous les éléments de cet élément `<div>` se rapportent aux notes de bas de page. Vous pouvez ensuite attacher un style à cet élément `<div>` afin qu'il apparaisse en utilisant un ensemble spécial de règles de style.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple de tag div</title>
  </head>
  <body>
    <div id = "content" align = "center">
      <h5>Articles de contenu</h5>
      <p>Le contenu réel va ici .....</p>
    </div>
  </body>
</html>
```



HTML Blocs - Regrouper le contenu

D'autre part, l'élément `` peut être utilisé pour grouper uniquement les éléments en ligne. Donc, si vous avez une partie d'une phrase ou d'un paragraphe que vous voulez regrouper, vous pouvez utiliser l'élément `` comme suit.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Exemple de balise de répartition</title>
  </head>

  <body>
    <p>C'est l'exemple de <span style = "color:green">balise span</span>
       et la <span style = "color:red">balise div</span> avec CSS</p>
  </body>

</html>
```

C'est l'exemple de **balise span** et la **balise div** avec CSS

Ces balises sont généralement utilisées avec CSS pour vous permettre d'attacher un style à une section d'une page.

HTML Blocs

Voici un exemple simple de balise <div>. Nous allons apprendre la feuille de style en cascade (CSS) dans un chapitre séparé, mais nous l'avons utilisé ici pour montrer l'utilisation de la balise <div> :

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML div Tag</title>
  </head>
  <body>
    <!-- Premier groupe de tags -->
    <div style = "color:red">
      <h4>C'est le premier groupe</h4>
      <p>Voici une liste de légumes</p>

      <ul>
        <li>Betterave</li>
        <li>Gingembre</li>
        <li>Patate</li>
        <li>Un radis</li>
      </ul>
    </div>
    <!-- Deuxième groupe de tags -->
    <div style = "color:green">
      <h4>C'est le deuxième groupe</h4>
      <p>Voici une liste de fruits</p>

      <ul>
        <li>Pomme</li>
        <li>Banana</li>
        <li>Mangue</li>
        <li>Fraise</li>
      </ul>
    </div>
  </body>
</html>
```

C'est le premier groupe

Voici une liste de légumes

- Betterave
- Gingembre
- Patate
- Un radis

C'est le deuxième groupe

Voici une liste de fruits

- Pomme
- banane
- Mangue
- fraise

HTML Blocs

La balise

 est un élément en ligne qui peut être utilisé pour regrouper des éléments en ligne dans un document HTML. Cette balise ne fournit pas non plus de changement visuel sur le bloc mais a plus de sens quand elle est utilisée avec CSS.

La différence entre la balise et la balise <div> est que la balise est utilisée avec des éléments en ligne alors que la balise <div> est utilisée avec des éléments de niveau bloc.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML span Tag</title>
  </head>
  <body>
    <p>Ceci est <span style = "color:red">rouge</span> et ceci est
       <span style = "color:green">vert</span></p>
  </body>
</html>
```

C'est rouge et c'est vert

CSS - display

Il existe 2 types d'éléments différents, des éléments de bloc (e.x: <div>) et des éléments en ligne (e.x,).

Eléments au niveau du bloc - Block-level Elements

Un élément de niveau bloc commence toujours sur une nouvelle ligne et occupe toute la largeur disponible (s'étire vers la gauche et la droite aussi loin que possible).

L'élément <div> est un élément de niveau bloc.

Exemples d'éléments de niveau bloc: <div>, <h1>-<h6>, <p>, <form>, <header>, <footer>, <nav>

Eléments en ligne – Inline-level Elements

Un élément en ligne ne démarre pas sur une nouvelle ligne et ne prend que la largeur nécessaire.

C'est un élément inséré dans un paragraphe.

Exemples d'éléments en ligne: , <a>,

Cependant, vous pouvez changer l'affichage de l'élément en utilisant la règle **display**.

Les valeurs les plus utilisées pour la règle d'affichage sont:

- **Inline** : Affiche un élément en tant qu'élément en ligne (comme ``). Toutes les propriétés de hauteur et de largeur n'auront aucun effet.
- **Block** : Affiche un élément en tant qu'élément de bloc (comme `<p>` et `<div>`). Il commence sur une nouvelle ligne, et prend toute la largeur.
- **Inline-block** : Affiche un élément en tant que conteneur de blocs en ligne. L'élément lui-même est formaté en tant qu'élément en ligne, mais vous pouvez appliquer des valeurs de hauteur et de largeur.
- **None** : L'élément est complètement supprimé de la mise en page.

CSS - display

display: inline;

```
<html>
  <head>
    <style>
      li {
        display: inline;
      }
    </style>
  </head>
  <body>
    <nav>
      <ul>
        <li>Page d'accueil</li>
        <li>Qui sommes-nous ?</li>
        <li>Contactez nous</li>
      </ul>
    </nav>
  </body>
</html>
```

Sans display: inline;

- Page d'accueil
- Qui sommes-nous ?
- Contactez nous

Avec display: inline;

Page d'accueil Qui sommes-nous ? Contactez nous

CSS - display

display: block;

```
<html>
  <head>
    <style>
      i {
        display: block;
      }
    </style>
  </head>
  <body>
    <p>Bonjour, Je m'appelle <i>Elon Musk</i>, j'ai <i>47 ans</i>, je suis un entrepreneur.
    </p>
  </body>
</html>
```

Avec *display: block;*

Sans display: block;

Bonjour, Je m'appelle *Elon Musk*, j'ai *47 ans*, je suis un entrepreneur.

Bonjour, Je m'appelle
Elon Musk
, j'ai
47 ans
, je suis un entrepreneur.

CSS - display

display: inline-block;

```
<html>
  <head>
    <style>
      li {
        display: inline-block;
        width: 150px;
      }
    </style>
  </head>
  <body>
    <nav>
      <ul>
        <li>Page d'accueil</li>
        <li>Qui sommes-nous ?</li>
        <li>Contactez nous</li>
      </ul>
    </nav>
  </body>
</html>
```

Sans display: inline-block;

- Page d'accueil
- Qui sommes-nous ?
- Contactez nous

Avec display: inline-block;

Page d'accueil Qui sommes-nous ? Contactez nous

CSS - float

La propriété **float** provoque le déplacement d'un élément d'un côté de la zone de contenu de l'élément parent, ce qui permet à un autre contenu de le contourner. La propriété **float** est utilisée pour le positionnement et la mise en page sur les pages Web.

La propriété **float** peut avoir l'une des valeurs suivantes:

- **left** - L'élément est placé sur le côté gauche de la zone de contenu de l'élément parent.
- **right** - L'élément est placé sur le côté droit de la zone de contenu de l'élément parent.

Dans son utilisation la plus simple, la propriété **float** peut être utilisée pour enruler le texte autour des images.

CSS - float

```
<html>
  <head>
    <style>
      img {
        float: left;
      }
    </style>
  </head>
  <body>
    <p>
       Android est un système
      et d'autres logiciels open source et conçu principalement
      pour les smartphones et les tablettes.
    </p>
  </body>
</html>
```

Sans float: left;



ANDROID Android est un système d'exploitation mobile développé par Google, basé sur une version modifiée du noyau Linux et d'autres logiciels open source et conçu principalement pour les appareils mobiles à écran tactile tels que les smartphones et les tablettes.

Avec float: left;



Android est un système d'exploitation mobile développé par Google, basé sur une version modifiée du noyau Linux et d'autres logiciels open source et conçu principalement pour les appareils mobiles à écran tactile tels que les smartphones et les tablettes.

La propriété clear

Si vous ne voulez pas qu'un élément flotte à côté d'un élément flottant, utilisez la propriété **clear**. La propriété **clear** spécifie quels éléments peuvent flotter à côté de l'élément flottant et de quel côté.

La propriété **clear** peut avoir l'une des valeurs suivantes:

- **left** - Aucun élément flottant autorisé sur le côté gauche
- **right** - Aucun élément flottant autorisé sur le côté droit
- **both** - Aucun élément flottant autorisé sur le côté gauche ou droit

La manière la plus courante d'utiliser la propriété **clear** est après avoir utilisé une propriété **float** sur un élément.

Lors de l'effacement des flotteurs en utilisant **clear**, vous devez faire correspondre le cote **clear** au cote **float**. Si un élément est flottant vers la gauche (**float: left**), alors vous devriez effacer vers la gauche (**clear: left**). Votre élément flottant continuera à flotter, mais l'élément effacé apparaîtra en dessous de celui-ci sur la page Web.

CSS – Pseudo classes

Les pseudo-classes CSS sont utilisées pour ajouter des effets spéciaux à certains sélecteurs.

Les pseudo-classes les plus couramment utilisées sont les suivantes:

- **:visited** Utilisez cette classe pour ajouter un style spécial à un lien visité.
- **:hover** Utilisez cette classe pour ajouter un style spécial à un élément lorsque vous le survolez.
- **:first-child** Utilisez cette classe pour ajouter un style spécial à un élément qui est le premier enfant d'un autre élément.
- **:last-child** Utilisez cette classe pour ajouter un style spécial à un élément qui est le dernier enfant d'un autre élément.
- **:nth-child** Utilisez cette classe pour ajouter un style spécial à un élément qui est l'enfant n d'un autre élément. (e.g: **:nth-child(3)** signifie le troisième enfant d'un certain élément, **:nth-child(2n)** signifie que chaque deuxième élément : *2eme, 4eme, 6eme, 8eme, ... élément*)

CSS – Pseudo classes

```
<html>
  <head>
    <style>
      li:hover {
        background-color: azure;
      }

      li:first-child {
        background-color: aqua;
      }

      li:last-child {
        background-color: lavender;
      }

      li:nth-child(3) {
        background-color: chartreuse;
      }
    </style>
  </head>
  <body>
    <ol>
      <li>Page d'accueil</li>
      <li>Liste des produits</li>
      <li>Qui sommes-nous ?</li>
      <li>Contactez nous</li>
    </ol>
  </body>
</html>
```

1. [Page d'accueil](#)
2. [Liste des produits](#)
3. [Qui sommes-nous ?](#)
4. [Contactez nous](#)



CSS – Disposition des pages, cadrage (wireframing) - Header, aside, section, footer

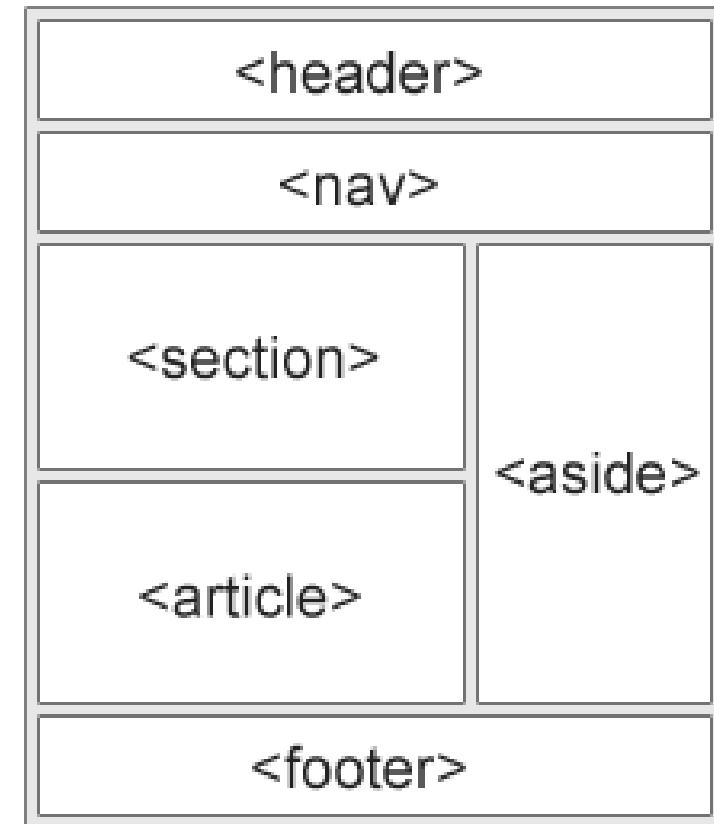
Un élément sémantique décrit clairement sa signification tant pour le navigateur que pour le développeur.

Exemples d'éléments non sémantiques : <div> et - Ne dit rien sur son contenu.

Exemples d'éléments sémantiques : <form>, <p>, et <article> - Définit clairement son contenu.

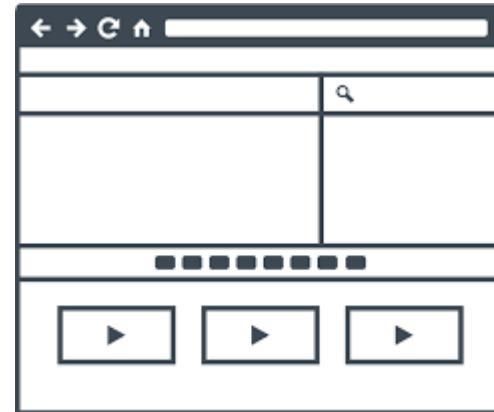
En HTML, certains éléments sémantiques peuvent être utilisés pour définir différentes parties d'une page web :

- <article> : Définit un article.
- <aside> : Définit un contenu hors du contenu de la page.
- <footer> : Définit un pied de page pour un document ou une section.
- <header> : Spécifie un en-tête pour un document ou une section.
- <main> : Précise le contenu principal d'un document.
- <nav> : Définit les liens de navigation.
- <section> : Définit une section dans un document.
- ...



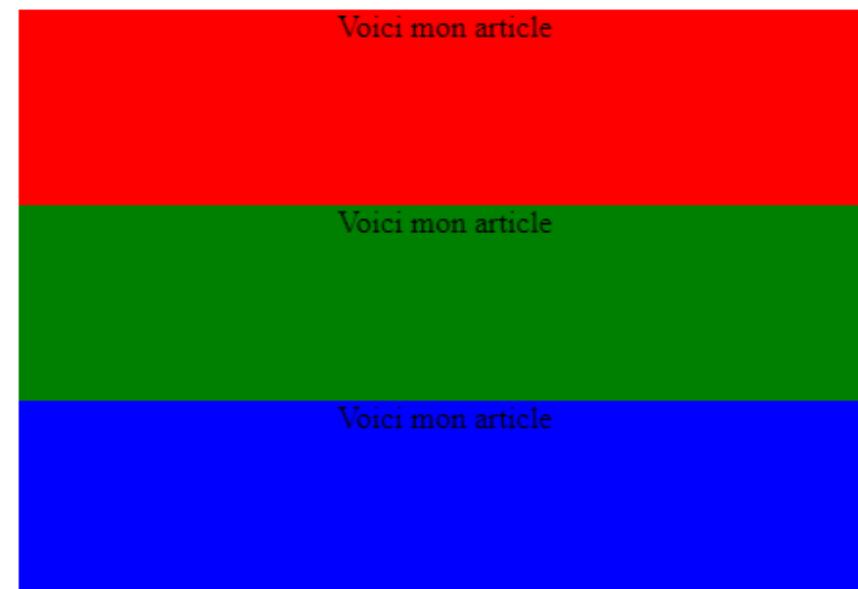
CSS – Cadrage (wireframing)

La règle de base à suivre pour la première fois lors de l'encadrement de votre page web.
Mettez des éléments de bloc les uns sur les autres et des éléments de bloc en ligne
les uns à côté des autres.



Dans le premier cas, la plupart des balises ont `display: block` par défaut, ce n'est donc pas un problème. Mais pour le second cas, il existe de nombreuses solutions que vous pouvez utiliser. La plus simple consiste à utiliser la FlexBox. D'autres solutions consistent à utiliser `display: inline-block` ou d'autres affichages.

```
article {  
    display: block;  
  
    text-align: center;  
}
```

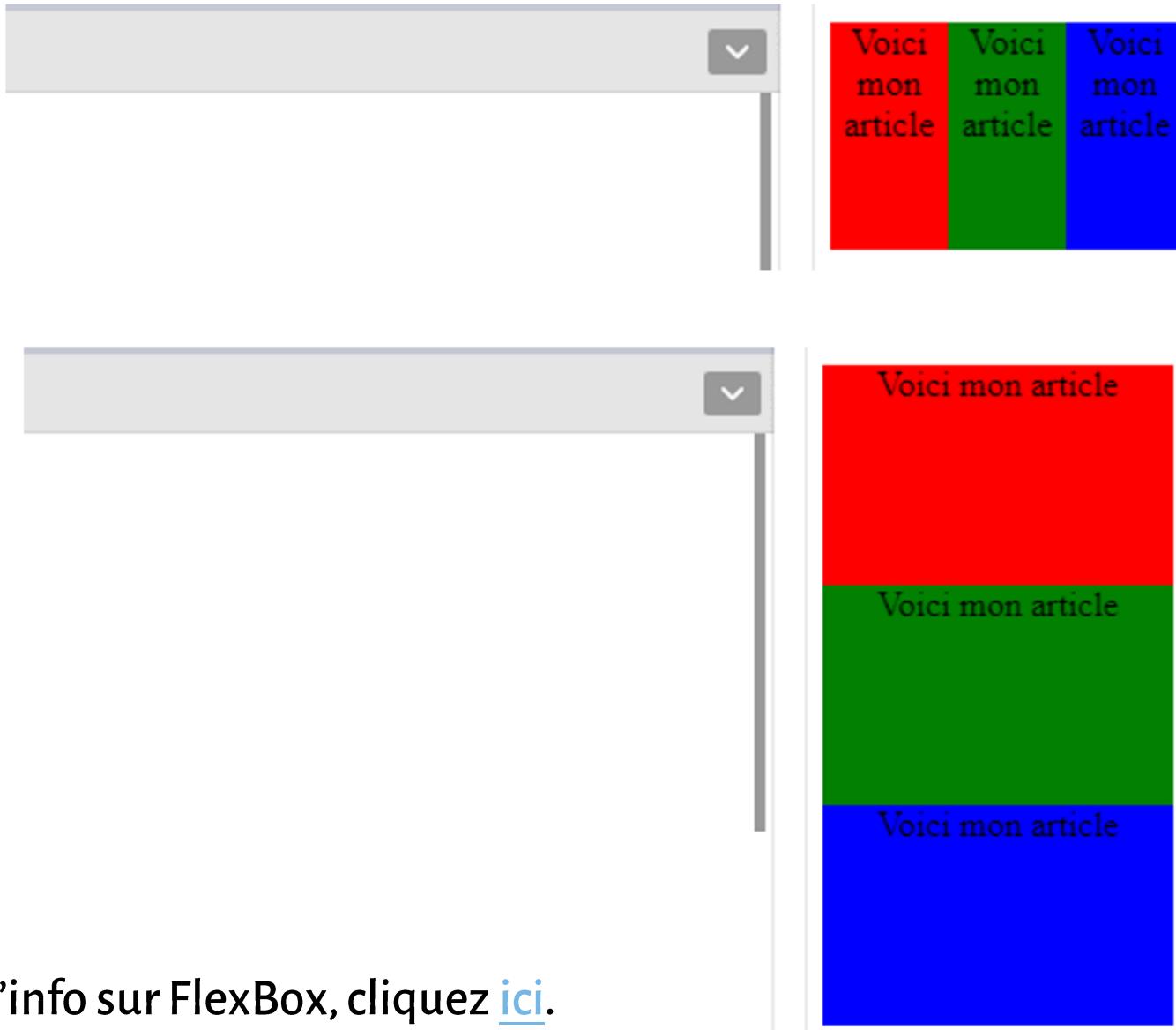


CSS – Cadrage (wireframing) FlexBox - Éléments adjacents

```
body {  
    display: flex;  
}  
  
article {  
    flex: 1; /* Régler l'élément pour  
qu'il grandisse et s'étire */  
    text-align: center;  
}
```

```
body {  
    display: flex;  
    flex-wrap: wrap;  
}
```

```
article {  
    flex: 1;  
    text-align: center;  
}
```



Pour plus d'info sur FlexBox, cliquez [ici](#).

CSS–Transitions

Les transitions CSS vous permettent de modifier les valeurs de propriété d'une manière douce (d'une valeur à une autre), sur une durée donnée.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1 {
        text-align: center;
        transition: word-spacing 5s, color 3s;
      }

      h1:hover {
        word-spacing: 50px;
        color: bisque;
      }
    </style>
  </head>
  <body>
    <h1>Titre de la page</h1>
  </body>
</html>
```

Titre de la page

Essayer de survoler l'en-tête avec le curseur.

CSS–Animations

Les animations CSS permettent l'animation de la plupart des éléments HTML

```
<!DOCTYPE html><html>
<head>
<style>
@keyframes entete {
  0% { background-color: red; }

  25% { background-color: yellow; }

  50% { background-color: blue; }

  100% { background-color: red; }
}

h1 {
  background-color: red;
  text-align: center;
  animation: entete 3s;
}
</style>
</head>
<body> <h1>Titre de la page</h1> </body></html>
```



CSS – Viewport (Layout width vs Device Width)

Si vous regardez la source HTML d'une page, vous verrez généralement la balise <meta> suivante dans le <head> du document :

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

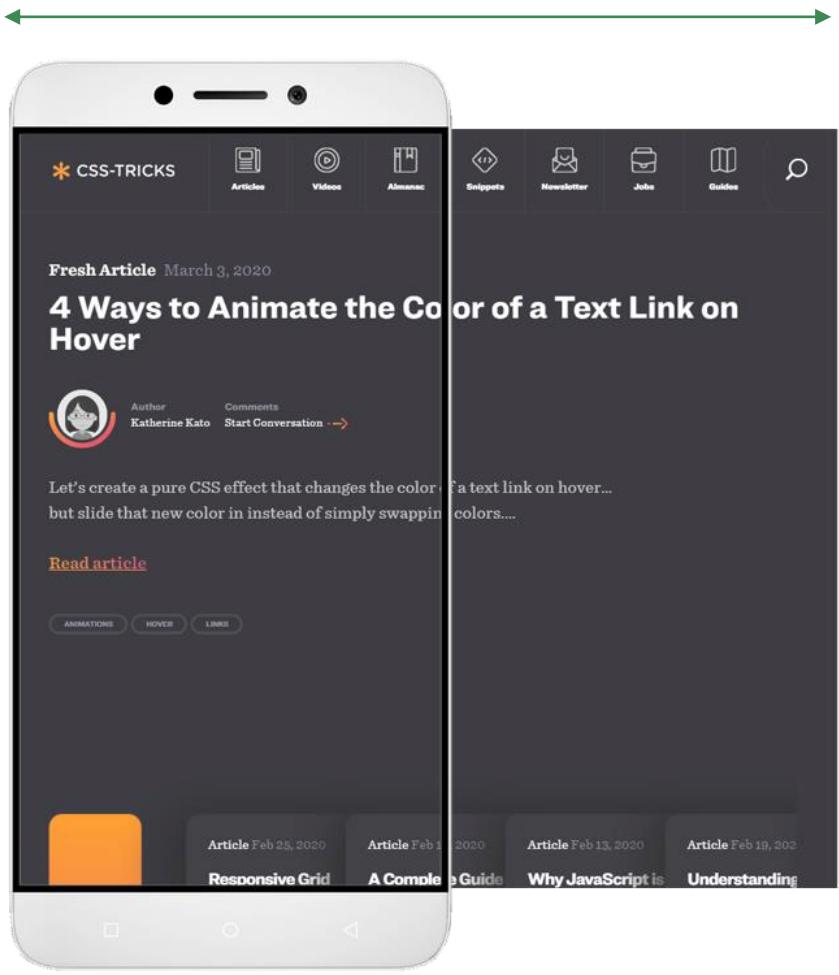
Cette méta-tag indique aux navigateurs mobiles qu'ils doivent régler la largeur de la fenêtre de visualisation en fonction de la largeur de l'appareil, et mettre le document à l'échelle à 100% de sa taille prévue, ce qui permet d'afficher le document à la taille optimisée pour le mobile que vous avez prévue.

Vous pourriez penser que les navigateurs mobiles rendent votre page en utilisant la largeur de l'appareil disponible. Mais non, les navigateurs mobiles rendent toutes les pages à une largeur fixe, quel que soit le téléphone utilisé. Chaque navigateur a sa propre largeur définie. Cette balise méta indique au navigateur de simplement appliquer la largeur de l'appareil à la page.

Vous devez donc toujours inclure la ligne HTML ci-dessus dans l'en-tête de vos documents.

CSS – Viewport (Layout width vs Device Width)

Layout width ~ 960px



Layout width = 320px



Device width = 320px

Device width = 320px

CSS – Media Queries

Maintenant, comment pouvez-vous ajouter des règles CSS basées sur la largeur de l'écran. Pour faire quelque chose quand l'écran est grand et un autre penser quand l'écran est petit.
Ici viennent les requêtes médias.

```
@media only screen and (max-width: 768px) {  
    /* mobile phones: */  
}
```

```
@media only screen and (min-width: 768px) {  
    /* Écrans desktop: */  
}
```