

# PROGRAMMATION ORIENTÉE OBJET ET WEB

Pr. OMARI Kamal

ENSA KHOURIBGA

28 février 2025

# Chapitre 3 : Exploration du Langage CSS

- 1 Introduction au langage de style CSS
  - Présentation du CSS
  - Syntaxe du CSS
  - Unités de mesure
  - Positions (center, left, right)
  - Fonts
  - Sélecteurs simples (élément, class, id)
  - Sélecteurs complexes
  - Pseudo classes
- 2 Utiliser les propriétés CSS
  - Typographie
  - Bordures et ombres
  - Marges et padding
  - Images
  - Couleurs de fond
  - background
  - Types de positionnement (relatif, absolu, float...)
- 3 Adapter une page web au dispositif d'affichage
  - Introduction au Responsive Design
  - Media Queries
- 4 Créer des animations
  - Manipulation des propriétés d'animation
  - Types d'animation (transition, transformation...)
  - Utilisation des keyFrames

# Objectifs du cours

Dans ce chapitre, nous allons explorer le langage CSS, indispensable pour la mise en forme des pages web. Vous vous familiariserez avec les concepts clés de CSS, tels que les sélecteurs, les propriétés, et les règles de style. À la fin de ce chapitre, vous serez en mesure de styliser des pages web, de modifier l'apparence des éléments HTML, de gérer la disposition des contenus, et d'appliquer des bonnes pratiques pour assurer la compatibilité et l'optimisation des styles sur différents navigateurs.

# Présentation du CSS

Le langage CSS (Cascading Style Sheets) est conçu pour gérer la mise en forme visuelle d'une page web, et non son contenu ou sa structure, qui sont pris en charge par HTML.

- CSS permet de styliser divers éléments tels que le texte (couleur, taille, casse, etc.), les images, les tableaux et les formulaires.
- Il offre aussi des outils pour dimensionner et positionner les éléments sur la page, afin d'améliorer l'apparence et l'organisation du contenu.
- CSS est utilisé pour créer des animations et des effets interactifs, ajoutant ainsi du dynamisme aux pages web.

## Application d'un style à un code HTML

Il existe 3 manières pour introduire le style à une balise donnée :

- **Directement dans une balise, avec l'attribut style :**

Cette méthode est appelée "style en ligne". Elle est pratique pour des modifications ponctuelles, mais elle peut rendre le code plus difficile à maintenir si utilisée à grande échelle.

### Exemple

```
<p style="color: red;">Texte en rouge</p>
```

# Application d'un style à un code HTML

- Dans l'en-tête (head), après la balise title :  
Ce style est appelé "style interne" ou "embedded style". Il est défini dans la balise <style> dans la section <head>.

## Exemple

```
<head>  
<title>Page</title>  
<style>  
p {color: blue;}  
</style>  
</head>
```

## Dans un fichier séparé avec une extension .css :

- **Dans l'en-tête (head), après la balise title :**  
Cette méthode est appelée "style externe" et est la technique recommandée. Elle permet de séparer le style de la structure HTML, ce qui facilite la maintenance et l'application cohérente des styles sur plusieurs pages.

### Exemple

```
<link rel="stylesheet" href="styles.css">
```

Introduction au langage de style CSS  
Utiliser les propriétés CSS  
Adapter une page web au dispositif d'affichage  
Créer des animations

Présentation du CSS  
Syntaxe du CSS  
Unités de mesure  
Positions (center, left, right)  
Fonts  
Sélecteurs simples (élément, class, id)  
Sélecteurs complexes  
Pseudo classes

## Meilleure pratique et conformité aux normes W3C

La troisième méthode, qui consiste à utiliser un fichier CSS externe pour définir les styles, est non seulement la meilleure pratique, mais aussi pleinement conforme aux normes du W3C (World Wide Web Consortium). Elle favorise une organisation propre du code et une gestion plus efficace des styles dans les projets de grande envergure, tout en étant la méthode la plus recommandée par les bonnes pratiques du W3C.



# Syntaxe du CSS

La syntaxe du CSS (Cascading Style Sheets) se compose de règles de style appliquées à des éléments HTML. Chaque règle CSS est constituée de deux parties principales : un sélecteur et un bloc de déclaration.

## Exemple

```
sélecteur {  
propriété: valeur;  
}
```

- **Sélecteur** : Identifie l'élément HTML auquel le style sera appliqué (par exemple, un paragraphe <p>, un titre <h1>, une classe .classname, etc.).
- **Propriété** : Définit l'aspect que vous souhaitez modifier (comme color, font-size, margin, etc.).
- **Valeur** : Spécifie la valeur de la propriété (comme red, 16px, 10px).

# Exemple

## Exemple

```
p {  
  color: blue;  
  font-size: 16px;  
  margin: 10px;  
}
```

Dans cet exemple :

- Le sélecteur p cible tous les éléments `<p>` (paragraphe) de la page.
- Les propriétés color, font-size, et margin sont définies avec des valeurs respectives de blue, 16px, et 10px.

Introduction au langage de style CSS  
Utiliser les propriétés CSS  
Adapter une page web au dispositif d'affichage  
Créer des animations

Présentation du CSS  
Syntaxe du CSS  
Unités de mesure  
Positions (center, left, right)  
Fonts  
Sélecteurs simples (élément, class, id)  
Sélecteurs complexes  
Pseudo classes

## Codification des couleurs (Rappel)

Méthode	Exemple
Couleurs nommées	background-color: grey
Couleurs en hexadécimal	background-color: #808080
Couleurs décomposées	background-color: RGB(128,128,128)

Table 1 – Méthodes de codification des couleurs

# Les unités absolues

Les unités absolues ne sont influencées par aucune autre dimension et sont utilisées pour exprimer des longueurs, telles que le padding, la hauteur d'un bloc, ou la taille des caractères. Les principales unités absolues incluent :

- Pixels (px)
- Pouces (in)
- Centimètres (cm)
- Millimètres (mm)
- Picas (pc)
- Points (pt)

Parmi ces unités, le pixel et le point sont les plus couramment utilisés.

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

# Relation entre les Unités de Mesure

## Relation entre les Unités de Mesure

- 1 pouce (in) = 96 pixels (px)
- 1 centimètre (cm) = 37,8 pixels (px)
- 1 millimètre (mm) = 3,78 pixels (px)

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

# Exemple

## Exemple

```
div {  
width : 150px ;  
height : 150px ;  
}
```

# Les unités relatives au texte

Les unités relatives au texte sont particulièrement utiles pour créer des mises en page flexibles et adaptatives. Contrairement aux unités absolues, elles s'ajustent en fonction de la taille de l'élément parent ou de la taille de la police par défaut. Les principales unités relatives au texte incluent :

- **em** : Cette unité est relative à la taille de police de l'élément parent. Par exemple, si la taille de police d'un élément parent est de 16 pixels, 1em correspond à 16 pixels.
- **rem** : Cette unité est relative à la taille de police de l'élément racine (généralement `<html>`). Par défaut, 1rem correspond à la taille de police définie dans le `<html>`, ce qui est souvent 16 pixels. Utiliser rem garantit une plus grande cohérence dans toute l'application.
- **% (pourcentage)** : Les tailles en pourcentage sont relatives à la taille de police de l'élément parent. Par exemple, si un élément a une taille de police de 100 % (ou 1em), il sera de la même taille que la taille de police de son parent.

# Les unités relatives au viewport

Les unités relatives au viewport permettent de définir des tailles en fonction des dimensions de la fenêtre d'affichage (viewport) du navigateur. Cela facilite la création de mises en page réactives qui s'adaptent à différentes tailles d'écran. Les principales unités relatives au viewport incluent :

- vw (viewport width) : Cette unité représente 1 % de la largeur du viewport. Par exemple, si la largeur de la fenêtre est de 1000 pixels, 1vw équivaut à 10 pixels.
- vh (viewport height) : Cette unité représente 1 % de la hauteur du viewport. Par exemple, si la hauteur de la fenêtre est de 800 pixels, 1vh correspond à 8 pixels.
- vmin : Cette unité représente 1 % de la dimension la plus petite entre la largeur et la hauteur du viewport. Par exemple, si la largeur est de 1000 pixels et la hauteur de 800 pixels, 1vmin équivaut à 8 pixels.
- vmax : Cette unité représente 1 % de la dimension la plus grande entre la largeur et la hauteur du viewport. Par exemple, dans le même



## Alignement du texte

L'alignement du texte est essentiel pour la présentation et la lisibilité du contenu sur une page web. En CSS, plusieurs propriétés permettent de contrôler l'alignement du texte. Les principales incluent :

- **text-align** : Cette propriété définit l'alignement horizontal du texte à l'intérieur d'un élément. Les valeurs possibles sont :
  - **left** : Aligne le texte à gauche (valeur par défaut pour les éléments block).
  - **right** : Aligne le texte à droite.
  - **center** : Centre le texte horizontalement.
  - **justify** : Justifie le texte, en étirant les lignes pour que le texte soit aligné à la fois à gauche et à droite.

Introduction au langage de style CSS  
Utiliser les propriétés CSS  
Adapter une page web au dispositif d'affichage  
Créer des animations

Présentation du CSS  
Syntaxe du CSS  
Unités de mesure  
**Positions (center, left, right)**  
Fonts  
Sélecteurs simples (élément, class, id)  
Sélecteurs complexes  
Pseudo classes

# Alignement du texte

## Exemple

```
p {text-align: center; }
```

## Frame Title

**vertical-align** : Cette propriété contrôle l'alignement vertical du texte à l'intérieur d'un élément en ligne ou d'une cellule de tableau. Les valeurs possibles incluent :

- **baseline** : Aligne le texte sur la ligne de base (valeur par défaut).
- **top** : Aligne le texte en haut de l'élément.
- **middle** : Aligne le texte au centre de l'élément.
- **bottom** : Aligne le texte en bas de l'élément.

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

**Positions** (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

# Exemple

## Exemple

```
td {vertical-align: middle; }
```

# Orientation du Texte en CSS

L'orientation du texte en CSS permet de modifier l'angle et la direction dans laquelle le texte est affiché. Cela peut être utile pour des mises en page créatives, des designs particuliers, ou pour améliorer la lisibilité dans certaines situations. Les principales propriétés utilisées pour contrôler l'orientation du texte incluent :

- **writing-mode** : Cette propriété définit la direction dans laquelle le texte est écrit. Les valeurs possibles sont :
  - **horizontal-tb** : Texte écrit de gauche à droite et de haut en bas (valeur par défaut pour les langues comme l'anglais).
  - **vertical-rl** : Texte écrit de haut en bas, en commençant par la droite (utilisé dans certaines langues asiatiques).
  - **vertical-lr** : Texte écrit de haut en bas, en commençant par la gauche.

Introduction au langage de style CSS  
Utiliser les propriétés CSS  
Adapter une page web au dispositif d'affichage  
Créer des animations

Présentation du CSS  
Syntaxe du CSS  
Unités de mesure  
**Positions (center, left, right)**  
Fonts  
Sélecteurs simples (élément, class, id)  
Sélecteurs complexes  
Pseudo classes

# Exemple

## Exemple

```
p {writing-mode: vertical-rl; }
```

## Direction en CSS

Les propriétés direction et unicode-bidi en CSS sont essentielles pour gérer le texte multilingue, notamment lorsque vous travaillez avec des langues ayant une direction d'écriture de droite à gauche (RTL) comme l'arabe ou l'hébreu.

### Propriété direction

La propriété direction détermine la direction du texte dans un élément. Les valeurs possibles sont :

- ltr (left to right) : Texte écrit de gauche à droite (valeur par défaut).
- rtl (right to left) : Texte écrit de droite à gauche.

# Fonts

Les polices (ou fonts) sont un élément crucial du design web, influençant non seulement l'esthétique d'un site, mais aussi sa lisibilité et son accessibilité. En CSS, plusieurs propriétés permettent de contrôler l'apparence des polices. Voici les principales :



# Propriétés de base des polices

font-family : Cette propriété définit la famille de polices à utiliser pour un élément. Vous pouvez spécifier une liste de polices, séparées par des virgules, et inclure une police générique à la fin. Par exemple :

## Exemple

```
body {font-family: "Arial", "Helvetica", sans-serif; }
```

# Propriétés de base des polices

font-size : Cette propriété contrôle la taille de la police. Les tailles peuvent être définies en unités absolues (comme px) ou relatives (comme em ou rem). Par exemple :

## Exemple

```
h1 {font-size: 2rem; /* 2 fois la taille de la police de  
l'élément racine */ }
```

# Propriétés de base des polices

font-weight : Cette propriété définit l'épaisseur de la police. Les valeurs courantes incluent :

- normal : Épaisseur normale.
- bold : Épaisseur en gras.
- bolder : Plus épais que bold.
- lighter : Plus léger que normal.
- Valeurs numériques de 100 à 900 (100 étant le plus léger et 900 le plus épais).

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

# Propriétés de base des polices

font-style : Cette propriété permet de définir le style de la police. Les valeurs possibles incluent :

- normal : Style normal (valeur par défaut).
- italic : Italique.
- oblique : Oblique.

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

# Sélecteurs Simples en CSS

Les sélecteurs en CSS permettent de cibler des éléments HTML pour leur appliquer des styles. Parmi les différents types de sélecteurs, les sélecteurs simples sont les plus fondamentaux. Ils se divisent principalement en trois catégories : les sélecteurs d'élément, de classe et d'ID.

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

# Sélecteur d'Élément

Le sélecteur d'élément, également appelé sélecteur de type, cible tous les éléments d'un type donné. Par exemple, pour appliquer un style à tous les paragraphes `<p>` d'une page, vous pouvez utiliser :

## Exemple

```
p {color: blue;  
font-size: 16px;  
}
```

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

# Sélecteur de Classe

Le sélecteur de classe cible tous les éléments qui possèdent une certaine classe. Pour utiliser un sélecteur de classe, vous devez préfixer le nom de la classe avec un point (.). Par exemple, pour styliser tous les éléments ayant la classe important :

## Exemple

```
.important {font-weight: bold;  
background-color: yellow;  
}
```

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

# Sélecteur de Classe

Pour appliquer ce style, vous pouvez l'utiliser dans votre HTML comme suit :

## Exemple

```
<p class="important">Ce texte est important.</p>
```



# Exemple

## Exemple.html

```
<table>
<tr><th>Nom</th><th>prénom</th><th>Age</th></tr>
<tr class="impaire"><td>MANSOURI</td><td>Hassan</td><td>23</td></tr>
<tr class="paire"><td>SAFIR</td><td>Laila</td><td>22</td></tr>
<tr class="impaire"><td>BICHRI</td><td>Karim</td><td>21</td></tr>
<tr class="paire"><td>HAMIM</td><td>mohamed</td><td>24</td></tr>
</table>
```

## Exemple.css

```
.impairebackground-color : rgba(255, 255, 0, 0.2);
```

# Sélecteur d'ID

Le sélecteur d'ID cible un élément unique ayant un ID spécifique. Pour utiliser un sélecteur d'ID, vous devez préfixer le nom de l'ID avec un dièse (#). Par exemple, pour styliser un élément avec l'ID header :

## Exemple

```
#header {font-size: 24px;  
text-align: center;  
}
```

Dans le HTML, vous l'utiliseriez de cette manière :

## Exemple

```
<h1 id="header">Bienvenue sur mon site</h1>
```

Introduction au langage de style CSS  
Utiliser les propriétés CSS  
Adapter une page web au dispositif d'affichage  
Créer des animations

Présentation du CSS  
Syntaxe du CSS  
Unités de mesure  
Positions (center, left, right)  
Fonts  
Sélecteurs simples (élément, class, id)  
Sélecteurs complexes  
Pseudo classes

# Exemple

## Exemple.html

```
<table border="1">
<tr><th>Nom</th><th>prénom</th><th>Age</th></tr>
<tr><td>MANSOURI</td><td>Hassan</td><td>23</td></tr>
<tr><td>SAFIR</td><td>Laila</td><td>22</td></tr>
<tr id="sel"><td>BICHRI</td><td>Karim</td><td>21</td></tr>
<tr><td>HAMIM</td><td>mohamed</td><td>24</td></tr>
</table>
```

## Exemple.css

```
#selbackground-color : rgba(255, 255, 0, 0.2);
```

## Sélecteurs complexes

Les sélecteurs complexes en CSS permettent de cibler des éléments de manière plus spécifique et flexible que les sélecteurs simples. Ils sont utiles pour appliquer des styles à des éléments en fonction de leur relation avec d'autres éléments, leur position dans le DOM ou des conditions particulières. Les principaux sélecteurs complexes incluent les sélecteurs combinés, les sélecteurs de descendants, de frères, et les sélecteurs d'attributs.

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

**Sélecteurs complexes**

Pseudo classes

## Sélecteur de Descendant

Le sélecteur de descendant cible un élément en fonction de sa relation hiérarchique avec un autre élément. Ce sélecteur s'écrit en séparant deux éléments par un espace. Par exemple, pour cibler tous les `<p>` qui se trouvent à l'intérieur d'une `<div>` :

Exemple.css

```
div p {color: red;}
```

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

**Sélecteurs complexes**

Pseudo classes

## Sélecteur Enfant Direct (>)

Le sélecteur enfant direct cible uniquement les éléments qui sont des enfants directs d'un autre élément. Par exemple, pour cibler les `<li>` qui sont des enfants directs d'une `<ul>` :

Exemple.css

```
ul > li {margin-bottom: 10px; }
```

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

**Sélecteurs complexes**

Pseudo classes

## Sélecteur de Frère Adjacent (+)

Le sélecteur de frère adjacent cible un élément qui est immédiatement précédé par un autre élément. Par exemple, pour cibler un paragraphe qui suit immédiatement un `<h2>` :

Exemple.css

```
h2 + p {font-size: 18px; }
```

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

**Sélecteurs complexes**

Pseudo classes

## Sélecteur de Frère Général (~)

Le sélecteur de frère général cible tous les éléments frères qui suivent un élément donné, pas uniquement le premier. Par exemple, pour cibler tous les `<p>` qui suivent un `<h2>` dans le même parent :

Exemple.css

```
h2 ~p {color: gray; }
```



## Sélecteur d'Attribut

Le sélecteur d'attribut cible un élément en fonction de la présence ou de la valeur d'un attribut donné. Par exemple, pour cibler tous les liens (<a>) qui contiennent un attribut `target="_blank"` :

### Exemple.css

```
a[target="_blank"] {color: green; }
```

Il existe plusieurs variantes des sélecteurs d'attributs :

- Présence d'attribut : `[attribute]` (ex : `[disabled]`)
- Valeur exacte : `[attribute="value"]` (ex : `[type="text"]`)
- Valeur commençant par : `[attribute^="value"]`
- Valeur se terminant par : `[attribute$="value"]`
- Valeur contenant : `[attribute="value"]`

## Sélecteurs de Pseudo-classes

Les pseudo-classes permettent de cibler des éléments dans des états spécifiques ou en fonction de leur position dans le DOM. Quelques exemples courants incluent :

:hover : cible un élément lorsque la souris passe dessus.

### Exemple.css

```
a:hover {color: red; }
```

Cela change la couleur du lien en rouge lorsque la souris est placée dessus.

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

# Sélecteurs de Pseudo-classes

`:nth-child(n)` : cible un élément en fonction de sa position dans un parent.

Exemple.css

```
li:nth-child(2) {font-weight: bold; }
```

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

# Sélecteurs de Pseudo-classes

:first-child : cible le premier enfant d'un élément.

Exemple.css

```
p:first-child {margin-top: 0; }
```

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Présentation du CSS

Syntaxe du CSS

Unités de mesure

Positions (center, left, right)

Fonts

Sélecteurs simples (élément, class, id)

Sélecteurs complexes

Pseudo classes

## Sélecteurs de Pseudo-classes

`:focus` Cible un élément lorsqu'il est en focus (par exemple, lorsqu'un utilisateur clique sur un champ de formulaire ou navigue avec le clavier).

### Exemple.css

```
input:focus {border-color: blue; }
```

Cela change la couleur de la bordure en bleu lorsque l'utilisateur clique dans un champ `<input>`.

## Propriétés du texte en CSS3

- Propriété font-style : spécifie un texte en italique. Valeurs possibles : normal, italic, oblique.
- Propriété font-variant : spécifie si un texte doit être affiché en petites capitales. Valeurs possibles : normal, small-caps.
- Propriété font-weight : permet de mettre en gras un texte. Valeurs possibles : de 100 à 900, normal, bold, bolder, lighter.
- Propriété white-space : spécifie l'espace blanc à l'intérieur d'un élément. Valeurs possibles : normal, pre, nowrap, pre-wrap, pre-line

## Propriétés du texte en CSS3

- Propriété word-spacing : l'espace entre les mots (en pixels).
- Propriété letter-spacing : l'espace entre deux lettres au sein d'un même mot (en pixels).
- Propriété text-decoration : spécifie la décoration du texte.  
Valeurs possibles : none, underline, overline , line-through, blink.
- Propriété text-transform : permet de contrôler la capitalisation du texte. Valeurs possible : none, uppercase (majuscules), lowercase (miniscule), capitalize (la première lettre de chaque mot sera en majuscule).

## Propriétés du texte en CSS3

- Propriété text-align : permet d'aligner horizontalement le texte. Valeurs possibles : left, center, right, justify
- Propriété text-indent : taille du retrait de première ligne (en pixels)
- Propriété line-height : hauteur de ligne (en pixels).
- Propriété vertical-align : permet d'aligner verticalement le texte. Valeurs possibles : baseline, super, top, text-top, middle, text-bottom, bottom, sub, inherit.



# Propriétés du texte en CSS3

## Exemple.css

```
a {  
text-decoration: none; /* Les liens ne seront plus soulignés */  
color: red; /* Les liens seront en rouge au lieu de bleu */  
font-style: italic; /* Les liens seront en italique */  
}  
a:hover /* Quand le visiteur pointe sur le lien */  
{  
text-decoration: underline; /* Le lien deviendra souligné quand  
on pointera dessus */  
color: green; /* Le lien sera écrit en vert quand on pointera  
dessus */  
background-color: #CFE1EB; /* Le fond du paragraphe change de  
couleur */  
}
```

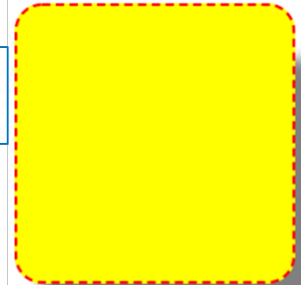
# Bordures et ombres

```
<div id="myDiv1"></div>
```

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px dashed red;  
  border-radius: 10px;  
  box-shadow: 10px 10px 5px gray;  
}
```

Ligne avec tirets.  
Autres types :  
Solid, dotted, ...

Rayon de l'arrondi



Distance horizontale

Rayon de l'arrondi

```
.class{  
  text-shadow: 2px 2px 3px #FFF;  
}
```

Distance verticale

Couleur de l'ombre

# Marges et padding

La propriété `margin` (retrait extérieur) définit la taille des marges sur les quatre côtés de l'élément.

C'est une propriété raccourcie qui permet de manipuler les autres propriétés de marges : `margin-top`, `margin-right`, `margin-bottom` et `margin-left`.

## Marges et padding

### Exemple.html

```
<div class="ex1">
margin : auto;
background : gold;
width : 66%;
</div>
<div class="ex2">
margin : 20px 0px 0px -20px;
background : gold;
width : 66%;
</div>
```

### style.css

```
.ex1 {
margin : auto;
background : gold;
width : 66%;
}
.ex2 {
margin : 20px 0px 0 -20px;
background : gold;
width : 66%;
}
```

# Marges et padding

La propriété padding (retrait intérieur) définit les différents écarts de remplissage sur les quatre côtés d'un élément (cf. les boîtes CSS). Elle synthétise padding-top, padding-right, padding-bottom, padding-left.

# Marges et padding

## Exemple.html

```
<h4>Bonjour le monde !</h4>  
<h3>Le remplissage n'est pas  
le même ici.</h3>
```

## Style.css

```
h4 {  
  background-color : green;  
  padding : 50px 20px 20px  
  50px;  
}  
h3 {  
  background-color : blue;  
  padding : 400px 5%;  
}
```

# Propriétés des images

- Propriétés width et height : définissent la taille de l'image
- Propriété max-width : permet à l'image de s'adapter à la taille de l'écran (Exemple 1)
- Propriétés margin-left et margin-right : ayant la valeur auto, ils permettent de centrer une image (Exemple 2)
- Propriété opacity : gère l'opacité de l'image (Exemple 3)
- Propriété filter : ajoute des effets visuels (comme le flou et la saturation) à l'image (Exemple 4).

# Propriétés des images

## Exemple 1

```
#img {max-width: 100%;  
height: auto;  
}
```

## Exemple 2

```
#img {  
display: block;  
margin-left: auto;  
margin-right: auto;  
width: 50%;  
}
```



Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Typographie

Bordures et ombres

Marges et padding

Images

Couleurs de fond

background

Types de positionnement (relatif, absolu, float...)

# Propriétés des images

## Exemple 3

```
#img {  
  Opacity : 0.5;  
}
```

## Exemple 4

```
#img  
{  
  filter: grayscale(100%);  
}
```

## Couleurs de fond

Propriété color définit la couleur du texte

Propriété background-color : définit la couleur du fond ;

### Exemple.html

```
<p>Ceci est un texte colorié  
en bleu</p>
```

### Exemple.css

```
p{color : blue;  
font-size : 24px;  
font-family : cursive;  
background-color : yellow;  
}
```

# background

L'élément image peut être inséré comme arrière plan d'un élément. Les propriétés associées sont :

- background-image : url de l'image de fond.
- background-repeat : mode de réplication de l'image de fond (repeat, repeat-x, repeat-y, no-repeat).
- background-attachment : défilement ou non de l'image de fond (scroll, fixed).
- background-position : position de l'image d'arrière plan (top, bottom, left, center, right).
- background-size (100% 100%)

# background

## Exemple.html

```
<div id="myDiv1"></div>
```

## Exemple.css

```
#myDiv1{  
height : 100px;  
width : 100px;  
border : 1px solid black;  
background-image :  
url("Test.jpg");  
background-size :100%;  
background-repeat :no-repeat;  
}
```

# Types de positionnement

Le principe du positionnement en CSS permet de définir l'emplacement des boîtes générées pour chaque élément de la page.

Il existe trois mécanismes de positionnement de base en CSS :

- Le flux normal : consiste à garder le flux normal des éléments selon leur ordre et leur type (bloc , inline).
- La propriété «position»
  - Relative
  - Absolue
  - Fixe
  - Static
- La propriété «float»
  - Définit la position flottante d'un élément à droite (valeur right) ou à gauche (valeur left) dans un document (ou dans la boîte conteneur).
  - Nettoyer les flottants : propriété clear

# Le positionnement relatif

On place une div en position relative avec la valeur de la propriété : relative.

La position relative d'un élément (div par exemple) est calculée d'après sa position originale dans la page (sa position dans le flux normal).

l'élément peut être déplacé vers la droite, la gauche, le haut ou le bas (top, left, bottom et right).

Introduction au langage de style CSS  
Utiliser les propriétés CSS  
Adapter une page web au dispositif d'affichage  
Créer des animations

Typographie  
Bordures et ombres  
Marges et padding  
Images  
Couleurs de fond  
background  
Types de positionnement (relatif, absolu, float...)

# Le positionnement fixe

Pareil que le positionnement absolu, mais le block reste toujours visible, même si on défile la page.

# Le positionnement static

C'est de le comportement normal (par défaut) pour chaque élément de page. L'élément est alors positionné dans le flux avec sa position.

Les propriétés top, right, bottom, left et z-index ne s'appliquent pas.

- La valeur static est la valeur par défaut de la propriété position.
- Un élément HTML positionné avec position : static sera positionné selon le flux normal de la page.



# Le positionnement absolu

Ce type de positionnement permet de placer un élément de la page exactement à l'emplacement souhaité.

- Une balise HTML avec une position absolue ne laisse aucun espace vide après qu'elle est positionnée.
- On place une balise HTML en position absolue en fixant la valeur de la propriété position à "absolute".
- Les propriétés right, left, top et bottom sont alors utilisées pour définir l'emplacement de la balise HTML.

Introduction au langage de style CSS  
Utiliser les propriétés CSS  
Adapter une page web au dispositif d'affichage  
Créer des animations

Typographie  
Bordures et ombres  
Marges et padding  
Images  
Couleurs de fond  
background  
Types de positionnement (relatif, absolu, float...)

# Exemple

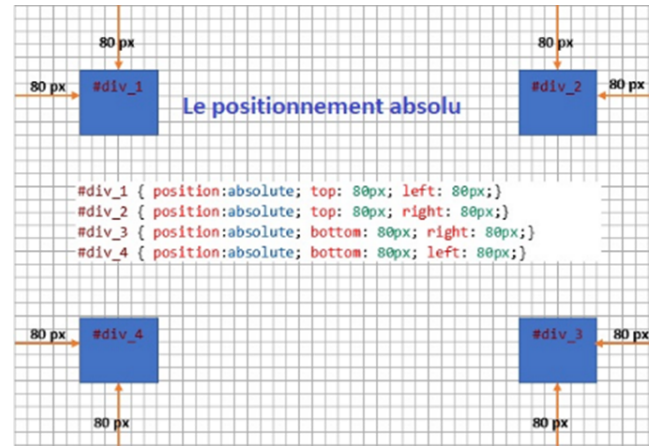


Figure 2 – Exemple

## La propriété «float»

La propriété float est utilisée pour positionner et formater le contenu. Par exemple, faire flotter une image à gauche du texte dans un conteneur.

La propriété float peut avoir les valeurs suivantes :

- left - L'élément flotte à gauche de son conteneur.
- right - L'élément flotte à droite de son conteneur.
- none - La valeur par défaut, l'élément ne flotte pas.
- inherit - L'élément hérite de la valeur flottante de son parent.

## Pourquoi utiliser Flexbox ?

Avant l'insertion de la méthode Flexbox, il existait quatre modes de mise en page :

- Block, pour les sections d'une page Web.
- Inline, pour le texte.
- Tableau, pour les données de tableau à deux dimensions.
- Position, pour la position explicite d'un élément.

La méthode Flexbox facilite la structuration et la mise en page souple et réactive, sans utiliser les marges et le positionnement.

La page est décomposée en un agencement de boîtes qui peuvent se suivre (boîtes voisines ) et s'imbriquer (boîtes ancêtres).

En définissant les tailles et les interactions entre les différentes boîtes, on obtient une adaptation automatique des éléments de la page lorsqu'elle est redimensionnée.

Introduction au langage de style CSS  
Utiliser les propriétés CSS  
Adapter une page web au dispositif d'affichage  
Créer des animations

Typographie  
Bordures et ombres  
Marges et padding  
Images  
Couleurs de fond  
background  
Types de positionnement (relatif, absolu, float...)

## Comment utiliser Flexbox ?

Le conteneur flex (flex container) ou bien la boîte dite «parent», contient les éléments flex (flex items) ou bien les boîtes «enfants».

L'idée est définir, en utilisant les différentes propriétés CSS, l'ordre des "boîtes enfants" dans la "boîte parent".

## Définir le conteneur flex (flex container)

Le conteneur est une balise html qui contient d'autres balises html.

Dans l'exemple ci-dessous, La conteneur correspond à la balise `<div>` pour laquelle on a défini la classe «.conteneur».

Cette balise `<div>` contient deux balises `<div>` et une balise `<figure>`. Ces dernières deviennent alors des "éléments flex" (flex items) mis en évidence ici par la classe ".element".

## Définir le conteneur flex (flex container)

### Exemple

```
<div class="conteneur">  
<div class="element">Élément 1</div>  
<figure class"element">Élément 2</figure>  
<div class"element">Élément 3</div>  
</div>
```

La boîte se transforme en "conteneur flex" dès lors qu'on lui attribue la propriété CSS "flex" comme montré dans le code CSS suivant :

### Exemple

```
.conteneur {  
display : flex;  
}
```

# Définir les axes du conteneur flex avec la propriété flex-direction

Axe principal : main axis.

Le conteneur flex possède un axe principal (main axis) sur lequel les éléments flex peuvent se suivre.

L'axe principal est défini par la propriété flex-direction qui peut prendre les valeurs suivantes :

- row : les éléments flex se suivent sur une ligne, ce qui correspond à la direction normale du sens d'écriture du document.
- row-reverse : les éléments flex se suivent sur une ligne en ordre inversé. ce qui correspond à la direction inverse du sens d'écriture du document.
- column : les éléments flex se suivent sur une colonne. column-reverse : les éléments flex se suivent sur une colonne en ordre inversé.



# Définir les axes du conteneur flex avec la propriété flex-direction

Axe secondaire : cross axis.

Le choix de l'axe principal impose la direction de l'axe secondaire (cross axis) qui lui est toujours perpendiculaire.

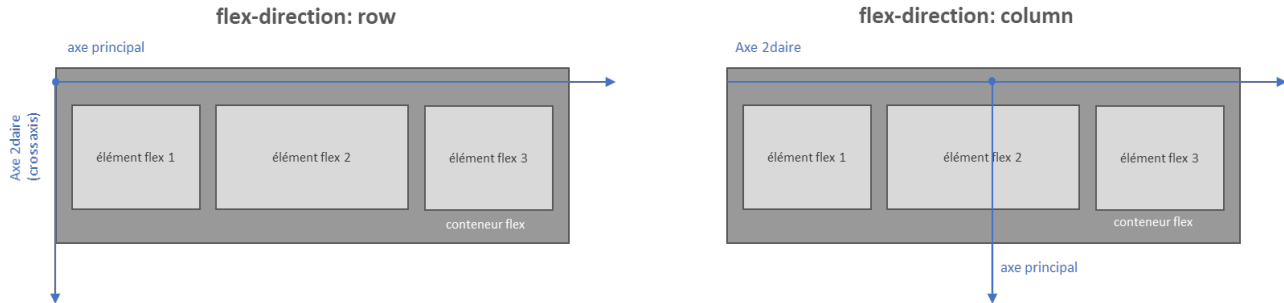


Figure 3 – Exemple de flex-direction en row et en column.

# Définir les axes du conteneur flex avec la propriété flex-direction

Le choix des axes est essentiel car leur orientation va déterminer le résultat de l'application des propriétés suivantes :

- justify-content : définit comment les éléments flex sont positionnés le long de l'axe principal ;
- align-items : définit comment les éléments flex sont positionnés le long de l'axe secondaire ;
- align-self : définit comment un seul élément flex est positionné le long de l'axe secondaire.

Introduction au langage de style CSS

Utiliser les propriétés CSS

Adapter une page web au dispositif d'affichage

Créer des animations

Typographie

Bordures et ombres

Marges et padding

Images

Couleurs de fond

background

Types de positionnement (relatif, absolu, float...)

## Positionner les éléments à l'intérieur d'un conteneur flex

Les éléments flexibles peuvent être alignés, justifiés et distribués le long des axes de leur conteneur grâce à différentes propriétés. Dans l'exemple suivant, on a choisi une direction "row", ce qui signifie que l'axe principal est l'axe horizontal et que l'axe secondaire est l'axe vertical.

## Positionner les éléments à l'intérieur d'un conteneur flex

### Exemple.html

```
<div class="conteneur">  
<div class="element1">élément  
1</div>  
<div class="element2">élément  
2</div>  
<div class="element3">élément  
3</div>  
<div class="element4">élément  
4</div>  
</div>
```

### Exemple.css

```
.conteneur {  
background-color : blue  
height : 230px;  
width : 500px;  
display : flex;  
flex-direction : row;  
align-items : stretch | flex-start  
| flex-end | center | baseline;  
justify-content : flex-start |  
flex-end | center | space-around |  
space-between | space-evenly;  
}
```

Pour les propriétés "align-items" et "justify-content", une seule valeur doit être sélectionnée.

# Introduction au Responsive Design

Les pages Web peuvent être consultées à l'aide de nombreux types d'appareils : ordinateurs de bureau, tablettes et téléphones. Ainsi, les pages Web ne doivent pas omettre d'informations pour s'adapter aux appareils plus petits, mais plutôt adapter leur contenu pour s'afficher correctement sur n'importe quel appareil utilisé.

La conception Web réactive (Responsive Design) rend la page Web compatible et adaptable à tous les appareils. Elle est basée uniquement sur HTML et CSS (elle n'est pas un programme ou un code JavaScript).

# Introduction au Responsive Design



Figure 4 – Principe du responsive design

Le Responsive web design consiste à utiliser les langages CSS et HTML pour redimensionner, masquer, réduire, agrandir ou déplacer le contenu d'une page pour s'adapter à l'écran d'affichage.

# Les media queries

La règle @media, introduite dans CSS2, permet de définir différentes règles de style pour différents types de médias.

Les medias queries dans CSS3 ont étendu l'idée des types de médias CSS2 : examiner la capacité du périphérique au lieu de chercher son type.

Les requêtes multimédias peuvent être utilisées pour vérifier de nombreuses propriétés, telles que :

- Largeur et hauteur de la fenêtre
- Largeur et hauteur de l'appareil
- Orientation (mode paysage ou portrait)
- Résolution

# Les media queries

L'exemple suivant change la couleur d'arrière-plan selon la règle suivante :

- Si la largeur de la fenêtre est 480 px au minimum, la couleur est le bleu clair.
- Si la largeur de la fenêtre est 200 px au minimum, la couleur est le lightgreen.
- Si la largeur de la fenêtre est inférieure à 200 px, la couleur reste blanche.



# Les media queries

## Exemple

```
@media screen and (min-width :200px){  
body {  
background-color :rgb(144,174,238);  
}  
}  
@mediascreen and (min-width :480px){  
body{  
background-color : lightgreen;  
}  
}
```

# Les media queries

Dans cet exemple, media queries est utilisée pour créer un menu de navigation réactif dont la conception varie selon les tailles d'écran.

## Exemple.html

```
<div class = "demo">
<a href = "#">Home</a>
<a href = "#">About</a>
<a href = "#">Tutorials</a>
<a href = "#">QA</a>
<a href = "#">Videos</a>
<a href = "#">Contact</a>
</div>
```

## Exemple.css

```
.demo {
overflow :hidden;
background-color :blue;
}
.demo a {
float : left;
display : block;
color : white;
text-align : center;
padding : 10px;
text-decoration : none;
}
@media screen and (max-width : 600px) {
.demo a {
float :none;
width :100%;
}
}
```

## Propriétés d'animation

La propriété raccourcie (ou bien les propriétés détaillées correspondantes ) est utilisée pour créer une animation CSS. Elle permet de configurer la durée, le minutage et d'autres détails de l'animation.

L'apparence visuelle de l'animation est définie en utilisant des règles CSS de mise en forme au sein de la règle @keyframes.

# Propriétés d'animation

Les propriétés détaillées rattachées à la propriété raccourcie animation sont :

- animation-delay : Cette propriété définit le délai entre le chargement de l'élément et démarrage de l'animation.
- animation-direction : Cette propriété précise si l'animation doit alterner entre deux directions de progression (faire des allers-retours) ou recommencer au début à chaque cycle de répétition.
- animation-duration : Cette propriété définit la durée d'un cycle de l'animation.
- animation-fill-mode : Cette propriété indique les valeurs à appliquer aux propriétés avant et après l'exécution de l'animation.

## Propriétés d'animation

- `animation-iteration-count` : Cette propriété détermine le nombre de répétition de l'animation. Pour répéter une animation infiniment, On utilise la valeur infinie.
- `animation-name` : Cette propriété déclare un nom à l'animation (utilisé comme référence à l'animation pour la règle `@keyframes` ).
- `animation-play-state` : Cette propriété permet d'interrompre (« pause ») ou de reprendre l'exécution d'une animation.
- `animation-timing-function` : Cette propriété configure la fonction de minutage d'une animation.

# Animations CSS

CSS permet l'animation d'éléments HTML sans utiliser JavaScript.

Une animation permet à un élément de passer progressivement d'un style à un autre en modifiant les propriétés CSS.

Les types d'animations en CSS :

- Les transformations
- Les transitions

# Les transformations 2D

Les transformations CSS 2D permettent de déplacer, faire pivoter, mettre à l'échelle et incliner des éléments.

La méthode `translate()`

- Exemple : déplacer l'élément `<div>` de 50 pixels vers la droite et de 100 pixels vers le bas par rapport à sa position actuelle :

## Exemple

```
div {  
width : 300px;  
height : 100px;  
background-color : cyan;  
border :1px solid black;  
}  
div:hover{  
transform : translate(50px,100px);  
}
```

# Les transformations 2D

La méthode rotate()

- Exemple : faire pivoter l'élément <div> dans le sens des aiguilles d'une montre de 20 degrés :

## Exemple

```
div {  
width : 300px;  
height : 100px;  
background-color : cyan;  
border : 1px solid black;  
}  
div:hover{  
transform : rotate(20deg);  
}
```



# Les transformations 2D

La méthode scale()

- Exemple : augmenter la taille de l'élément <div> de deux fois sur sa largeur et trois fois sa hauteur :

## Exemple

```
div {  
width : 300px;  
height : 100px;  
background-color : cyan;  
border : 1px solid black;  
}  
div:hover{  
transform :scale(2,3);  
}
```

# Les transformations 2D

La méthode skew()

- Exemple : incliner l'élément <div> de 20 degrés le long de l'axe X et de 10 degrés le long de l'axe Y :

## Exemple

```
div {  
width : 300px;  
height : 100px;  
background-color : cyan;  
border : 1px solid black;  
}  
div:hover{  
transform :skew(20deg,10deg);  
}
```

# Les transformations 2D

La méthode rotateX()

- Permet de faire une rotation par rapport à l'axe X (on peut utiliser aussi rotateY et rotateZ).

## Exemple

```
div {  
width : 300px;  
height : 100px;  
background-color : cyan;  
border : 1px solid black;  
}  
div:hover{  
transform : rotateX(60deg);  
}
```

# Transitions CSS

Les transitions CSS permettent de modifier les valeurs des propriétés sur une durée donnée.

Exemple : un élément `<div>` rouge de 100 px \* 100 px subit un effet de transition pour la propriété width, d'une durée de 2 secondes :

## Exemple

```
div {width : 100px;  
height : 100px;  
background : red;  
transition : width 2s;/* ou bien height 2s*/  
}div:hover{  
height :300px;/* ou bien height 300px*/  
}
```

## La règle @keyframes

La spécification d'un style CSS à l'intérieur de la règle @keyframes permet à une animation de passer progressivement du style actuel au nouveau style.

### Exemple

```
@keyframes example {  
  from {background-color : red;}  
  to {background-color : yellow;}  
}  
/* The element to apply the animation to */  
div {  
  width : 100px;  
  height : 100px;  
  background-color : red;  
  animation-name : example; /*définit le nom de l'animation*/  
  animation-duration : 4s; /*définit la durée de l'animation*/  
}
```

# La règle @keyframes

## Exemple

```
@keyframes example {  
0% {background-color : red;}  
25% {background-color : yellow;}  
50% {background-color : blue;}  
100% {background-color : green;}  
}  
/* The element to apply the animation to */  
div {  
width : 100px;  
height : 100px;  
background-color : red;  
animation-name : example;  
animation-duration : 4s;  
}
```

## La règle @keyframes

### Exemple.html

```
div {  
width : 100px;  
height : 100px;  
position : relative;  
background-color : red;  
animation-name : example;  
animation-duration : 4s;  
}
```

### Exemple.css

```
@keyframes example {0%  
{background-color :red; left :0px;  
top :0px;}25% {background-color  
:yellow; left :200px; top :0px;}50%  
{background-color :blue; left :200px;  
top :200px;}75% {background-color  
:green; left :0px; top :200px;}100%  
{background-color :red; left :0px; top  
:0px;}}
```