

Licence Développement Web



Chapitre 2

CSS



D'après Achref EL AMRI

Introduction

3

- HTML permet de structurer le contenu d'un site Web (paragraphes, tableaux, listes, etc.).
- Mais, ce langage ne permet pas de mettre en page le contenu du site.
 - ▣ Modifier la couleur du texte
 - ▣ Centrer des paragraphes, images, etc.
 - ▣ Ajouter des espaces entre les éléments du site
 - ▣ Etc.
- Il faut avoir recours aux feuilles de style CSS.

Introduction

4

- CSS (**C**ascading **S**tyle **S**heets) est un langage complémentaire du HTML permettant de gérer la présentation du contenu d'un site Web.
 - ▣ La version actuelle est CSS 3
- Les navigateurs sont chargés de lire et d'interpréter le code CSS comme pour HTML.
- Les feuilles de style CSS permettent de réduire la complexité du code HTML, d'accélérer le chargement d'une page Web, d'éviter les redondances dans le code HTML, etc.

Plan

5

- Intégration du CSS
- Syntaxe CSS
 - ▣ Sélecteurs
 - ▣ Propriétés
 - Texte
 - Liens

Intégration du CSS

Intégration du CSS

7

- Les règles de style CSS peuvent être déclarées avec les trois méthodes suivantes dans:
 - ▣ Un fichier externe « .css » (**méthode recommandée**)
 - Balise **<link>**
 - ▣ L'entête d'un document HTML (dans la balise **<head>**)
 - Balise **<style>**
 - ▣ Les balises HTML (dans le corps du document HTML: balise **<body>**)
 - Attribut **style**

Intégration du CSS

8

Méthode 1 - Fichier externe

- C'est la méthode la plus utilisée et recommandée.
- Elle permet de réduire la complexité et la taille d'un fichier HTML.
- Le fichier de style « .css » est séparé du fichier de contenu « .html ».
- Il faut lier ces deux fichiers grâce à la balise **<link>** introduite dans la partie entête du fichier HTML (balise **<head>**).

Intégration du CSS

9

Méthode 1 - Fichier externe

- La balise **<link>** permet d'informer le navigateur d'utiliser un fichier externe.
- Elle comprend principalement trois attributs:
 - ▣ **rel:** (obligatoire) spécifier la relation entre les fichiers liés
 - ▣ **type:** (optionnel) spécifier le type du fichier externe
 - ▣ **href:** (obligatoire) spécifier le fichier externe
- Exemple:

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

Intégration du CSS

10

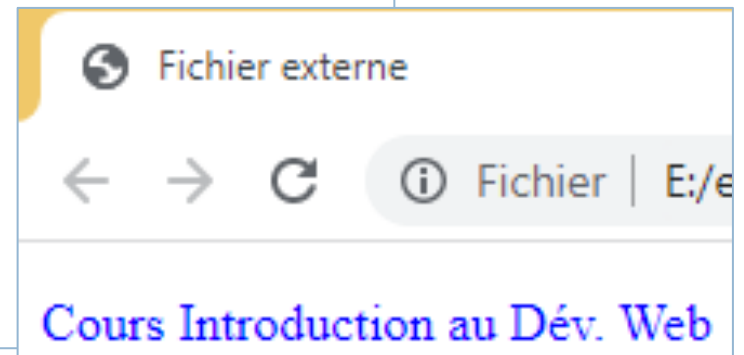
Méthode 1 - Fichier externe

- Exemple: Modifier la couleur d'un paragraphe en bleu.

```
<!DOCTYPE html>
<html>
<head>

  <title>Fichier externe</title>
  <link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
  <p>
    Cours Introduction au Dév. Web
  </p>
</body>
</html>
```

```
p {
  color: blue;
}
```

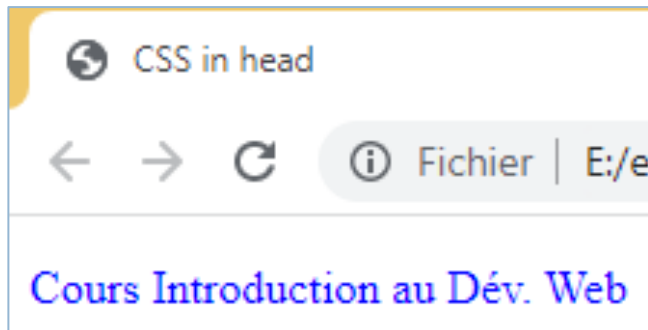


Intégration du CSS

11

Méthode 2 - Entête HTML (balise <head>)

- La balise **<style>** permet de définir des règles de style pour un document HTML.
 - ▣ L'attribut **type** permet de spécifier le type du code utilisé pour le style
 - Par défaut, text/css



```
<!DOCTYPE html>
<html>
<head>
  <title>CSS in head</title>
  <style type="text/css">
    p {
      color: blue;
    }
  </style>
</head>
<body>
  <p>
    Cours Introduction au Dév. Web
  </p>
</body>
</html>
```

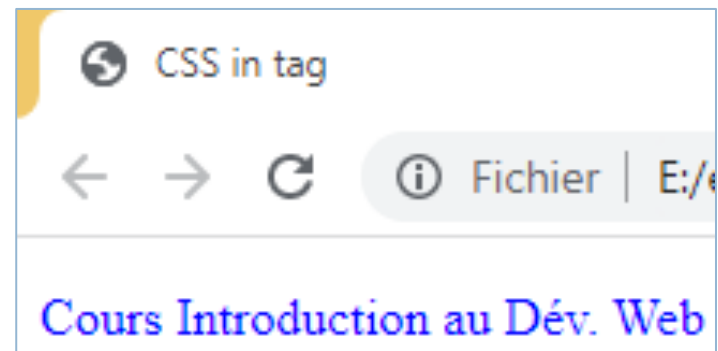
Intégration du CSS

12

Méthode 3 - Balises HTML (balise <body>)

- L'attribut **style** d'une balise quelconque permet de définir le style d'un élément HTML.
- C'est la méthode la moins utilisée et déconseillée.
 - ▣ Elle permet de charger le code d'un document HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS in tag</title>
</head>
<body>
  <p style="color: blue;">
    Cours Introduction au Dév. Web
  </p>
</body>
</html>
```



Intégration du CSS

13

Combinaison de styles

- Les règles de style peuvent être définies dans plusieurs endroits et seront **combinées** pour formater l'affichage d'un élément HTML.
- Exemple:

```
p {  
    color: ■ red;  
}
```

```
<!DOCTYPE html>  
<html>  
<head>  
  
    <title>Fichier externe</title>  
    <link rel="stylesheet" type="text/css" href="style.css"/>  
    <style>  
        p {  
            font-style: italic;  
        }  
    </style>  
</head>  
<body>  
    <p style="background-color: ■ black;">  
        Cours Introduction au Dév. Web  
    </p>  
</body>  
</html>
```

Intégration du CSS

14

Ordre de priorité de styles

- Une même règle de style peut être déclarée dans plusieurs endroits pour un élément HTML.
 - ▣ Le navigateur se chargera d'utiliser le style adéquat
- Il existe un ordre de priorité pour les styles:
 1. Style défini comme attribut d'une balise HTML
 2. Style défini dans l'entête du document HTML
 3. Style défini dans un fichier externe
 4. Style par défaut défini par le navigateur

Intégration du CSS

15

Ordre de priorité de styles

□ Exemple: Modifier la couleur d'un paragraphe

```
<!DOCTYPE html>
<html>
<head>

  <title>Fichier externe</title>
  <link rel="stylesheet" type="text/css" href="style.css"/>
  <style>
    p {
      color: cyan;
    }
  </style>
</head>
<body>
  <p style="color: green;">
    Cours Introduction au Dév. Web
  </p>
</body>
</html>
```

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

Cours Introduction au Dév. Web

Intégration du CSS

16

Héritage et conflits

- Le style défini pour un élément parent doit être appliqué à ses éléments enfants.
- Mais, si l'élément enfant définit son propre style, ce dernier sera priorisé et appliqué.
- Si plusieurs styles sont définis dans le même fichier pour un même élément, le dernier style lu sera appliqué.

Intégration du CSS

17

Héritage et conflits

□ Exemple:

```
<!DOCTYPE html>
<html>
<head>

  <title>Héritage en CSS</title>
  <link rel="stylesheet" href="style_heritage.css"/>
</head>
<body>
  <h1>Titre de niveau 1</h1>
  <p>Premier paragraphe avec du <strong>texte important</strong></p>
  <p>Deuxième paragraphe</p>
  <p style="color: red;">Troisième paragraphe</p>
</body>
</html>
```

Titre de niveau 1

Premier paragraphe avec du **texte important**

Deuxième paragraphe

Troisième paragraphe

```
body {
  color: blue;
}
p {
  color: purple;
}
strong {
  color: green;
}
```

18

Version à préconiser

Intégration du CSS

19

Méthode 1 - Fichier externe

- La balise **<link>** permet d'informer le navigateur d'utiliser un fichier externe.
- Elle comprend principalement trois attributs:
 - ▣ **rel:** (obligatoire) spécifier la relation entre les fichiers liés
 - ▣ **type:** (optionnel) spécifier le type du fichier externe
 - ▣ **href:** (obligatoire) spécifier le fichier externe
- Exemple:

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

Syntaxe CSS - Sélecteurs

Syntaxe CSS

21

Commentaires

- On peut écrire des commentaires dans un fichier CSS pour se repérer dans un fichier long.
 - ▣ Les commentaires seront invisibles sur la page Web
 - ▣ Il faut les mettre entre « /* » et « */ »
- Exemple:

```
/* exemple de fichier CSS */  
p {  
    color: ■red; /* modifier la couleur des paragraphes en rouge */  
}
```

Syntaxe CSS

22

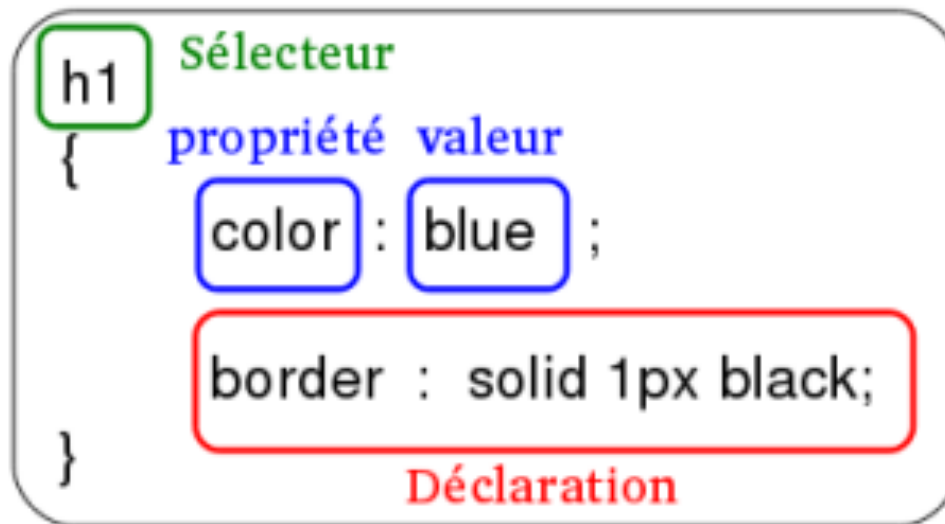
- Une feuille de style CSS comprend un ensemble de règles définissant la mise en page d'un document HTML.
- Une règle CSS est composée de trois éléments:
 - ▣ **Sélecteur:** sélectionner un élément HTML à mettre en page (paragraphe, titre, etc.)
 - ▣ **Propriété:** définir comment mettre en page l'élément sélectionné (modifier son texte, sa couleur, etc.)
 - ▣ **Valeur:** affecter une valeur à chaque propriété (couleur bleue pour le texte)

Syntaxe CSS

23

- Exemple: Modifier la couleur du texte et la bordure des titres de niveau 1.

Une règle CSS



Syntaxe CSS

24

Sélecteurs

- Il existe plusieurs types de sélecteurs:
 - ▣ **Universel**: appliquer le style à toutes les balises d'un document HTML (paragraphes, titres, tableaux, etc.)
 - ▣ **De balise (de type)**: appliquer le style à toutes les balises de même type (paragraphes)
 - ▣ **De classe**: appliquer le style à une seule balise spécifiée par sa classe (attribut **class**)
 - ▣ **D'identifiant**: appliquer le style à une seule balise spécifiée par son identifiant (attribut **id**)
 - ▣ Etc.

Syntaxe CSS

25

Sélecteur universel

- Toutes les balises d'un document HTML seront sélectionnées sans exception.
 - ▣ Ce sélecteur est identifié par « * »
- Exemple: Colorier en bleu tous les éléments HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemples chapitre 2</title>
  <link rel="stylesheet" href="style_chap2.css"/>
</head>
<body>
  <h1>Licence</h1>
  <p>Cours Introduction au Dév. Web</p>
  <p>Cours Introduction à l'IA</p>
</body>
</html>
```

```
* {
  color: blue;
}
```



Syntaxe CSS

26

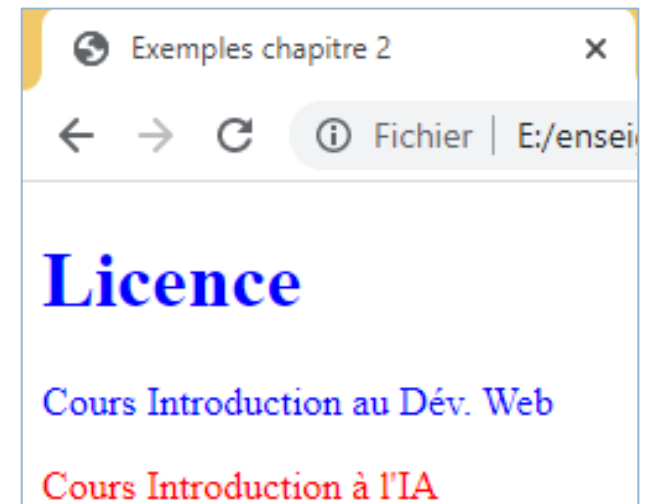
Sélecteur universel

- Si un autre style spécifique est déclaré dans le document HTML, il sera plus prioritaire par rapport au style universel.

- Exemple:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemples chapitre 2</title>
  <link rel="stylesheet" href="style_chap2.css"/>
</head>
<body>
  <h1>Licence</h1>
  <p>Cours Introduction au Dév. Web</p>
  <p style="color: red;">Cours Introduction à l'IA</p>
</body>
</html>
```

```
* {
  color: blue;
}
```



Syntaxe CSS

27

Sélecteur de balise (de type)

- Le nom d'une balise HTML joue le rôle d'un sélecteur (p, h1, h2, etc.).
 - ▣ Le style déclaré sera appliqué à tous les éléments du même type d'un document HTML

□ Exemple:

```
h1 {  
  color: blue;  
}  
p {  
  color: red;  
}
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Exemples chapitre 2</title>  
    <link rel="stylesheet" href="style_chap2.css">  
  </head>  
  <body>  
    <h1>Licence</h1>  
    <p>Cours Introduction au Dév. Web</p>  
    <p>Cours Introduction à l'IA</p>  
  </body>  
</html>
```



Syntaxe CSS

28

Sélecteur de balise (de type)

- Un style peut être appliqué à plusieurs sélecteurs séparés par des virgules.

- Exemple:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemples chapitre 2</title>
  <link rel="stylesheet" href="style_chap2.css"/>
</head>
<body>
  <h1>Licence</h1>
  <h2>INSA</h2>
  <p>Cours Introduction au Dév. Web</p>
  <p>Cours Introduction à l'IA</p>
</body>
</html>
```

```
h1,h2 {
  color: blue;
}
p {
  color: red;
}
```



Syntaxe CSS

29

Sélecteur de classe (attribut class)

- Avec les sélecteurs de balise, le style est appliqué à tous les éléments définis par la même balise.
- Mais, si on veut appliquer un style différent à un élément HTML.
 - ▣ Il faut utiliser l'attribut **class** de la balise HTML associée à l'élément
 - ▣ La valeur de l'attribut **class** doit être précédée par un point dans le sélecteur du fichier de style CSS

Syntaxe CSS

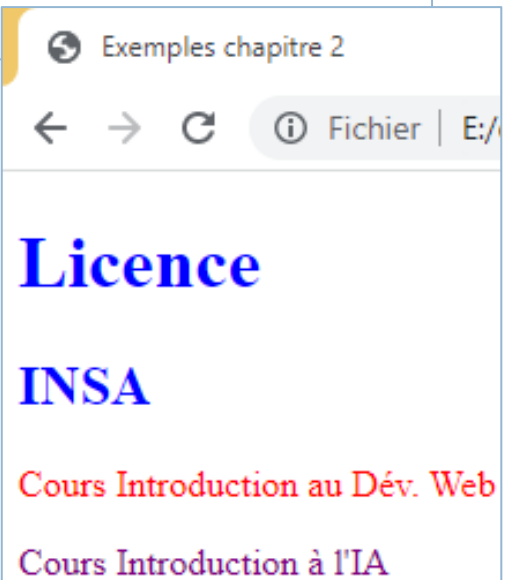
30

Sélecteur de classe (attribut class)

- Exemple: Modifier la couleur du texte du deuxième paragraphe en violet.

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemples chapitre 2</title>
  <link rel="stylesheet" href="style_chap2.css"/>
</head>
<body>
  <h1>Licence</h1>
  <h2>INSA</h2>
  <p>Cours Introduction au Dév. Web</p>
  <p class="p2">Cours Introduction à l'IA</p>
</body>
</html>
```

```
p {
  color: red;
}
.p2 {
  color: purple;
}
```



Syntaxe CSS

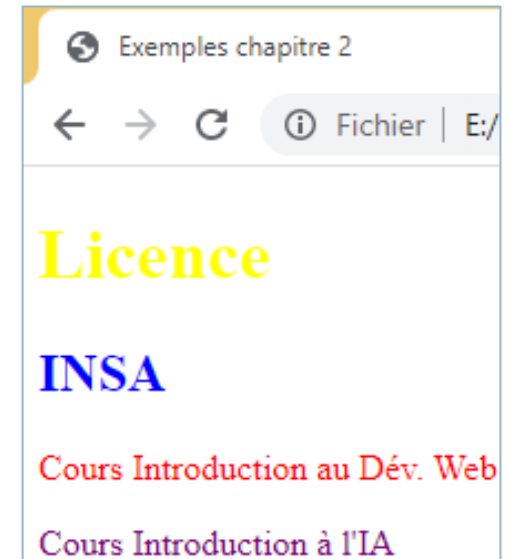
31

Sélecteur de classe (attribut class)

- Si la même classe est spécifiée à plusieurs éléments HTML, il faut précéder le sélecteur de classe par le nom de la balise associée en CSS.
- Exemple: Modifier la couleur d'un titre de niveau 1 en jaune.

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemples chapitre 2</title>
  <link rel="stylesheet" href="style_chap2.css"/>
</head>
<body>
  <h1 class="p2">Licence</h1>
  <h2>INSA</h2>
  <p>Cours Introduction au Dév. Web</p>
  <p class="p2">Cours Introduction à l'IA</p>
</body>
</html>
```

```
.p2 {
  color: purple;
}
h1.p2 {
  color: yellow;
}
```



Syntaxe CSS

32

Sélecteur d'identifiant (attribut id)

- ❑ L'attribut **id** est identique à l'attribut **class** sauf qu'il doit être unique dans un document HTML.
- ❑ Le style ne sera appliqué qu'à un seul élément HTML.
- ❑ La valeur affectée à l'attribut **id** dans la balise HTML doit être précédée par « # » dans le sélecteur du fichier de style CSS.

Syntaxe CSS

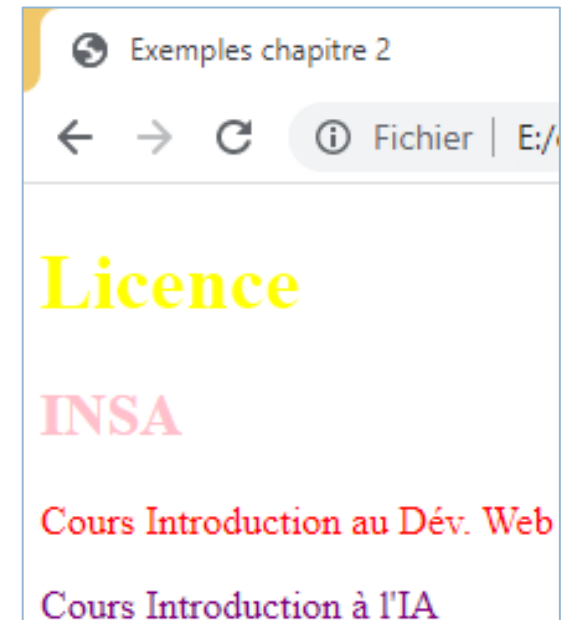
33

Sélecteur d'identifiant (attribut id)

- Exemple: Modifier la couleur d'un titre de niveau 2 en rose.

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemples chapitre 2</title>
  <link rel="stylesheet" href="style_chap2.css"/>
</head>
<body>
  <h1 class="p2">Licence</h1>
  <h2 id="insa">INSA</h2>
  <p>Cours Introduction au Dév. Web</p>
  <p class="p2">Cours Introduction à l'IA</p>
</body>
</html>
```

```
#insa {
  color: pink;
}
```



Syntaxe CSS

34

Sélecteur d'identifiant (attribut id)

- Si le même identifiant est spécifié à plusieurs éléments en HTML, le sélecteur doit être précédé par le nom de la balise associée en CSS.
- Exemple: Modifier la couleur du texte d'un paragraphe en vert.

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemples chapitre 2</title>
  <link rel="stylesheet" href="style_chap2.css"/>
</head>
<body>
  <h1 class="p2">Licence</h1>
  <h2 id="ins">INSA</h2>
  <p id="ins">Cours Introduction au Dév. Web</p>
  <p class="p2">Cours Introduction à l'IA</p>
</body>
</html>
```

```
#ins {
  color: pink;
}

p#ins {
  color: green;
}
```



Syntaxe CSS

35

Balises universelles `` et `<div>`

- On peut modifier la couleur d'un paragraphe mais non pas la couleur d'un mot de ce paragraphe.
 - ▣ **Solution:** Balise ``
- Afin de définir le même style pour plusieurs éléments HTML différents, il faut utiliser les attributs **class** et **id** pour chaque élément à part.
 - ▣ **Solution:** Balise `<div>`
- Les deux attributs **class** et **id** des balises `` et `<div>` permettent de définir le style.

Syntaxe CSS

36

Balises universelles `` et `<div>`

- La balise `` est de type *inline* (placée dans un paragraphe: balise `` comme exemple) permettant de sélectionner des mots dans un paragraphe.
- La balise `<div>` est de type *block* (balise `<p>` comme exemple) permettant de regrouper des éléments HTML et effectuer un saut de ligne avant et après le bloc.

Syntaxe CSS

37

Balises universelles et <div>

- Exemple: colorier le mot « Introduction » en cyan.

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemples chapitre 2</title>
  <link rel="stylesheet" href="style_chap2.css"/>
</head>
<body>
  <h1 class="p2">Licence</h1>
  <h2 id="ins">INSA</h2>
  <p id="ins">Cours <span id="intro">Introduction</span> au Dév. Web</p>
  <p class="p2">Cours Introduction à l'IA</p>
</body>
</html>
```

```
#ins {
  color: green;
}
#intro {
  color: cyan;
}
```

Exemples chapitre 2

← → ↻ ⓘ Fichier | E:/

Licence

INSA

Cours Introduction au Dév. Web

Cours Introduction à l'IA

Syntaxe CSS

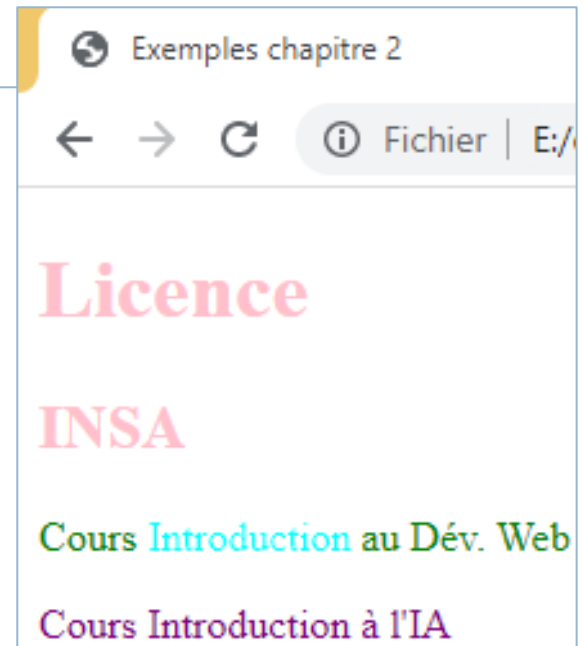
38

Balises universelles et <div>

- Exemple: colorier les deux titres de niveaux 1 et 2 en rose.

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemples chapitre 2</title>
  <link rel="stylesheet" href="style_chap2.css"/>
</head>
<body>
  <div id="group">
    <h1>Licence</h1>
    <h2>INSA</h2>
  </div>
  <p id="ins">Cours <span id="intro">Introduction</span> au D v. Web</p>
  <p class="p2">Cours Introduction   l'IA</p>
</body>
</html>
```

```
#group {
  color: pink;
}
```



Syntaxe CSS

39

Sélecteurs descendants

- On peut définir le style d'un élément contenu dans un autre d'une manière **directe** ou **indirecte**.
- Le sélecteur comprend la balise mère suivie de la balise fille séparées par un espace.
- Exemple: **** contenue dans **<p>**

```
<p>Ce texte est écrit en <em>bleu</em></p>  
<p>Ce texte est <b>écrit en <em>bleu</em></b></p>
```

```
p em {  
    color: ■blue;  
}
```

Ce texte est écrit en *bleu*
Ce texte est **écrit en** *bleu*


Syntaxe CSS

40

Sélecteurs enfants

- On peut définir le style d'un élément contenu dans un autre d'une manière **directe**.
- Le sélecteur comprend la balise mère suivie de la balise fille séparées par « > ».
- Exemple: **** contenue dans **<p>**

```
<p>Ce texte est écrit en <em>jaune</em></p>  
<p>Ce texte est <b>écrit en <em>jaune</em></b></p>
```

```
p > em {  
    color: yellow;  
}
```

Ce texte est écrit en *jaune*

Ce texte est **écrit en *jaune***

Syntaxe CSS

41

Sélecteurs adjacents

- On peut définir le style d'un élément suivant un autre d'une manière **immédiate**.
- Le sélecteur comprend les noms de balises associées aux éléments séparés par « + ».
- Exemple: **<h3>** suivie immédiatement de **<p>**

```
<h3>Notes</h3>
<p>DS, examen et PM</p>
<p>TP, DS et examen</p>
```

```
h3 + p {
  color: ■ red;
}
```

Notes

DS, examen et PM

TP, DS et examen

Syntaxe CSS

42

Autres sélecteurs

- Il existe bien évidemment d'autres sélecteurs à définir pour un document HTML:
 - ▣ Une balise qui comporte un attribut
 - Exemple: **a[title] {}**
 - ▣ Une balise qui comporte un attribut avec une valeur
 - Exemple: **a[title = "Université"] {}**
 - ▣ Une balise qui comporte un attribut avec un mot de la valeur
 - Exemple: **a[title* = "Université"] {}**

Syntaxe CSS - Propriétés

Syntaxe CSS- Texte

49

Propriétés

- En CSS, le texte peut être formaté avec plusieurs propriétés comme:
 - ▣ **font-size**: modifier la taille du texte
 - ▣ **font-style**: modifier le style du texte
 - ▣ **font-family**: modifier la fonte (police) du texte utilisée
 - ▣ **font-weight**: modifier le poids du texte
 - ▣ **text-decoration**: modifier la décoration du texte
 - ▣ etc.

Syntaxe CSS- Texte

50

Propriété font-size

- Elle permet de spécifier la taille du texte exprimée par une valeur:
 - ▣ **Fixe en pixels:** 16px par défaut
 - ▣ **Relative:** taille proportionnelle aux préférences des visiteurs (méthode recommandée)
 - xx-small, x-small, small, medium, large, x-large, xx-large
 - En **em**
 - 1em: taille normale
 - > 1em: agrandir la taille (exemple: 1.3em)
 - < 1em: réduire la taille (exemple: 0.8em)

Syntaxe CSS- Texte

51

Propriété font-size

□ Exemples:

```
<p class="p10">Texte en 10 px</p>
<p>Texte normal</p>
<p class="p20">Texte en 20 px</p>
<p class="psmall">Texte en small</p>
<p class="pmedium">Texte en medium</p>
<p class="plarge">Texte en large</p>
<p class="pem1">Texte en 1.5em</p>
<p class="pem2">Texte en 0.8em</p>
```

```
.p10 {
    font-size: 10px;
}
.p20 {
    font-size: 20px;
}
.psmall {
    font-size: small;
}
.pmedium {
    font-size: medium;
}
.plarge {
    font-size: large;
}
.pem1 {
    font-size: 1.5em;
}
.pem2 {
    font-size: 0.8em;
}
```

Texte en 10 px

Texte normal

Texte en 20 px

Texte en small

Texte en medium

Texte en large

Texte en 1.5em

Texte en 0.8em

Syntaxe CSS- Texte

52

Propriété font-style

- Elle permet de spécifier le style du texte et prend comme valeur:
 - ▣ **normal**: pour un texte normal (par défaut)
 - Annuler une écriture en italique ou oblique
 - ▣ **italic**: pour un texte écrit en italique
 - ▣ **oblique**: pour un texte écrit en oblique
 - Les lettres seront penchées

Syntaxe CSS- Texte

53

Propriété font-style

□ Exemples:

```
<p class="pnormal">Texte en normal</p>  
<p class="pitalic">Texte en italique</p>  
<p class="poblique">Texte en oblique</p>
```

```
.pnormal {  
    font-style: normal;  
}  
.pitalic {  
    font-style: italic;  
}  
.poblique {  
    font-style: oblique;  
}
```

Texte en normal

Texte en italique

Texte en oblique

Syntaxe CSS- Texte

54

Propriété font-family

- Elle permet de spécifier la police à utiliser.
- Il faut déclarer plusieurs fontes.
 - ▣ Si aucune fonte n'est installée chez les visiteurs du site, la fonte par défaut du navigateur sera utilisée
- La fonte par défaut des navigateurs est *serif* ou *sans-serif*.
 - ▣ Il faut la déclarer en dernière position comme valeur de font-family

Syntaxe CSS- Texte

55

Propriété font-family

- Il existe plusieurs fontes comme Arial, Georgia, Times New Roman, Tw Cen MT, Verdana, etc.

- Exemples:

```
<p class="parial">Texte en Arial</p>
<p class="pgeorgia">Texte en Georgia</p>
<p class="ptimes">Texte en Times New Roman</p>
<p class="pcen">Texte en TW Cen MT</p>
<p class="pverdana">Texte en Verdana</p>
```

Texte en Arial

Texte en Georgia

Texte en Times New Roman

Texte en TW Cen MT

Texte en Verdana

```
.parial {
    font-family: Arial;
}
.pgeorgia {
    font-family: Georgia;
}
.ptimes {
    font-family: 'Times New Roman';
}
.pcen {
    font-family: 'Tw Cen MT';
}
.pverdana {
    font-family: Verdana;
}
```

Syntaxe CSS- Texte

56

Propriété font-family

- Si plusieurs fontes sont déclarées, le navigateur essaie avec la première, passera à la seconde en cas d'échec de la première et ainsi de suite.
- Exemple:

```
<p class="pchoix">Texte avec fonte choisie par le navigateur</p>
```

```
.pchoix {  
    font-family: Arial, Helvetica, sans-serif;  
}
```

Texte avec fonte choisie par le navigateur

Syntaxe CSS- Texte

57

Propriété font-weight

- Elle permet d'écrire en gras du texte et prend comme valeur:
 - ▣ **normal**: écriture normale du texte (par défaut), vaut 400
 - ▣ **bold**: écriture en gras du texte, vaut 700
 - ▣ **lighter, bolder**: écriture en gras par rapport à l'élément parent
 - ▣ **100-900**: pour réduire/augmenter le niveau de graisse

Syntaxe CSS- Texte

58

Propriété font-weight

□ Exemples:

```
<p class="pgnormal">Texte en gras normal</p>
<p class="pgbold">Texte en bold</p>
<p class="pgbolder">Texte en bolder</p>
<p class="pglighter">Texte en lighter</p>
<p class="pg900">Texte en gras 900</p>
```

Texte en gras normal

Texte en bold

Texte en bolder

Texte en lighter

Texte en gras 900

```
.pgnormal {
    font-weight: normal;
}
.pgbold {
    font-weight: bold;
}
.pgbolder {
    font-weight: bolder;
}
.pglighter {
    font-weight: lighter;
}
.pg900 {
    font-weight: 900;
}
```

Syntaxe CSS- Texte

59

Propriété font

- C'est un raccourci vers les propriétés précédentes (font-size, font-family, font-weight, etc.)
- Il permet de regrouper toutes les propriétés en une seule déclaration selon l'ordre suivant:
 - ▣ **font-style font-weight font-size font-family**

- Exemple:

```
<p class="pfont">Texte en italique, gras, large et Verdana</p>
```

```
.pfont {  
  font: italic bold large Verdana;  
}
```

Texte en italique, gras, large et Verdana

Syntaxe CSS- Texte

60

Propriété text-decoration

- Elle permet de souligner, barrer, mettre une ligne au dessus d'un texte et prend comme valeur:
 - ▣ **none**: écriture normale par défaut
 - ▣ **underline**: souligner un texte
 - ▣ **overline**: mettre une ligne au dessus du texte
 - ▣ **line-through**: barrer un texte
 - ▣ **blink**: faire clignoter un texte
 - Non supportée par tous les navigateurs

Syntaxe CSS- Texte

61

Propriété text-decoration

□ Exemples:

```
<p class="decornone">Texte normal</p>
<p class="decorunder">Texte souligné</p>
<p class="decorover">Texte avec ligne au dessus</p>
<p class="decorbar">Texte barré</p>
<p class="decorblink">Texte clignotant</p>
```

Texte normal

Texte souligné

Texte avec ligne au dessus

~~Texte barré~~

Texte clignotant

```
.decornone {
    text-decoration: none;
}
.decorunder {
    text-decoration: underline;
}
.decorover {
    text-decoration: overline;
}
.decorbar {
    text-decoration: line-through;
}
.decorblink {
    text-decoration: blink;
    animation-name: blinker;
    animation-duration: 1s;
    animation-timing-function: linear;
    animation-iteration-count: infinite;
}
@keyframes blinker {
    0% {opacity: 1.0;}
    50% {opacity: 0.0;}
    100% {opacity: 1.0;}
}
```


Syntaxe CSS- Texte

62

Propriété text-align

- Elle permet d'aligner un texte (à gauche, à droite, centré ou justifié).
- Elle prend comme valeur:
 - ▣ **left**: alignement à gauche (par défaut)
 - ▣ **right**: alignement à droite
 - ▣ **center**: alignement centré
 - ▣ **justify**: alignement justifié

Syntaxe CSS- Texte

63

Propriété text-align

□ Exemples:

```
<p class="alignl">Introduction au Dév. Web <br> HTML, CSS et javascript</p>
<p class="alignc">Introduction au Dév. Web <br> HTML, CSS et javascript</p>
<p class="alignr">Introduction au Dév. Web <br> HTML, CSS et javascript</p>
<p class="alignj">Introduction au Dév. Web <br> HTML, CSS et javascript</p>
```

Introduction au Dév. Web
HTML, CSS et javascript

Introduction au Dév. Web
HTML, CSS et javascript

Introduction au Dév. Web
HTML, CSS et javascript

Introduction au Dév. Web
HTML, CSS et javascript

```
.alignl {
    text-align: left;
}
.alignc {
    text-align: center;
}
.alignr {
    text-align: right;
}
.alignj {
    text-align: justify;
}
```

Syntaxe CSS- Texte

64

Propriété color

- Elle permet de modifier la couleur d'un texte et prend comme valeur:
 - ▣ **nom de couleur:** au total 16 couleurs écrites en Anglais (white, silver, red, black, etc.)
 - ▣ **nombre en hexadécimal:** commence par « # » et comprend six caractères en hexadécimal (RGB)
 - ▣ **quantité RGB:** commence par « rgb » et comprend trois nombres en décimal




Syntaxe CSS- Texte

65

Propriété color

□ Exemples:

```
<h1 class="hcolor1">Titre en argent</h1>  
<h1 class="hcolor2">Titre en rouge</h1>  
<h1 class="hcolor3">Titre en corail</h1>
```

```
.hcolor1 {  
  color:  silver;  
}  
.hcolor2 {  
  color:  #FF0000;  
}  
.hcolor3 {  
  color:  rgb(231, 62, 1);  
}
```

Titre en argent

Titre en rouge

Titre en corail

Syntaxe CSS- Texte

66

Propriété background-color

- Elle permet de modifier la couleur de fond d'un texte et fonctionne comme la propriété **color**.

- Exemples:

```
.hcolor1 {  
  color: ■ silver;  
  background-color: ■ black;  
}  
.hcolor2 {  
  color: ■ #FF0000;  
  background-color: ■ #00FF00;  
}  
.hcolor3 {  
  color: ■ rgb(231, 62, 1);  
  background-color: ■ rgb(253, 233, 224);  
}
```

```
<h1 class="hcolor1">Titre en argent</h1>  
<h1 class="hcolor2">Titre en rouge</h1>  
<h1 class="hcolor3">Titre en corail</h1>
```

Titre en argent

Titre en rouge

Titre en corail

Syntaxe CSS- Texte

67

Propriété background-color

- On peut modifier la couleur de fond d'une page Web en agissant sur la balise **<body>**.
- Exemple:

```
<body class="bcolor">
```

```
.bcolor {  
    background-color: pink;  
}
```

Introduction au D v. Web
HTML, CSS et javascript

Titre en argent

Titre en rouge

Titre en corail

Syntaxe CSS- Liens

68

- Par défaut, un lien créé en HTML est souligné et coloré en bleu.
- Il existe des pseudo formats apportés par le CSS permettant de modifier l'apparence des éléments HTML d'une manière dynamique.
 - ▣ Au survol (le curseur pointe sur l'élément)
 - ▣ Au clic (on clique sur l'élément)
 - ▣ Au focus (on sélectionne l'élément)
 - ▣ À la visite d'un lien

Syntaxe CSS- Liens

69

- Les pseudo formats sont définis dans le sélecteur d'une règle CSS avec la syntaxe suivante:
 - ▣ **sélecteur:pseudo format**
- Parmi les pseudo formats pour les liens, on peut citer:
 - ▣ **:hover** (lien survolé)
 - ▣ **:active** (lien cliqué): maintien du clic ou clic bouton droit
 - ▣ **:focus** (lien sélectionné)
 - ▣ **:visited** (lien visité)

Syntaxe CSS- Liens

70

□ Exemple: Modifier le style d'un lien

```
<!DOCTYPE html>
<html>
<head>
  <title>Liens</title>
  <link rel="stylesheet" href="style_liens.css"/>
</head>
<body>
  <a href="https://www.google.com">Accéder à Google</a>
</body>
</html>
```

[Accéder à Google](https://www.google.com)

```
a {
  text-decoration: none;
  color: pink;
  font-style: italic;
}
```

Accéder à Google

Syntaxe CSS- Liens

71

□ Exemples:

```
a:visited {  
    color: ■ aqua;  
}  
a:active {  
    color: ■ yellow;  
}  
a:focus {  
    color: ■ gray;  
}  
a:hover {  
    text-decoration: underline;  
    color: □ white;  
    background-color: ■ black;  
}
```

Accéder à Google

Accéder à Google

Accéder à Google

Accéder à Google

72

The CSS Box Model

MARGES ET ESPACES INTÉRIEURS EN CSS

Gestion de l'espacement avec margin et padding



Le CSS Box Model (Modèle de Boîte en CSS)

74

- Le **CSS Box Model** est un concept fondamental en CSS qui décrit la structure d'affichage des éléments HTML sous forme de boîtes. Chaque élément est une boîte qui comprend plusieurs couches :
- 1. **Le contenu (content)** → Texte ou image à l'intérieur de l'élément.
- 2. **Le padding (padding)** → Espace autour du contenu, à l'intérieur de la bordure.
- 3. **La bordure (border)** → Entoure le padding et le contenu.
- 4. **La marge (margin)** → Espace extérieur entre l'élément et les autres éléments.

- | Margin (marge extérieure) |
- | ----- |
- | | Border (bordure) | |
- | | ----- | |
- | | | Padding (espace) | | |
- | | | ----- | | |
- | | | Content (texte) | | |
- | | | ----- | | |
- | | ----- | |
- | ----- |

Exemple

76

CSS

```
.box {  
  width: 200px;  
  height: 100px;  
  background-color: lightblue;  
  padding: 20px;  
  border: 5px solid blue;  
  margin: 30px;  
}
```

HTML

```
<div class="box">
```

Ceci est une boîte avec le modèle CSS Box Model

```
</div>
```

Démo

77

□ <https://codepen.io/rtomczak/pen/PwYrLOo>

Introduction

- En CSS, la gestion des espacements est essentielle pour la mise en page.
- **`margin`** définit l'espace extérieur entre un élément et les autres.
- **`padding`** définit l'espace intérieur entre le contenu et la bordure d'un élément.
- Ces propriétés permettent d'améliorer la lisibilité et l'organisation du contenu.

Margin vs Padding

- **`margin`** : crée de l'espace autour d'un élément, séparant les éléments entre eux.
- **`padding`** : crée de l'espace entre le contenu et la bordure de l'élément.
- Un élément peut avoir à la fois un **`margin`** et un **`padding`**.

Valeurs possibles

- Unités fixes : ``px``, ``em``, ``rem``, etc.
- Pourcentage ``%`` : basé sur la taille du conteneur parent.
- ``auto`` : permet au navigateur de calculer la marge automatiquement.
- Valeurs multiples : ``margin: 10px 20px 30px 40px;`` (haut, droite, bas, gauche).



Explication des Valeurs Acceptées

81

- **px (pixels)** : Unité fixe (ex : margin: 10px;)
- **% (pourcentage)** : Par rapport à l'élément parent (ex : padding: 5%;)
- **em** : Relative à la taille de la police (ex : margin: 2em;)
- **rem** : Relative à la taille de la police de l'élément racine
- **auto** : Laisse le navigateur gérer la valeur (ex : margin: auto; pour centrer)
- **inherit** : Hérite de la valeur du parent

Démo

82

- <https://codepen.io/rtomczak/pen/LEPKaRB>

HTML

```
1 <div class="container">
2   je suis dans le conteneur
3 <div class="box">je suis dans la boîte</div>
4 </div>
5
```

CSS

```
1 .container {
2   width: 300px;
3   background-color: lightblue;
4   padding: 20px; /* Espace intérieur */
5   margin: 3px auto; /* Espace extérieur et centré
   */
6 }
7 .box {
8   width: 100px;
9   height: 100px;
10  background-color: coral;
11  padding: 50px; /* Espace interne */
12  /*padding-right: 1px;*/
13  margin: 15px; /* Espace externe */
14 }
```

je suis dans le conteneur

je suis dans la
boîte



Autres Façons d'Écrire le Padding :

84

- `padding-right: 20px;` → **Ajoute uniquement à droite**
- `padding: 0 20px 0 0;` → **(haut, droite, bas, gauche)**
- `padding: 0px 20px;` → **(haut/bas, droite/gauche)**



Résumé Visuel :

85

Boîte CSS



Padding = Espace entre le contenu et la bordure



Margin = Espace autour de la boîte

87









Flexbox en CSS

Une Disposition Flexible pour les Éléments

Flexbox

88

- ❑  **Flexbox en CSS : Une Disposition Flexible pour les Éléments**
- ❑ **Flexbox** (Flexible Box Layout) est un modèle de disposition en CSS conçu pour organiser les éléments de manière efficace, que ce soit en **ligne** ou en **colonne**.
- ❑ Il est particulièrement utile pour **aligner, distribuer et redimensionner les éléments** dans un conteneur.

- ❑  **Pourquoi utiliser Flexbox ?**
- ❑  Facilite l'alignement des éléments (horizontaux et verticaux)
- ❑  Gère la répartition de l'espace entre les éléments
- ❑  Permet un ajustement automatique des tailles des éléments
- ❑  Évite les problèmes liés au float, inline-block et position



Les Concepts Clés de Flexbox :

89

❑ 1 Le Conteneur (Parent)

- ❑ On applique `display: flex;` sur un élément parent pour activer Flexbox :

```
.container {  
    display: flex; /* Active Flexbox */  
    flex-direction: row; /* Définit la direction principale */  
}
```

❑ 2 Les Éléments Flexibles (Enfants)

- ❑ Les éléments à l'intérieur du conteneur deviennent flexibles :

```
.item {  
    flex: 1; /* Permet aux éléments de prendre l'espace disponible */  
    padding: 10px;  
    border: 2px solid orange;  
}
```



Les Propriétés Importantes de Flexbox :

90



1. flex-direction → Définit l'axe principal


- row (*par défaut*) : Alignement horizontal (de gauche à droite)
- row-reverse : Alignement horizontal inversé (de droite à gauche)
- column : Alignement vertical (de haut en bas)
- column-reverse : Alignement vertical inversé (de bas en haut)

```
.container {  
  display: flex;  
  flex-direction: row; /* Les éléments s'affichent en ligne */  
}
```



Les Propriétés Importantes de Flexbox :

91


-  **2. justify-content → Alignement des éléments sur l'axe principal**
 - flex-start (*par défaut*) : Alignement à gauche
 - flex-end : Alignement à droite
 - center : Alignement au centre
 - space-between : Espacement maximal entre les éléments
 - space-around : Espacement autour des éléments

```
.container {  
  display: flex;  
  justify-content: center; /* Centre les éléments horizontalement */  
}
```



Les Propriétés Importantes de Flexbox :

92


- ❑  **3. align-items → Alignement des éléments sur l'axe secondaire**
 - stretch (*par défaut*) : Éléments étirés pour remplir le conteneur
 - flex-start : Alignement en haut
 - flex-end : Alignement en bas
 - center : Alignement centré verticalement

```
.container {  
  display: flex;  
  align-items: center; /* Aligne les éléments au centre verticalement */  
}
```



Les Propriétés Importantes de Flexbox :

93


- ❑  **4. flex-wrap → Gestion du retour à la ligne**
 - nowrap (*par défaut*) : Tous les éléments restent sur une seule ligne
 - wrap : Les éléments passent à la ligne si nécessaire
 - wrap-reverse : Même chose mais en inversé

```
.container {  
  display: flex;  
  flex-wrap: wrap; /* Permet aux éléments de passer à la ligne si nécessaire */  
}
```



Les Propriétés Importantes de Flexbox :

94

- ❑  **5. flex (pour les enfants) → Contrôle de la taille des éléments**
 - flex: 1; → L'élément occupe tout l'espace disponible
 - flex: 2; → L'élément prend 2 fois plus d'espace qu'un autre élément flex: 1;

```
.item {  
  flex: 1; /* Chaque élément prendra un espace égal */  
}
```


□ Conclusion

- **Flexbox est essentiel** pour créer des mises en page responsives et dynamiques.
- Il simplifie l'**alignement et la disposition** des éléments.
- Il permet d'éviter l'**utilisation de float et position** pour l'organisation des éléments.

Démo

96

- <https://codepen.io/rtomczak/pen/XJrLQKy>

HTML

```
1
2
3 <div class="container">
4   <div class="item">1</div>
5   <div class="item">2</div>
6   <div class="item">3</div>
7 </div>
8
9
```

CSS

```
1 /* Appliquer Flexbox au conteneur */
2 .container {
3   display: flex; /* Active Flexbox */
4   flex-direction: row; /*column Organise les éléments en ligne */
5   justify-content: space-around; /* Répartit les éléments uniformément */
6   align-items: center; /* Centre verticalement */
7   height: 200px; /* Hauteur du conteneur */
8   background-color: lightgray; /* Fond gris clair */
9   /*width: 300px;*/
10 }
11
12 /* Style des éléments enfants */
13 .item {
14   width: 50px; /* Largeur de chaque boîte */
15   height: 50px; /* Hauteur de chaque boîte */
16   background-color: coral; /* Couleur de fond */
17   text-align: center; /* Texte centré */
18   line-height: 50px; /* Aligne le texte verticalement */
19   font-weight: bold; /* Texte en gras */
20   color: white; /* Couleur du texte */
21   border-radius: 5px; /* Coins arrondis */
22 }
```

1

2

3

98



CSS Grid






Un Système de Grilles Puissant

- Le **CSS Grid Layout** est un modèle de disposition qui permet de créer des mises en page **bidimensionnelles** (lignes et colonnes).
- Contrairement à **Flexbox**, qui gère une seule dimension à la fois, **CSS Grid** est parfait pour structurer des interfaces complexes



Pourquoi utiliser CSS Grid ?

100

- ❑  **Pourquoi utiliser CSS Grid ?**
- ❑  Créer des mises en page **alignées et structurées**
- ❑  Gérer **lignes ET colonnes** simultanément
- ❑  Moins de code CSS pour des mises en page avancées
- ❑  Meilleur contrôle sur l'organisation des éléments



Les Concepts Clés de CSS Grid

101

1 Le Conteneur Grid (Parent)

- Pour activer Grid, on utilise `display: grid;` sur un élément parent.

```
.container {  
    display: grid; /* Active CSS Grid */  
    grid-template-columns: 100px 200px 100px; /* Définit 3 colonnes */  
    grid-template-rows: 50px 100px; /* Définit 2 lignes */  
}
```

2 Les Éléments Grid (Enfants)

- Les éléments à l'intérieur du `.container` sont automatiquement placés dans les cellules de la grille.

```
.item {  
    background-color: coral;  
    padding: 10px;  
    border: 2px solid black;  
}
```



Propriétés Importantes de CSS Grid

102

- **1. grid-template-columns & grid-template-rows** →
Définit la structure de la grille

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 2fr 1fr; /* Trois colonnes de largeurs proportionnelles */  
  grid-template-rows: 100px 150px; /* Deux lignes avec hauteurs fixes */  
}
```

-  **Explication :**
 - 1 fr signifie une fraction de l'espace disponible.
 - 2fr prendra **deux fois plus d'espace** que 1 fr.



Propriétés Importantes de CSS Grid

103

- **2. gap (ou grid-gap)** → Espacement entre les éléments

```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-template-rows: repeat(2, 100px);  
  gap: 20px; /* Ajoute un espace entre les colonnes et les lignes */  
}
```



Propriétés Importantes de CSS Grid

104

- 3. **grid-column & grid-row** → **Contrôle individuel des éléments**

```
.item1 {  
  grid-column: 1 / 3; /* Étend l'élément de la colonne 1 à 3 */  
  grid-row: 1 / 2; /* Place l'élément dans la première ligne */  
}
```



Propriétés Importantes de CSS Grid

105

- **4. justify-items & align-items** → Alignement dans chaque cellule

```
.container {  
  justify-items: center; /* Centre horizontalement tous les éléments */  
  align-items: center; /* Centre verticalement tous les éléments */  
}
```



Propriétés Importantes de CSS Grid

106

- **5. justify-content & align-content** → Alignement global de la grille

```
.container {  
  justify-content: space-between; /* Répartit l'espace horizontalement */  
  align-content: center; /* Centre la grille verticalement */  
}
```



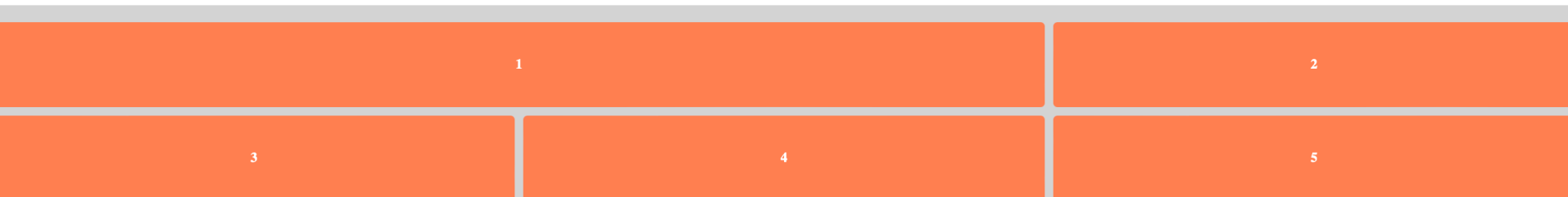
Propriétés Importantes de CSS Grid

107

- Démo
- <https://codepen.io/rtomczak/pen/yyBdrRG>

```
1 <div class="container">
2   <div class="item item1">1</div>
3   <div class="item">2</div>
4   <div class="item">3</div>
5   <div class="item">4</div>
6   <div class="item">5</div>
7 </div>
8
```

```
1 /* Activation de CSS Grid */
2 .container {
3   display: grid;
4   grid-template-columns: repeat(3, 1fr); /* Trois colonnes égales */
5   grid-template-rows: repeat(2, 100px); /* Deux lignes de 100px */
6   gap: 10px; /* Espacement entre les cellules */
7   background-color: lightgray;
8   padding: 20px;
9 }
10
11 /* Style des éléments de la grille */
12 .item {
13   background-color: coral;
14   text-align: center;
15   line-height: 100px;
16   font-weight: bold;
17   color: white;
18   border-radius: 5px;
19 }
20
21 /* Élément spécifique qui occupe deux colonnes */
22 .item1 {
23   grid-column: 1 / 3; /* S'étend sur deux colonnes */
24 }
25
```





Flexbox vs Grid :

109

CSS Grid	Flexbox
Gestion 2D (lignes & colonnes)	Gestion 1D (ligne OU colonne)
Idéal pour des layouts fixes & structurés	Idéal pour aligner et distribuer des éléments
Chaque élément peut être placé précisément	Les éléments s'adaptent selon l'espace disponible
Utilisation avec grid-template, gap, etc.	Utilisation avec justify-content, flex-wrap, etc.

💡 **Flexbox** = Parfait pour des mises en page simples et dynamiques.

💡 **CSS Grid** = Idéal pour des structures de pages complexes.

exemple

110

- Flexbox et grid CSS

113

Concepts Avancés en CSS





Pseudo-éléments avancés

114


□ Titre : Pseudo-éléments CSS

- ::before et ::after : ajout de contenu virtuel
- ::first-letter, ::first-line : stylisation du texte
- Exemple

```
p::first-letter {  
    font-size: 2em;  
    color: red;  
}
```

- ❑  **::before et ::after**
- ❑  Ces pseudo-éléments permettent d'**ajouter du contenu virtuel** avant ou après un élément.
 Ils sont souvent utilisés avec content pour insérer du texte, des icônes, ou des styles décoratifs.
- ❑  Ajoute une fusée avant chaque <p> sans modifier le HTML.

```
p::before {  
  content: "🚀";  
  font-size: 1.5em;  
  color: blue;  
}
```

- **::first-letter et ::first-line**
-  Permettent de **styler uniquement la première lettre ou la première ligne** d'un élément.

```
p::first-letter {  
    font-size: 2em;  
    color: red;  
}
```

- **Explication :**
 - **::first-letter** cible la **première lettre** de chaque **<p>**.
 - **font-size: 2em;** → Agrandit la première lettre.
 - **color: red;** → Change sa couleur en rouge.



Autres Exemples Utiles :

117

- ☐  ::first-line
- ☐  ::after

```
h1::after {  
  content: " 🔥";  
  color: orange;  
}
```

- ☐  Ajoute un emoji 🔥 après chaque <h1>.

Démo

118

```
/* Styliser la première lettre
de chaque paragraphe */
p::first-letter {
  font-size: 2.5em;
  color: red;
  font-weight: bold;
}

/* Appliquer un style spécial
à la première ligne du
paragraphe */
p::first-line {
  font-weight: bold;
  color: blue;
}
```

```
/* Ajouter du contenu avant chaque
<h1> */
h1::before {
  content: "🔥 ";
  color: orange;
  font-size: 1.2em;
}

/* Ajouter du contenu après chaque
<h1> */
h1::after {
  content: " ✨";
  color: gold;
  font-size: 1.2em;
}
```

Bienvenue sur CSS ✨

Ceci est un exemple illustrant l'utilisation des pseudo-éléments en CSS. Grâce aux pseudo-éléments, nous pouvons styliser certaines parties du texte sans modifier le HTML.

Seconde ligne

Démo

120

- <https://codepen.io/rtomczak/pen/vEBqwYZ>

121



Propriétés de Positionnement en CSS

- **Positionnement et Affichage**
- **Titre : Propriétés de positionnement**
 - position: static | relative | absolute | fixed | sticky
 - Différences entre inline, block, inline-block
 - z-index et contexte d'empilement

124



Propriétés de Positionnement en CSS

- **Clipping et Masques**
- **Titre : Effets visuels avec clip-path et mask-image**
 - clip-path: circle(); pour créer des formes
 - mask-image pour des effets de transparence
 - Exemple :

126



Propriétés de Positionnement en CSS

- **Variables CSS**
- **Titre : Optimisation avec les variables CSS**
 - `--ma-couleur: #3498db;`
 - Utilisation avec `var(--ma-couleur)`
 - Exemple :

128



Propriétés de Positionnement en CSS

- **Responsive Design**
- **Titre : Techniques pour un site adaptatif**
 - min-width, max-width, vh, vw
 - clamp(), calc(), min(), max()
 - Exemple :

130



Propriétés de Positionnement en CSS

- **Mode sombre et accessibilité**
- **Titre : Mode sombre et préférences utilisateur**
 - @media (prefers-color-scheme: dark)
 - :focus-visible pour l'accessibilité
 - Exemple :

132



Propriétés de Positionnement en CSS

- **Animations et Transitions**
- **Titre : Dynamiser le site avec CSS**
 - @keyframes et animation
 - transition et cubic-bezier()
 - Exemple



@keyframes et animation

134

- L'animation CSS utilise @keyframes pour définir une série d'étapes intermédiaires entre l'état initial et final d'un élément. Ensuite, la propriété animation applique cette animation à un élément.
- **Exemple d'animation avec @keyframes**

```

@keyframes bounce {
  0% {
    transform: translateY(0);
  }
  50% {
    transform: translateY(-20px);
  }
  100% {
    transform: translateY(0);
  }
}

.animated-box {
  width: 100px;
  height: 100px;
  background-color: coral;
  animation: bounce 1s infinite;
}

```

💡 Explication :

- @keyframes bounce définit le mouvement vertical de l'élément.
- L'élément .animated-box applique cette animation avec une durée de 1 seconde et une répétition infinie.

Démo

136

- <https://codepen.io/rtomczak/pen/MYgMNGz>
- <https://codepen.io/rtomczak/pen/MYgMNGz>

Pour finir...

138

- ❑ Le CSS permet de mettre en forme les éléments d'une page Web insérés avec HTML.
- ❑ Une règle d'une feuille de style CSS comprend un sélecteur, propriétés et valeurs.
- ❑ Il existe plusieurs catégories de sélecteurs: universel, de balise, d'identifiant, de classe, etc.
- ❑ Il existe des pseudo formats apportant un aspect dynamique au langage CSS.

FIN