COURS WEB 10

HTML / CSS

Cours N°10

FERRANDEZ BENJAMIN

Récapitulatif

Flexbox

F L E X B O X

C O M M E N T Ç A F O N C T I O N N E ?

- On active le flexbox dans notre container :

```css
.container {
    display: flex;
}
```

FLEXBOX PROPOSE PLUSIEURS PROPRIÉTÉS POUR POSITIONNER LES ÉLÉMENTS :

- `justify-content` : contrôle l'alignement horizontal.

  Exemples : `flex-start`, `center`, `space-between`, `space-around`.

- `align-items` : contrôle l'alignement vertical.

  Exemples : `flex-start`, `center`, `stretch`, `baseline`.

- `flex-direction` : définit la direction des éléments.

  Exemples : `row` (par défaut), `row-reverse`, `column`, `column-reverse`.

Mes projets

Nom projet numéro 1

Le projet numéro 1 dont je suis super fière. cliquez sur l'image pour acceder à Google

Nom projet numéro 2

Le projet numéro 2 dont je suis super fière. cliquez sur l'image pour acceder à Youtube

Nom projet numéro 3

Le projet numéro 3 dont je suis super fière. cliquez sur l'image pour acceder à Wikipédia

CSS PARENT :

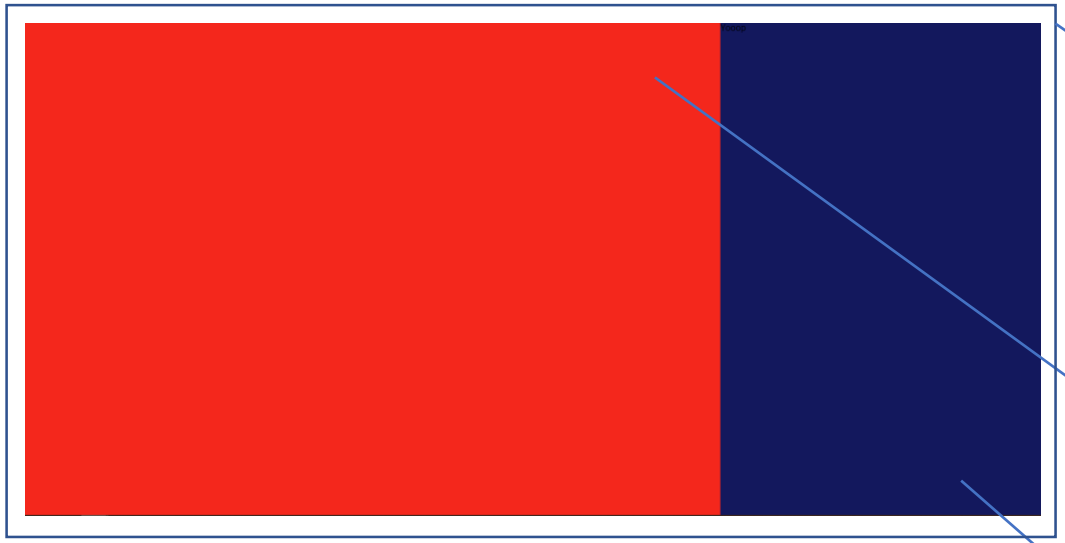
```css
body
{
    background-color: rgb(255, 255, 255);
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    min-height: 100vh;
    width: 100%;
    gap: 25px;
}
```

HTML PARENT + ENFANTS :

```html
<body>
    <div class="enfantDeBody" id="blocContainer">
        <div class="enfantDeblocContainer" id="container-card">
            <div class="enfantDeContainer-Card">
                Ici du texte par exemple
            </div>
            <div class="enfantDeContainer-Card">
                Ici du texte par exemple
            </div>
            <div class="enfantDeContainer-Card">
                Ici du texte par exemple
            </div>
        </div>
    </div>
</body>
```

Propriété *FLEX* et gestion des tailles

EXEMPLE HTML | CSS

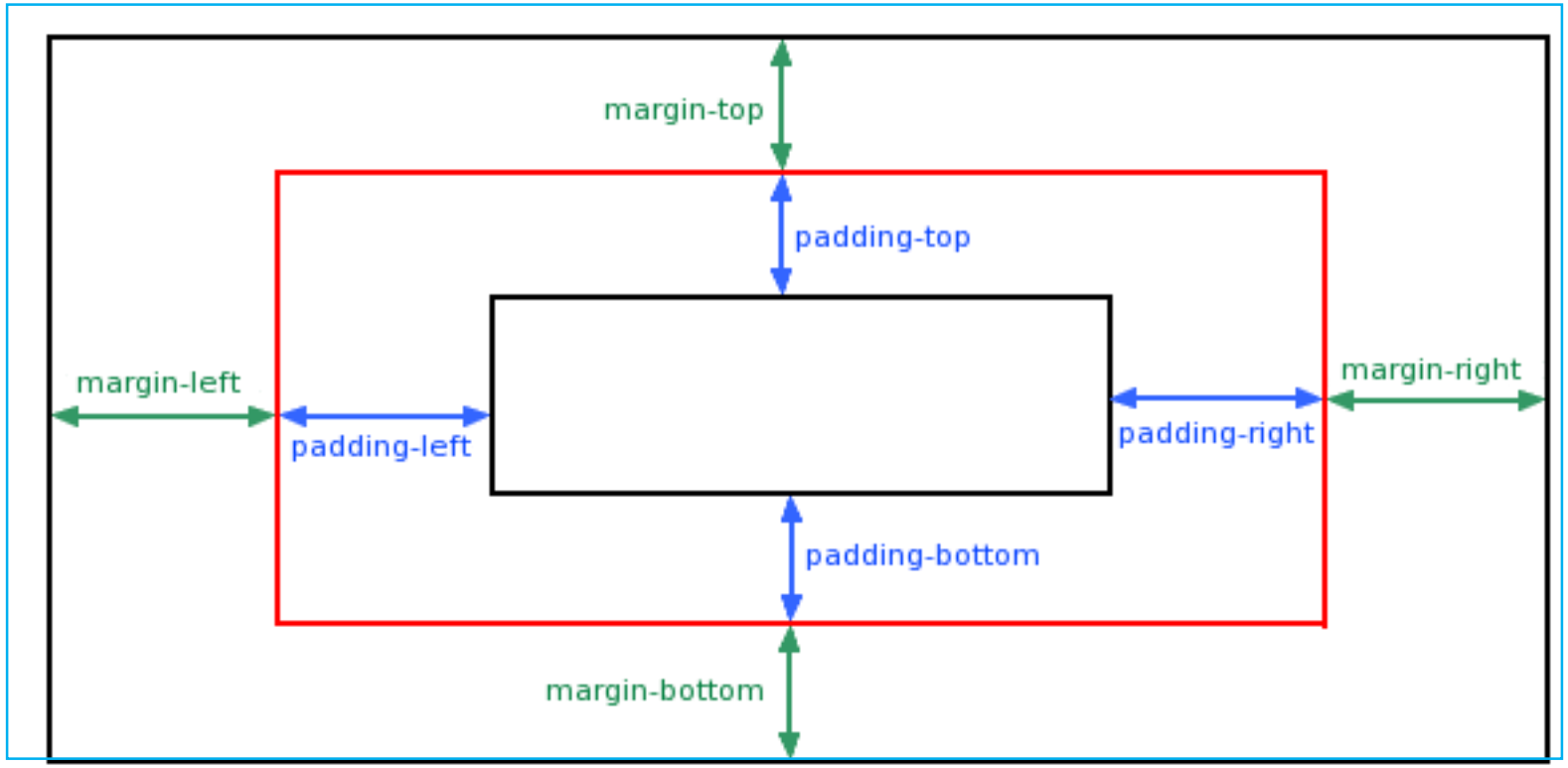


```css
.container-hobbies
{
    display: flex;
    justify-content: center;
    align-items: center;
    width: 100%;
    height: 100vh;
}

.first-hobbies {
    background-color: red;
    flex: 7; /* Obtient 7 part de l'espace disponible */
    height: 100%;
}

.second-hobbies {
    background-color: rgb(10, 25, 94);
    flex: 3; /* Obtient 3 part de l'espace disponible */
    height: 100%;
}
```

Margin/Padding

Les margin définissent les marges **extérieures**

Les padding définissent les marges **intérieures**



Positions

5 TYPES DE POSITIONS

Type de position	Description
`static`	- Position par défaut, suit l'ordre normal du document sans possibilité de déplacement avec `top`, `right`, `bottom`, `left`.
`relative`	- Positionné par rapport à sa position d'origine, peut être déplacé légèrement sans affecter les autres éléments du flux.
`absolute`	- Positionné par rapport au conteneur positionné le plus proche, retiré du flux normal, peut se superposer aux autres éléments.
`fixed`	- Positionné par rapport à la fenêtre, reste fixe même lorsque l'utilisateur fait défiler la page. Retiré du flux.
`sticky`	- D'abord `relative`, puis devient fixe lorsqu'il atteint une position donnée dans la fenêtre. Idéal pour des barres de navigation "collantes".

Unité de Mesures

Unité de mesure en CSS

- em : Unité de mesure de référence, surtout utilisé pour le texte, 1em = taille normale du texte

- px : unité de mesure en pixel, le plus commun

- % : Très utile pour faire des tailles relatives à la taille de l'écran de la personne. (Responsive etc..)

LES PX

Définition :

- Unité absolue qui représente un point fixe sur l’écran

Caractéristiques :

- Toujours de la même taille, quelle que soit la taille de l’écran ou du conteneur parent.
- Idéal pour un contrôle précis et fixe.

Utilisation typique :

- Boutons, bordures ou éléments ou la précision absolue est essentielle

E X E M P L E

```css
p {
  font-size: 16px;
}
```

Les em

Définition :

- Unité relative basée sur la taille de la police du parent

Caractéristiques :

- • Si la taille de la police du parent change, la valeur en em change également
- • 1 em correspond à la taille de la police du parent
- • Si un élément enfant utilise 2em, cela correspondra au double de la taille de la police parent.

Utilisation typique :

- Créer des mises en page adaptatives ou une hiérarchie de texte
- Souvent employé pour définir des marges, paddings ou intelignes relatifs

E X E M P L E

```
body {
  font-size: 16px; /* Base */
}
h1 {
  font-size: 2em; /* 2 * 16px = 32px */
}
```

Les %

Définition :

- Unité relative, souvent basée sur une dimension parent spécifique (largeur, hauteur etc)

Caractéristiques :

- • Relatif à la taille du conteneur parent (et non seulement à la police comme em)
- • Flexible et particulièrement utilise pour le design réactif

Utilisation typique :

- Largeur et hauteur des éléments (ex: images ou divs) pour adapter le contenue à l’écran

E X E M P L E

```
div {
  width: 50%; /* 50% de la largeur du conteneur parent */
}
```

Les REM

Définition :

- Unité relative basée sur la taille de la police racine du document (définie sur l'élément <html>).

Point à noter → Différence avec em :

- • Tandis que em est relatif à la taille de la police de son parent, rem est toujours relatif à la taille de la police définie sur l'élément <html>
- • Cela garantit une plus grande cohérence et facilite le contrôle global des proportions

Caractéristiques :

- • Toujours relative à la taille de police de l'élément racine (<html>), indépendamment des éléments parents.
- •Garantie de cohérence : les valeurs ne dépendent pas du contexte local (contrairement à EM), cequi permet une structure plus prévisible.
- • Facile à ajuster : en modifiant la taille de police de l'élément <html>, vous pouvez adapter tout le design

EXEMPLE

```css
html {
  font-size: 16px; /* Taille de base */
}

body {
  font-size: 1rem; /* Toujours 16px, car basé sur l'élément racine */
}

h1 {
  font-size: 2rem; /* 2 * 16px = 32px */
}
```

Résumé

Unité	Nature	Relatif à...	Avantages
`px`	Absolue	Taille fixe (pixels)	Contrôle précis.
`em`	Relative	Taille de la police parent	Hiérarchies locales de texte.
`rem`	Relative	Taille de la police racine	Cohérence globale et design réactif.
`%`	Relative	Dimensions du conteneur parent	Design réactif et flexible.

Pour un web design moderne et adaptable, il est courant de privilégier des unités relatives (em, %, rem) afin de faciliter l'adaptabilité et l'accessibilité des sites sur différents appareils

Class / Identifiant

Class

Puis, utilisation de la classe dans mon HTML :

```html
<div class="search-box">

</div>
```

```css
.search-box {
  height: 300px;
  width: 300px;
}
.search-box_light {
  background-color: #DEF;
  color: #777;
}
.search-box__btn {
  padding: 4px;
}
.search-box__btn_max_visible {
  font-weight: bold;
}
```

Identifiant

Puis, utilisation du CSS grâce à son identifiant :

```html
<div class="search-box" id ="my_great_id">

</div>
```

```css
.search-box {
  height: 300px;
  width: 300px;
}
.search-box_light {
  background-color: #DEF;
  color: #777;
}
.search-box__btn {
  padding: 4px;
}
.search-box__btn_max_visible {
  font-weight: bold;
}
```

```css
#my_great_id
{
  background-color: red;
}
```