

# Introduction à <div> et au CSS

## <div> en HTML

### Définition

La balise <div> (*division*) est un **conteneur générique** en HTML. Elle permet de **regrouper plusieurs éléments** pour structurer une page.

- Par défaut, elle **n'a aucun style visuel** : c'est une boîte invisible sur laquelle on pourra appliquer du CSS (couleurs, marges, bordures, etc.).
- C'est un **élément block** : il s'affiche sur une **nouvelle ligne** et occupe **toute la largeur** disponible.

**Bonne pratique** : Utiliser <div> uniquement si **aucune balise sémantique** ne convient (<section>, <article>, <header>, etc.). Cela rendra le code plus clair et accessible.

### Exemple

```
<div>
  <h2>Section Présentation</h2>
  <p>Ceci est un paragraphe regroupé dans une div.</p>
</div>
```

Sans CSS, rien ne se voit. Avec du CSS :

```
div {
  background-color: lightblue;
  border: 2px solid navy;
  padding: 20px;
  margin: 10px 0;
}
```

Résultat : le titre et le paragraphe sont regroupés dans une boîte bleue.

## Utilisation de <div>

- Structurer le contenu** : créer des zones logiques (ex : menu, contenu, pied de page).
- Appliquer un style** : avec `class` ou `id`.

```
<div class="menu"> ... </div>
<div id="contenu"> ... </div>
```

- Créer des mises en page** : colonnes, grilles, zones responsives, etc.

## CSS (Cascading Style Sheets : feuilles de style en cascade)

### Objectif du CSS

- Séparer le fond (HTML) de la forme (CSS)**.
- Appliquer un style uniforme : couleurs, marges, bordures, polices, effets...

### Couleurs

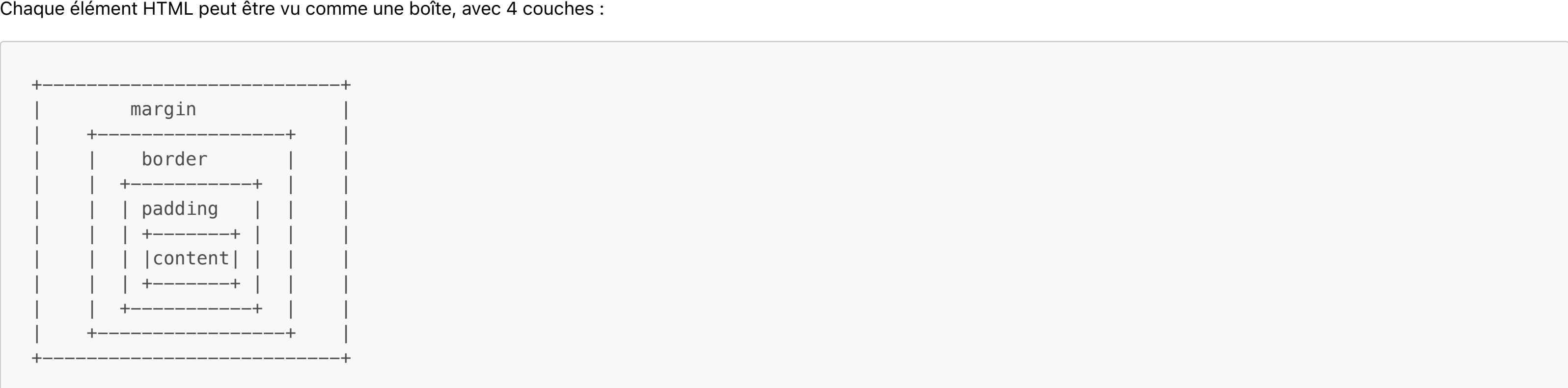
On peut définir une couleur en CSS avec :

- Nom prédéfini** : `red`, `blue`, `lightgray`...
- Code hexadécimal** : `#ff0000` (rouge)
- Code RGB** : `rgb(255, 0, 0)`
- Code RGBA** : `rgba(0, 0, 255, 0.5)` (bleu transparent)

```
p {
  color: #333;
  background-color: rgb(200, 200, 255);
}
```

### Le modèle de boîte CSS (Box Model)

Chaque élément HTML peut être vu comme une boîte, avec 4 couches :



**Content → Padding → Border → Margin**

### Propriétés associées :

- `content` : le contenu lui-même.
- `padding` : espace **intérieur** entre le contenu et la bordure.
- `border` : bordure visible autour de l'élément.
- `margin` : espace **extérieur** entre l'élément et les autres.

```
p {
  padding: 10px;
  border: 1px solid black;
  margin: 20px;
}
```

**Astuce développeur** : Vous pouvez visualiser le box model avec les outils du navigateur ([clic droit > Inspecter](#)).

### Exemple : le modèle de boîte en pratique

#### HTML

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Modèle de boîte CSS</title>
  <link rel="stylesheet" href="box-model.css">
</head>
<body>
  <h1>Exemple : Modèle de boîte</h1>

  <div class="box">Contenu</div>
  <p>Observe les effets du padding, de la bordure et des marges.</p>
</body>
</html>
```

#### CSS

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  text-align: center;
}

.box {
  background-color: lightblue;
  padding: 30px;
  border: 10px solid black;
  margin: 40px auto;
  width: 200px;
  font-weight: bold;
  line-height: 1.5;
}
```

### TP : Manipuler le box model

#### Objectifs

- Comprendre visuellement `padding`, `margin`, `border`.
- Observer les effets de chaque propriété.

#### Étapes

- Créer une page avec 3 boîtes :

```
<div class="box box1">Boîte 1</div>
<div class="box box2">Boîte 2</div>
<div class="box box3">Boîte 3</div>
```

- CSS :

```
.box {
  width: 200px;
  height: 100px;
  margin: 20px auto;
  text-align: center;
  line-height: 100px;
  font-weight: bold;
  background-color: lightblue;
}

.box1 {
  margin: 80px auto;
  border: 2px solid black;
}

.box2 {
  padding: 40px;
  border: 2px solid green;
  height: auto;
}

.box3 {
  border: 15px solid red;
  border-radius: 10px;
}
```

#### Exercices

- Changer les couleurs (`background-color`, `border`).
- Tester `margin: 80px 0 20px 50px`;
- Comparer une boîte avec `padding: 0` ; et `padding: 50px`;
- Essayer des bordures `dashed`, `dotted`, `double`.
- Bonus : ajouter une ombre avec `box-shadow`.

## TP CSS complet (étapes)

### Étape 0 : Préparation

- Créer un dossier `tp_css`
- Créer un fichier `index.html` et `style.css`
- Lier le CSS dans le HTML :

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

### Étape 1 : Style global

```
body {
  background-color: lightgray;
  font-family: Arial, sans-serif;
}
h1 {
  color: darkblue;
}
```

### Questions

- Que se passe-t-il si on remplace `lightgray` par `yellow` ?
- Quelle est la différence entre `body` et `h1` ici ?

### Étape 2 : Lisibilité du texte

```
p {
  line-height: 1.6;
  text-align: justify;
}
h2 {
  border-bottom: 2px solid #ccc;
  padding-bottom: 5px;
}
```

### Étape 3 : Liens interactifs

```
a {
  color: darkred;
  text-decoration: none;
}
a:hover {
  color: orange;
  text-decoration: underline;
}
```

### Étape 4 : Images

```
img {
  display: block;
  margin: 10px auto;
  border: 3px solid #ccc;
  border-radius: 8px;
}
```

### Étape 5 : Listes

```
ul {
  list-style-type: square;
  background-color: #eef;
  padding: 10px;
  border-radius: 5px;
}
ol {
  margin-left: 40px;
}
```

### Étape finale : Personnalisation

- Changer la couleur du titre.
- Utiliser un fond dégradé :

```
body {
  background: linear-gradient(to right, #a8bdea, #fed6e3);
}
```

- Ajouter `:active` aux liens :

```
a:active {
  color: purple;
}
```