

# Formation Vue.js

## Chapitre 1

### HTML et CSS

Ibrahim ALAME

2025

#### Objectifs

Ce chapitre couvre les fondamentaux du développement web :

- Comprendre la structure d'une page HTML5
- Maîtriser les balises HTML essentielles
- Appliquer des styles CSS aux éléments
- Positionner et mettre en forme des éléments
- Créer des formulaires et des tableaux
- Intégrer des médias (images, audio, vidéo)

**Prérequis :** Aucun - Ce chapitre est destiné aux débutants.

**Durée estimée :** 8 à 10 heures

# Contents

<b>1 Introduction</b>	<b>3</b>
1.1 Structure de base d'une page HTML5 . . . . .	3
1.2 Afficher un document HTML5 dans le navigateur . . . . .	4
1.2.1 Méthode simple : double-clic . . . . .	4
1.2.2 Méthode recommandée : serveur local . . . . .	4
1.3 Où écrire le CSS ? . . . . .	4
1.3.1 CSS en ligne (inline) . . . . .	4
1.3.2 CSS interne (embedded) . . . . .	5
1.3.3 CSS externe (recommandé) . . . . .	5
1.4 Validation et compatibilité entre navigateurs . . . . .	5
1.4.1 Valider son code . . . . .	5
1.4.2 Compatibilité navigateurs . . . . .	6
1.5 Bonnes pratiques, règles et commentaires . . . . .	6
1.5.1 Commentaires HTML . . . . .	6
1.5.2 Commentaires CSS . . . . .	6
1.5.3 Bonnes pratiques . . . . .	6
<b>2 Les bases du HTML</b>	<b>8</b>
2.1 Éléments, balises et attributs . . . . .	8
2.1.1 Balises auto-fermantes . . . . .	8
2.2 Les éléments de texte essentiels . . . . .	8
2.2.1 Titres (Headings) . . . . .	8
2.2.2 Paragraphes et sauts de ligne . . . . .	9
2.2.3 Mise en forme du texte . . . . .	9
2.3 Les listes . . . . .	9
2.3.1 Liste non ordonnée (à puces) . . . . .	9
2.3.2 Liste ordonnée (numérotée) . . . . .	10
2.3.3 Liste de définitions . . . . .	10
2.3.4 Listes imbriquées . . . . .	10
2.4 Les liens hypertextes . . . . .	11
2.4.1 Liens externes . . . . .	11
2.4.2 Liens internes . . . . .	11
2.4.3 Autres types de liens . . . . .	11
<b>3 Les bases du CSS</b>	<b>12</b>
3.1 Sélecteurs, propriétés et valeurs . . . . .	12
3.1.1 Sélecteurs de base . . . . .	12
3.2 Les éléments div et span . . . . .	13
3.2.1 Différence block vs inline . . . . .	13
3.3 Les sélecteurs avancés . . . . .	14
3.3.1 Combinateurs . . . . .	14
3.3.2 Sélecteurs d'attributs . . . . .	14
3.3.3 Pseudo-classes . . . . .	14
3.3.4 Pseudo-éléments . . . . .	15
3.4 La cascade et l'héritage . . . . .	15
3.4.1 Spécificité . . . . .	15

3.4.2 Héritage . . . . .	16
<b>4 Formater du texte et positionner des éléments</b>	<b>17</b>
4.1 Les propriétés de police (font-*) . . . . .	17
4.1.1 Couleur et opacité . . . . .	17
4.2 Les propriétés de texte (text-*) . . . . .	18
4.3 Le modèle des boîtes (Box Model) . . . . .	18
4.3.1 Dimensions . . . . .	18
4.3.2 Bordures . . . . .	19
4.3.3 Marges intérieures (padding) . . . . .	20
4.3.4 Marges extérieures (margin) . . . . .	20
4.3.5 Ombres des boîtes . . . . .	21
4.4 La propriété display . . . . .	21
4.4.1 Flexbox (bases) . . . . .	21
4.5 La propriété position . . . . .	22
4.5.1 Relative . . . . .	22
4.5.2 Absolute . . . . .	23
4.5.3 Fixed . . . . .	23
4.5.4 Sticky . . . . .	23
4.6 Z-index et empilement . . . . .	23
<b>5 Fonctionnalités avancées</b>	<b>25</b>
5.1 Gestion du background . . . . .	25
5.1.1 Couleur de fond . . . . .	25
5.1.2 Image de fond . . . . .	25
5.1.3 Images multiples et dégradés . . . . .	25
5.2 Intégrer des images, de l'audio et de la vidéo . . . . .	26
5.2.1 Images . . . . .	26
5.2.2 Audio . . . . .	26
5.2.3 Vidéo . . . . .	27
5.3 Les tableaux . . . . .	27
5.3.1 Structure de base . . . . .	27
5.3.2 Fusion de cellules . . . . .	28
5.3.3 Mise en forme CSS . . . . .	28
5.4 Les formulaires . . . . .	28
5.4.1 Structure de base . . . . .	28
5.4.2 Types de champs input . . . . .	29
5.4.3 Zone de texte multi-lignes . . . . .	29
5.4.4 Cases à cocher et boutons radio . . . . .	29
5.4.5 Listes déroulantes . . . . .	30
5.4.6 Attributs de validation . . . . .	30
5.4.7 Mise en forme des formulaires . . . . .	31
<b>6 Résumé et points clés</b>	<b>33</b>

## 1 Introduction

Le développement web repose sur trois technologies fondamentales qui travaillent ensemble pour créer des pages web interactives :

- **HTML** (HyperText Markup Language) : définit la *structure* et le *contenu* de la page
- **CSS** (Cascading Style Sheets) : gère la *présentation* et le *style* visuel
- **JavaScript** : ajoute l'*interactivité* et le comportement dynamique

Dans ce chapitre, nous nous concentrerons sur HTML et CSS, les deux piliers de toute page web.

### 1.1 Structure de base d'une page HTML5

#### Définition

**HTML5** est la cinquième et dernière version majeure du langage HTML. Elle introduit de nouvelles balises sémantiques, des API JavaScript et un meilleur support multimédia.

Voici la structure minimale d'un document HTML5 :

Listing 1: Structure de base HTML5

```

1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width,
6          initial-scale=1.0">
7      <title>Titre de la page</title>
8  </head>
9  <body>
10     <!-- Contenu visible de la page -->
11     <h1>Bienvenue sur ma page</h1>
12     <p>Ceci est un paragraphe.</p>
13 </body>
</html>
```

Explications ligne par ligne :

1. `<!DOCTYPE html>` : Déclaration du type de document (HTML5)
2. `<html lang="fr">` : Élément racine avec la langue du document
3. `<head>` : En-tête contenant les métadonnées (non affiché)
4. `<meta charset="UTF-8">` : Encodage des caractères (accents, symboles)
5. `<meta name="viewport" ...>` : Configuration pour les écrans mobiles

6. `<title>` : Titre affiché dans l'onglet du navigateur

7. `<body>` : Corps contenant le contenu visible

### Important

L'encodage **UTF-8** est essentiel pour afficher correctement les caractères accentués (é, è, à, etc.). Placez toujours cette balise en premier dans le `<head>`.

## 1.2 Afficher un document HTML5 dans le navigateur

Pour visualiser une page HTML, plusieurs options s'offrent à vous :

### 1.2.1 Méthode simple : double-clic

1. Créez un fichier avec l'extension `.html` (ex: `index.html`)
2. Double-cliquez sur le fichier
3. Le navigateur par défaut ouvre la page

### 1.2.2 Méthode recommandée : serveur local

Pour un développement professionnel, utilisez un serveur local :

#### Astuce

##### Extension Live Server (VS Code) :

1. Installez l'extension "Live Server" dans VS Code
2. Clic droit sur votre fichier HTML → "Open with Live Server"
3. La page s'actualise automatiquement à chaque modification !

## 1.3 Où écrire le CSS ?

Il existe trois façons d'intégrer du CSS dans un document HTML :

### 1.3.1 CSS en ligne (inline)

Le style est appliqué directement sur l'élément via l'attribut `style` :

```
1 <p style="color: blue; font-size: 18px;">
2   Texte bleu en 18 pixels
3 </p>
```

**Avantage** : Rapide pour tester. **Inconvénient** : Difficile à maintenir.

### 1.3.2 CSS interne (embedded)

Le style est défini dans une balise `<style>` dans le `<head>` :

```

1  <head>
2    <style>
3      p {
4        color: blue;
5        font-size: 18px;
6      }
7    </style>
8  </head>

```

**Avantage** : Tout dans un fichier. **Inconvénient** : Non réutilisable.

### 1.3.3 CSS externe (recommandé)

Le style est dans un fichier séparé `.css` lié au HTML :

```

1  <head>
2    <link rel="stylesheet" href="style.css">
3  </head>

```

Fichier `style.css` :

```

1  p {
2    color: blue;
3    font-size: 18px;
4  }

```

#### À retenir

**Privilégiez toujours le CSS externe :**

- Séparation des responsabilités (structure vs présentation)
- Réutilisation sur plusieurs pages
- Meilleure maintenance et lisibilité
- Mise en cache par le navigateur (performances)

## 1.4 Validation et compatibilité entre navigateurs

### 1.4.1 Valider son code

Un code HTML/CSS valide garantit un affichage cohérent. Utilisez les validateurs officiels du W3C :

- **HTML** : <https://validator.w3.org/>
- **CSS** : <https://jigsaw.w3.org/css-validator/>

### 1.4.2 Compatibilité navigateurs

Tous les navigateurs n'interprètent pas le CSS de la même façon. Pour vérifier la compatibilité d'une propriété :

- Can I Use : <https://caniuse.com/>
- MDN Web Docs : <https://developer.mozilla.org/>

#### Astuce

**Reset CSS** : Les navigateurs appliquent des styles par défaut différents. Un "reset CSS" ou "normalize.css" uniformise ces styles pour partir d'une base commune.

## 1.5 Bonnes pratiques, règles et commentaires

### 1.5.1 Commentaires HTML

```
1 <!-- Ceci est un commentaire HTML -->
2 <!--
3   Commentaire
4   sur plusieurs lignes
5 -->
```

### 1.5.2 Commentaires CSS

```
1 /* Ceci est un commentaire CSS */
2 /*
3   Commentaire
4   sur plusieurs lignes
5 */
```

### 1.5.3 Bonnes pratiques

1. **Indentation cohérente** : 2 ou 4 espaces (jamais de tabulations)
2. **Noms significatifs** : `.bouton-principal` plutôt que `.bp`
3. **Minuscules** : balises et attributs en minuscules
4. **Guillemets doubles** : pour les valeurs d'attributs
5. **Fermer les balises** : même si HTML5 est plus permissif
6. **Organiser le CSS** : regrouper par sections (header, main, footer)

## Important

### Évitez les erreurs courantes :

- Oublier de fermer une balise ou une accolade CSS
- Confondre `class` et `id`
- Utiliser des espaces dans les noms de classes
- Imbriquer incorrectement les balises

## 2 Les bases du HTML

### 2.1 Éléments, balises et attributs

#### Définition

**Élément HTML** : Unité de base d'un document HTML, composée d'une balise ouvrante, d'un contenu et d'une balise fermante.

**Balise** : Marqueur qui délimite un élément. Syntaxe : `<nom>` (ouvrante) et `</nom>` (fermante).

**Attribut** : Information supplémentaire ajoutée à une balise. Syntaxe : `nom="valeur"`

Listing 2: Anatomie d'un élément HTML

```

1 <a href="https://example.com" target="_blank">Cliquez ici</a>
2 <!--
3   <a>           : balise ouvrante
4   href="..."     : attribut avec sa valeur
5   target="..."   : second attribut
6   Cliquez ici   : contenu textuel
7   </a>          : balise fermante
8 -->

```

#### 2.1.1 Balises auto-fermantes

Certaines balises n'ont pas de contenu et se ferment elles-mêmes :

```

1 <br>           <!-- Saut de ligne -->
2 <hr>           <!-- Ligne horizontale -->
3 <!-- Image -->
4 <input type="text"> <!-- Champ de formulaire -->
5 <meta charset="UTF-8"> <!-- Metadonnee -->

```

## 2.2 Les éléments de texte essentiels

### 2.2.1 Titres (Headings)

HTML propose six niveaux de titres, de `<h1>` (le plus important) à `<h6>` (le moins important) :

```

1 <h1>Titre principal (un seul par page)</h1>
2 <h2>Titre de section</h2>
3 <h3>Sous-section</h3>
4 <h4>Sous-sous-section</h4>
5 <h5>Niveau 5</h5>
6 <h6>Niveau 6</h6>

```

## Important

**SEO et accessibilité :** Utilisez un seul `<h1>` par page et respectez la hiérarchie (ne passez pas de `<h2>` à `<h4>` directement).

### 2.2.2 Paragraphes et sauts de ligne

```

1 <p>Ceci est un paragraphe. Le texte est automatiquement
2     regroupe sur une ou plusieurs lignes.</p>
3
4 <p>Un autre paragraphe avec un<br>saut de ligne force.</p>

```

## Astuce

Les espaces multiples et retours à la ligne dans le code source sont ignorés par le navigateur. Utilisez `<br>` pour forcer un saut de ligne.

### 2.2.3 Mise en forme du texte

```

1 <strong>Texte important (gras sémantique)</strong>
2 <em>Texte accentué (italique sémantique)</em>
3 <mark>Texte surligné</mark>
4 <small>Texte plus petit</small>
5 <del>Texte supprimé (barre)</del>
6 <ins>Texte inséré (souligné)</ins>
7 <sub>Indice</sub> et <sup>Exposant</sup>
8 <code>Code informatique</code>

```

## À retenir

Sémantique vs Présentation :

- `<strong>` = importance sémantique (pas juste du gras visuel)
- `<em>` = emphase sémantique (pas juste de l'italique)
- Préférez ces balises à `<b>` et `<i>` pour l'accessibilité

## 2.3 Les listes

### 2.3.1 Liste non ordonnée (à puces)

```

1 <ul>
2     <li>Premier élément</li>
3     <li>Deuxième élément</li>
4     <li>Troisième élément</li>
5 </ul>

```

### 2.3.2 Liste ordonnée (numérotée)

```

1 <ol>
2   <li>Etape 1</li>
3   <li>Etape 2</li>
4   <li>Etape 3</li>
5 </ol>
6
7 <!-- Avec attributs -->
8 <ol start="5" type="A" reversed>
9   <li>Element E</li>
10  <li>Element D</li>
11 </ol>

```

Attributs de `<ol>` :

- `start` : numéro de départ
- `type` : style (1, A, a, I, i)
- `reversed` : ordre décroissant

### 2.3.3 Liste de définitions

```

1 <dl>
2   <dt>HTML</dt>
3   <dd>Langage de balisage pour structurer les pages web</dd>
4
5   <dt>CSS</dt>
6   <dd>Feuilles de style pour la présentation visuelle</dd>
7 </dl>

```

### 2.3.4 Listes imbriquées

```

1 <ul>
2   <li>Fruits
3     <ul>
4       <li>Pommes</li>
5       <li>Oranges</li>
6     </ul>
7   </li>
8   <li>Légumes
9     <ul>
10    <li>Carottes</li>
11    <li>Tomates</li>
12  </ul>
13 </li>
14 </ul>

```

## 2.4 Les liens hypertextes

La balise `<a>` (anchor) crée des liens cliquables :

### 2.4.1 Liens externes

```

1  <!-- Lien simple -->
2  <a href="https://www.google.com">Aller sur Google</a>
3
4  <!-- Ouvrir dans un nouvel onglet -->
5  <a href="https://www.google.com" target="_blank"
6      rel="noopener noreferrer">
7      Google (nouvel onglet)
8  </a>

```

#### Important

Ajoutez toujours `rel="noopener noreferrer"` avec `target="_blank"` pour des raisons de sécurité.

### 2.4.2 Liens internes

```

1  <!-- Vers une autre page du site -->
2  <a href="contact.html">Page Contact</a>
3  <a href="pages/about.html">A propos</a>
4  <a href="../index.html">Retour accueil</a>
5
6  <!-- Vers une ancre dans la page -->
7  <a href="#section2">Aller à la section 2</a>
8
9  <!-- L'ancre cible -->
10 <h2 id="section2">Section 2</h2>

```

### 2.4.3 Autres types de liens

```

1  <!-- Lien email -->
2  <a href="mailto:contact@example.com">Nous contacter</a>
3
4  <!-- Lien telephone -->
5  <a href="tel:+33123456789">Appeler</a>
6
7  <!-- Lien de telechargement -->
8  <a href="document.pdf" download>Telecharger le PDF</a>

```

## 3 Les bases du CSS

### 3.1 Sélecteurs, propriétés et valeurs

#### Définition

Une **règle CSS** se compose de :

- Un **sélecteur** : cible les éléments à styliser
- Un **bloc de déclaration** : ensemble de propriétés entre accolades
- Des **déclarations** : paires propriété-valeur séparées par ;

Listing 3: Syntaxe d'une règle CSS

```

1 selecteur {
2   propriete1: valeur1;
3   propriete2: valeur2;
4 }
5
6 /* Exemple concret */
7 p {
8   color: blue;
9   font-size: 16px;
10  line-height: 1.5;
11 }
```

#### 3.1.1 Sélecteurs de base

```

1 /* Selecteur de type (balise) */
2 p { color: black; }
3 h1 { font-size: 32px; }
4
5 /* Selecteur de classe (reutilisable) */
6 .important { color: red; }
7 .bouton { padding: 10px 20px; }
8
9 /* Selecteur d'identifiant (unique) */
10 #header { background: gray; }
11 #menu-principal { width: 100%; }
12
13 /* Selecteur universel */
14 * { margin: 0; padding: 0; }
```

## Exemple

### Utilisation en HTML :

```

1 <p class="important">Texte important en rouge</p>
2 <p class="important bouton">Peut avoir plusieurs classes</p>
3 <div id="header">Un seul element avec cet ID</div>
```

## Important

### Classe vs ID :

- **Classe (.)** : réutilisable sur plusieurs éléments
- **ID (#)** : unique, un seul élément par page
- Privilégiez les classes pour le CSS, réservez les ID pour JavaScript

## 3.2 Les éléments div et span

### Définition

**div** et **span** sont des conteneurs génériques sans signification sémantique :

- **<div>** : conteneur de type **block** (prend toute la largeur)
- **<span>** : conteneur de type **inline** (dans le flux du texte)

```

1 <!-- div : bloc pour regrouper des elements -->
2 <div class="carte">
3   <h3>Titre de la carte</h3>
4   <p>Contenu de la carte</p>
5 </div>
6
7 <!-- span : inline pour styliser du texte -->
8 <p>Le prix est de <span class="prix">29,99 EUR</span></p>
```

### 3.2.1 Différence block vs inline

Caractéristique	Block	Inline
Largeur par défaut	100% du parent	Contenu uniquement
Retour à la ligne	Oui (avant et après)	Non
Dimensions (width/height)	Modifiables	Non modifiables
Marges verticales	Appliquées	Ignorées
<b>Exemples</b>	div, p, h1-h6, ul, li	span, a, strong, em

Table 1: Comparaison block vs inline

### 3.3 Les sélecteurs avancés

#### 3.3.1 Combinateurs

```

1  /* Descendant (n'importe quel niveau) */
2  article p { color: gray; }

3
4  /* Enfant direct */
5  ul > li { list-style: square; }

6
7  /* Frere adjacent (juste apres) */
8  h2 + p { font-weight: bold; }

9
10 /* Freres generaux (tous apres) */
11 h2 ~ p { margin-left: 20px; }

```

#### 3.3.2 Sélecteurs d'attributs

```

1  /* Element avec attribut */
2  [disabled] { opacity: 0.5; }

3
4  /* Attribut avec valeur exacte */
5  [type="text"] { border: 1px solid gray; }

6
7  /* Attribut commençant par */
8  [href^="https"] { color: green; }

9
10 /* Attribut finissant par */
11 [href$=".pdf"] { color: red; }

12
13 /* Attribut contenant */
14 [class*="btn"] { cursor: pointer; }

```

#### 3.3.3 Pseudo-classes

```

1  /* Etats interactifs */
2  a:hover { text-decoration: underline; }
3  a:active { color: red; }
4  a:visited { color: purple; }
5  input:focus { border-color: blue; }

6
7  /* Position dans le parent */
8  li:first-child { font-weight: bold; }
9  li:last-child { border-bottom: none; }
10 li:nth-child(odd) { background: #f0f0f0; }
11 li:nth-child(3n) { color: red; } /* Tous les 3 */

12
13 /* Autres */
14 input:disabled { background: #ccc; }
15 input:checked + label { color: green; }

```

```
16 p:empty { display: none; }
```

### 3.3.4 Pseudo-éléments

```
1 /* Première lettre / ligne */
2 p::first-letter { font-size: 200%; }
3 p::first-line { font-weight: bold; }
4
5 /* Contenu généré */
6 .required::after {
7   content: " * ";
8   color: red;
9 }
10
11 blockquote::before {
12   content: "\201C"; /* Guillemet ouvrant */
13   font-size: 2em;
14 }
```

## 3.4 La cascade et l'héritage

### Définition

**Cascade** : Mécanisme qui détermine quelle règle s'applique quand plusieurs règles ciblent le même élément.

**Héritage** : Certaines propriétés CSS se transmettent automatiquement des parents aux enfants.

### 3.4.1 Spécificité

La spécificité détermine quelle règle "gagne". Calculez-la ainsi :

Type	Poids	Exemple
Style inline	1000	style="..."
ID	100	#header
Classe, pseudo-classe, attribut	10	.btn, :hover, [type]
Elément, pseudo-élément	1	p, ::before

Table 2: Calcul de la spécificité

```
1 /* Specificite : 0-0-1 = 1 */
2 p { color: black; }
3
4 /* Specificite : 0-1-0 = 10 */
5 .texte { color: blue; }
6
7 /* Specificite : 0-1-1 = 11 */
8 p.texte { color: green; }
9
```

```
10 /* Specificite : 1-0-0 = 100 */
11 #intro { color: red; }
12
13 /* Le rouge gagne car 100 > 11 > 10 > 1 */
```

### 3.4.2 Héritage

Propriétés héritées par défaut : `color`, `font-*`, `line-height`, `text-*...`

Propriétés NON héritées : `margin`, `padding`, `border`, `background...`

```
1 body {
2     color: #333;           /* Herite par tous les enfants */
3     font-family: Arial;   /* Herite par tous les enfants */
4 }
5
6 div {
7     border: 1px solid;    /* NON herite */
8 }
9
10 /* Forcer l'heritage */
11 .enfant {
12     border: inherit;
13 }
```

## 4 Formater du texte et positionner des éléments

### 4.1 Les propriétés de police (font-\*)

```

1 p {
2   /* Taille de police */
3   font-size: 16px;           /* Pixels (absolu) */
4   font-size: 1rem;          /* Relatif à html (recommandé) */
5   font-size: 1.2em;         /* Relatif au parent */
6
7   /* Famille de police */
8   font-family: Arial, Helvetica, sans-serif;
9   font-family: 'Times New Roman', Georgia, serif;
10  font-family: 'Courier New', monospace;
11
12  /* Style et graisse */
13  font-style: italic;        /* normal, italic, oblique */
14  font-weight: bold;         /* normal, bold, 100-900 */
15
16  /* Hauteur de ligne */
17  line-height: 1.6;          /* Sans unité = multiplicateur */
18  line-height: 24px;         /* Valeur fixe */
19
20  /* Raccourci */
21  font: italic bold 16px/1.5 Arial, sans-serif;
22}

```

#### 4.1.1 Couleur et opacité

```

1 p {
2   /* Couleurs nommées */
3   color: red;
4   color: blue;
5
6   /* Hexadecimal */
7   color: #ff0000;           /* Rouge */
8   color: #f00;              /* Raccourci */
9
10  /* RGB et RGBA */
11  color: rgb(255, 0, 0);
12  color: rgba(255, 0, 0, 0.5); /* 50% transparent */
13
14  /* HSL et HSLA */
15  color: hsl(0, 100%, 50%);    /* Teinte, Saturation, Luminosité */
16  color: hsla(0, 100%, 50%, 0.5);
17
18  /* Opacité de l'élément entier */
19  opacity: 0.8;              /* 0 = invisible, 1 = opaque */
20}

```

## 4.2 Les propriétés de texte (text-\*)

```

1 p {
2     /* Alignement */
3     text-align: left;          /* left, right, center, justify */
4
5     /* Decoration */
6     text-decoration: underline; /* none, underline, overline, line
7         -through */
8     text-decoration: underline wavy red; /* Style et couleur */
9
10    /* Transformation */
11   text-transform: uppercase; /* uppercase, lowercase,
12       capitalize, none */
13
14    /* Indentation (premiere ligne) */
15   text-indent: 2em;
16
17    /* Espacement */
18   letter-spacing: 2px; /* Entre les lettres */
19   word-spacing: 5px; /* Entre les mots */
20
21    /* Ombre du texte */
22   text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
23       X      Y      flou    couleur */
24 }
```

## 4.3 Le modèle des boîtes (Box Model)

### Définition

Le **Box Model** définit comment chaque élément HTML est rendu :

1. **Content** : le contenu (texte, image...)
2. **Padding** : espace intérieur (entre contenu et bordure)
3. **Border** : la bordure
4. **Margin** : espace extérieur (entre l'élément et les autres)

### 4.3.1 Dimensions

```

1 .box {
2     /* Dimensions fixes */
3     width: 300px;
4     height: 200px;
5
6     /* Dimensions relatives */
7     width: 50%;           /* 50% du parent */
```

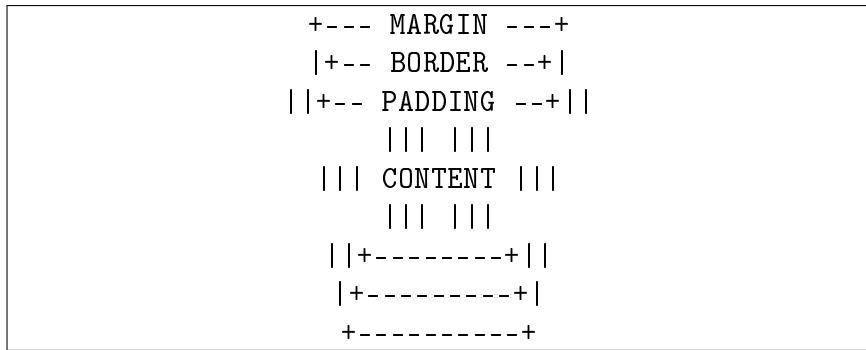


Figure 1: Schéma du Box Model

```

8  width: 100vw;           /* 100% de la largeur viewport */
9  height: 100vh;          /* 100% de la hauteur viewport */
10
11 /* Limites */
12 min-width: 200px;
13 max-width: 800px;
14 min-height: 100px;
15 max-height: 500px;
16
17 /* IMPORTANT : box-sizing */
18 box-sizing: border-box; /* width inclut padding et border */
19 box-sizing: content-box; /* Par défaut : width = contenu seul */
20

```

### Important

Ajoutez toujours `box-sizing: border-box` à tous vos éléments pour un calcul intuitif des dimensions :

```

1  * , *::before, *::after {
2    box-sizing: border-box;
3  }

```

### 4.3.2 Bordures

```

1 .box {
2   /* Syntaxe complète */
3   border-width: 2px;
4   border-style: solid;    /* solid, dashed, dotted, double, none */
5   border-color: #333;
6
7   /* Raccourci */
8   border: 2px solid #333;
9
10  /* Par côte */

```

```

11 border-top: 1px solid red;
12 border-right: 2px dashed blue;
13 border-bottom: 3px dotted green;
14 border-left: none;
15
16 /* Coins arrondis */
17 border-radius: 10px; /* Tous les coins */
18 border-radius: 10px 20px; /* Haut-gauche/bas-droite,
   haut-droite/bas-gauche */
19 border-radius: 10px 20px 30px 40px; /* HG, HD, BD, BG */
20 border-radius: 50%; /* Cercle parfait */
21 }

```

### 4.3.3 Marges intérieures (padding)

```

1 .box {
2   /* Par cote */
3   padding-top: 10px;
4   padding-right: 20px;
5   padding-bottom: 10px;
6   padding-left: 20px;
7
8   /* Raccourcis */
9   padding: 10px; /* 4 cotes identiques */
10  padding: 10px 20px; /* Vertical, Horizontal */
11  padding: 10px 20px 15px; /* Haut, Horizontal, Bas */
12  padding: 10px 20px 15px 25px; /* Haut, Droite, Bas, Gauche */
13 }

```

### 4.3.4 Marges extérieures (margin)

```

1 .box {
2   /* Meme syntaxe que padding */
3   margin: 20px;
4   margin: 10px 20px;
5   margin: 10px auto; /* Centrer horizontalement un bloc */
6
7   /* Marges négatives (possible) */
8   margin-top: -10px; /* Chevauche l'element précédent */
9 }

```

## Astuce

Centrer un bloc horizontalement :

```

1 .container {
2   width: 800px;           /* Largeur definie obligatoire */
3   margin: 0 auto;         /* Marges gauche/droite automatiques
4   */

```

### 4.3.5 Ombres des boîtes

```

1 .box {
2   /* Ombre externe (par defaut) */
3   box-shadow: 5px 5px 10px rgba(0, 0, 0, 0.3);
4   /*           X      Y      flou      couleur */
5
6   /* Ombre interne */
7   box-shadow: inset 0 0 10px rgba(0, 0, 0, 0.5);
8
9   /* Ombres multiples */
10  box-shadow:
11    0 2px 5px rgba(0, 0, 0, 0.2),
12    0 5px 15px rgba(0, 0, 0, 0.1);
13 }

```

## 4.4 La propriété display

```

1 .element {
2   /* Valeurs classiques */
3   display: block;          /* Prend toute la largeur */
4   display: inline;         /* Dans le flux du texte */
5   display: inline-block;    /* Inline mais avec dimensions
6   modifiables */
6   display: none;          /* Cache completement l'element */
7
8   /* Valeurs modernes */
9   display: flex;           /* Conteneur flexbox */
10  display: grid;           /* Conteneur grid */
11 }

```

### 4.4.1 Flexbox (bases)

```

1 .container {
2   display: flex;
3
4   /* Direction des elements */
5   flex-direction: row;       /* Horizontal (defaut) */

```

```

6   flex-direction: column;           /* Vertical */
7
8   /* Alignement sur l'axe principal */
9   justify-content: flex-start;    /* Debut */
10  justify-content: center;       /* Centre */
11  justify-content: flex-end;      /* Fin */
12  justify-content: space-between; /* Espaces entre */
13  justify-content: space-around; /* Espaces autour */
14
15  /* Alignement sur l'axe secondaire */
16  align-items: stretch;          /* Etire (defaut) */
17  align-items: center;           /* Centre */
18  align-items: flex-start;        /* Debut */
19  align-items: flex-end;          /* Fin */
20
21  /* Espace entre les elements */
22  gap: 20px;
23 }

```

## Exemple

Centrer parfaitement un élément :

```

1 .container {
2   display: flex;
3   justify-content: center; /* Centre horizontal */
4   align-items: center;     /* Centre vertical */
5   height: 100vh;          /* Hauteur du viewport */
6 }

```

## 4.5 La propriété position

```

1 .element {
2   position: static;      /* Par defaut, flux normal */
3   position: relative;    /* Positionne par rapport a sa position
                           normale */
4   position: absolute;    /* Positionne par rapport au parent
                           positionne */
5   position: fixed;       /* Positionne par rapport au viewport (fixe) */
6   position: sticky;      /* Hybride : relative puis fixed */
7 }

```

### 4.5.1 Relative

```

1 .element {
2   position: relative;
3   top: 10px;           /* Decale de 10px vers le bas */
4   left: 20px;          /* Decale de 20px vers la droite */

```

```

5  /* L'espace original est conserve dans le flux */
6 }

```

#### 4.5.2 Absolute

```

1 .parent {
2   position: relative; /* Reference pour l'enfant absolu */
3 }
4
5 .enfant {
6   position: absolute;
7   top: 0;
8   right: 0;           /* Coin superieur droit du parent */
9   /* Sort du flux normal */
10 }

```

#### 4.5.3 Fixed

```

1 .navbar {
2   position: fixed;
3   top: 0;
4   left: 0;
5   width: 100%;
6   /* Reste fixe lors du scroll */
7 }

```

#### 4.5.4 Sticky

```

1 .header {
2   position: sticky;
3   top: 0;
4   /* Se comporte comme relative jusqu'a top: 0, puis fixed */
5 }

```

### 4.6 Z-index et empilement

```

1 .element {
2   position: relative; /* z-index necessite un position != static */
3   z-index: 10;          /* Plus eleve = au-dessus */
4 }
5
6 .autre {
7   position: absolute;
8   z-index: 5;           /* En dessous de .element */
9 }

```

**Important**

Le `z-index` ne fonctionne que sur les éléments positionnés (`relative`, `absolute`, `fixed`, `sticky`).

## 5 Fonctionnalités avancées

### 5.1 Gestion du background

#### 5.1.1 Couleur de fond

```

1  .element {
2    background-color: #f5f5f5;
3    background-color: rgba(255, 255, 255, 0.9);
4 }
```

#### 5.1.2 Image de fond

```

1  .hero {
2    /* Image de fond */
3    background-image: url('images/hero.jpg');
4
5    /* Position */
6    background-position: center center;
7    background-position: top left;
8    background-position: 50% 50%;
9
10   /* Repetition */
11  background-repeat: no-repeat; /* repeat, repeat-x, repeat-y */
12
13   /* Taille */
14  background-size: cover;      /* Couvre tout, peut rogner */
15  background-size: contain;    /* Contenu entier, peut laisser
16    des espaces */
17  background-size: 100% auto;
18
19   /* Attachement */
20  background-attachment: fixed; /* Effet parallaxe */
21  background-attachment: scroll; /* Defaut */
22
23   /* Raccourci */
24  background: url('hero.jpg') center/cover no-repeat fixed;
```

#### 5.1.3 Images multiples et dégradés

```

1  .element {
2    /* Plusieurs images (la premiere est au-dessus) */
3    background-image:
4      url('overlay.png'),
5      url('background.jpg');
6
7    /* Degrade linéaire */
8    background: linear-gradient(to right, #ff6b6b, #4ecdc4);
```

```

9  background: linear-gradient(45deg, red, blue);
10 background: linear-gradient(to bottom,
11   #fff 0%,
12   #f0f0f0 50%,
13   #ddd 100%);
14
15 /* Degrade radial */
16 background: radial-gradient(circle, #fff, #000);
17 background: radial-gradient(ellipse at top left, red, blue);
18 }

```

## 5.2 Intégrer des images, de l'audio et de la vidéo

### 5.2.1 Images

```

1 <!-- Image simple -->
2 
3
4 <!-- Image avec dimensions -->
5 
6
7 <!-- Image responsive -->
8 
9
10 <!-- Avec figure et légende -->
11 <figure>
12   
13   <figcaption>Figure 1 : Evolution des ventes 2024</figcaption>
14 </figure>

```

#### Important

L'attribut **alt** est **obligatoire** pour l'accessibilité. Décrivez l'image pour les personnes malvoyantes et les moteurs de recherche.

### 5.2.2 Audio

```

1 <audio controls>
2   <source src="musique.mp3" type="audio/mpeg">
3   <source src="musique.ogg" type="audio/ogg">
4   Votre navigateur ne supporte pas l'élément audio.
5 </audio>
6
7 <!-- Avec attributs -->
8 <audio controls autoplay loop muted>
9   <source src="musique.mp3" type="audio/mpeg">
10 </audio>

```

### 5.2.3 Vidéo

```

1 <video controls width="640" height="360" poster="apercu.jpg">
2   <source src="video.mp4" type="video/mp4">
3   <source src="video.webm" type="video/webm">
4   Votre navigateur ne supporte pas l'element video.
5 </video>
6
7 <!-- Attributs disponibles -->
8 <!-- controls : affiche les controles -->
9 <!-- autoplay : lecture automatique (souvent bloquee) -->
10 <!-- loop : boucle infinie -->
11 <!-- muted : son coupe -->
12 <!-- poster : image d'apercu -->
```

## 5.3 Les tableaux

### 5.3.1 Structure de base

```

1 <table>
2   <caption>Liste des etudiants</caption>
3
4   <thead>
5     <tr>
6       <th>Nom</th>
7       <th>Prenom</th>
8       <th>Note</th>
9     </tr>
10    </thead>
11
12   <tbody>
13     <tr>
14       <td>Dupont</td>
15       <td>Marie</td>
16       <td>16/20</td>
17     </tr>
18     <tr>
19       <td>Martin</td>
20       <td>Pierre</td>
21       <td>14/20</td>
22     </tr>
23   </tbody>
24
25   <tfoot>
26     <tr>
27       <td colspan="2">Moyenne</td>
28       <td>15/20</td>
29     </tr>
30   </tfoot>
31 </table>
```

### 5.3.2 Fusion de cellules

```

1 <!-- Fusion horizontale -->
2 <td colspan="3">Cette cellule occupe 3 colonnes</td>
3
4 <!-- Fusion verticale -->
5 <td rowspan="2">Cette cellule occupe 2 lignes</td>

```

### 5.3.3 Mise en forme CSS

```

1 table {
2   width: 100%;
3   border-collapse: collapse; /* Fusionne les bordures */
4   margin: 20px 0;
5 }
6
7 th, td {
8   border: 1px solid #ddd;
9   padding: 12px;
10  text-align: left;
11 }
12
13 th {
14   background-color: #4CAF50;
15   color: white;
16 }
17
18 /* Lignes alternées */
19 tbody tr:nth-child(even) {
20   background-color: #f9f9f9;
21 }
22
23 /* Survol */
24 tbody tr:hover {
25   background-color: #f1f1f1;
26 }

```

## 5.4 Les formulaires

### 5.4.1 Structure de base

```

1 <form action="/submit" method="POST">
2   <!-- Champ texte -->
3   <label for="nom">Nom :</label>
4   <input type="text" id="nom" name="nom" placeholder="Votre nom"
      required>
5
6   <!-- Email -->
7   <label for="email">Email :</label>

```

```

8   <input type="email" id="email" name="email" required>
9
10  <!-- Mot de passe -->
11  <label for="mdp">Mot de passe :</label>
12  <input type="password" id="mdp" name="mdp" minlength="8">
13
14  <!-- Bouton de soumission -->
15  <button type="submit">Envoyer</button>
16 </form>

```

#### 5.4.2 Types de champs input

```

1  <!-- Texte -->
2  <input type="text" placeholder="Texte libre">
3  <input type="email" placeholder="email@example.com">
4  <input type="password" placeholder="Mot de passe">
5  <input type="tel" placeholder="06 12 34 56 78">
6  <input type="url" placeholder="https://..."/>
7  <input type="search" placeholder="Rechercher..."/>
8
9  <!-- Nombres -->
10 <input type="number" min="0" max="100" step="5">
11 <input type="range" min="0" max="100" value="50">
12
13 <!-- Date et heure -->
14 <input type="date">
15 <input type="time">
16 <input type="datetime-local">
17 <input type="month">
18 <input type="week">
19
20 <!-- Autres -->
21 <input type="color" value="#ff0000">
22 <input type="file" accept="image/*" multiple>
23 <input type="hidden" name="token" value="abc123">

```

#### 5.4.3 Zone de texte multi-lignes

```

1  <label for="message">Message :</label>
2  <textarea id="message" name="message" rows="5" cols="40"
3           placeholder="Votre message..."></textarea>

```

#### 5.4.4 Cases à cocher et boutons radio

```

1  <!-- Cases à cocher (choix multiple) -->
2  <fieldset>
3    <legend>Interets</legend>
4    <label>

```

```

5      <input type="checkbox" name="interets" value="sport"> Sport
6  </label>
7  <label>
8      <input type="checkbox" name="interets" value="musique">
9          Musique
10 </label>
11 <label>
12     <input type="checkbox" name="interets" value="lecture"
13         checked> Lecture
14 </label>
15 </fieldset>
16
17 <!-- Boutons radio (choix unique) -->
18 <fieldset>
19     <legend>Genre</legend>
20     <label>
21         <input type="radio" name="genre" value="h" required> Homme
22     </label>
23     <label>
24         <input type="radio" name="genre" value="f"> Femme
25     </label>
26     <label>
27         <input type="radio" name="genre" value="autre"> Autre
28     </label>
29 </fieldset>

```

#### 5.4.5 Listes déroulantes

```

1  <label for="pays">Pays :</label>
2  <select id="pays" name="pays">
3      <option value="">-- Choisir --</option>
4      <option value="fr">France</option>
5      <option value="be">Belgique</option>
6      <option value="ch" selected>Suisse</option>
7      <optgroup label="Amerique">
8          <option value="us">Etats-Unis</option>
9          <option value="ca">Canada</option>
10     </optgroup>
11 </select>
12
13 <!-- Selection multiple -->
14 <select name="langues" multiple size="4">
15     <option value="fr">Francais</option>
16     <option value="en">Anglais</option>
17     <option value="es">Espagnol</option>
18     <option value="de">Allemand</option>
19 </select>

```

#### 5.4.6 Attributs de validation

```

1 <input type="text" required>                                <!-- Obligatoire
2 <input type="text" minlength="3" maxlength="50"> <!-- Longueur
3 <input type="number" min="0" max="100">                  <!-- Valeurs min/
   max -->
4 <input type="text" pattern="[A-Za-z]{3,}>                <!-- Expression
   reguliere -->
5 <input type="email" readonly>                            <!-- Lecture
   seule -->
6 <input type="text" disabled>                             <!-- Desactive
   -->
7 <input type="text" autofocus>                            <!-- Focus
   automatique -->

```

#### 5.4.7 Mise en forme des formulaires

```

1 form {
2   max-width: 500px;
3   margin: 0 auto;
4   padding: 20px;
5 }
6
7 label {
8   display: block;
9   margin-bottom: 5px;
10  font-weight: bold;
11 }
12
13 input, textarea, select {
14   width: 100%;
15   padding: 10px;
16   margin-bottom: 15px;
17   border: 1px solid #ddd;
18   border-radius: 5px;
19   font-size: 16px;
20 }
21
22 input:focus, textarea:focus, select:focus {
23   outline: none;
24   border-color: #4CAF50;
25   box-shadow: 0 0 5px rgba(76, 175, 80, 0.3);
26 }
27
28 button[type="submit"] {
29   background: #4CAF50;
30   color: white;
31   padding: 12px 30px;
32   border: none;
33   border-radius: 5px;

```

```
34     cursor: pointer;
35     font-size: 16px;
36   }
37
38 button[type="submit"] :hover {
39   background: #45a049;
40 }
```

## 6 Résumé et points clés

### À retenir

#### HTML - À retenir :

- Structure de base : `<!DOCTYPE html>, <html>, <head>, <body>`
- Toujours spécifier `charset="UTF-8"` et `lang="fr"`
- Utiliser les balises sémantiques (`<strong>`, `<em>`, `<header>`, `<nav>`...)
- Respecter la hiérarchie des titres (h1 → h6)
- Attribut `alt` obligatoire sur les images
- Lier labels et inputs avec `for/id`

### À retenir

#### CSS - À retenir :

- Privilégier le CSS externe (`<link rel="stylesheet">`)
- Utiliser les classes plutôt que les IDs pour le style
- Comprendre la spécificité : inline > ID > classe > élément
- Toujours utiliser `box-sizing: border-box`
- Flexbox pour les mises en page modernes
- Propriétés héritées vs non héritées

### À retenir

#### Bonnes pratiques :

- Valider son code sur [validator.w3.org](http://validator.w3.org)
- Tester sur plusieurs navigateurs
- Utiliser des noms de classes explicites
- Commenter son code
- Organiser ses fichiers (structure de dossiers claire)

## Prochain chapitre : JavaScript

*Nous verrons comment ajouter de l'interactivité à nos pages web !*