

COURS 1 : BASES DU HTML

1. Pourquoi apprendre le HTML et le CSS

HTML et CSS sont les **fondations du web**.

- Le **HTML** sert à créer la structure et le contenu d'une page web.
- Le **CSS** sert à rendre cette page esthétique et lisible.

Sans HTML, il n'y a pas de contenu. Sans CSS, le contenu existe mais reste brut et peu attractif. Ces deux langages sont indispensables pour créer des sites web.

2. Définition et rôle du HTML et du CSS

HTML (HyperText Markup Language – 1991)

- C'est un **langage de balisage**, pas un langage de programmation.
- Il sert à **indiquer au navigateur la nature du contenu** :
 - ceci est un titre
 - ceci est un paragraphe
 - ceci est une image, un lien, une liste, etc.
- Le navigateur lit le HTML pour savoir **comment afficher les informations**.

CSS (Cascading Style Sheets – 1996)

- C'est un **langage de styles**.
- Il sert à **mettre en forme le contenu HTML** :
 - changer la couleur du texte
 - modifier la taille
 - ajouter des marges, des bordures
 - gérer la disposition des éléments

HTML = structure

CSS = présentation

3. Travailler en local ou en production

- **En local** : le site est sur ton ordinateur. Tu ouvres les fichiers avec un navigateur pour tester ton code.
- **En production** : le site est en ligne, hébergé sur un serveur et accessible à tout le monde via Internet.

4. Les éditeurs de texte

Un éditeur de texte permet d'écrire du code HTML et CSS.

Exemples : VS Code, Sublime Text, Notepad++.

Ils offrent des aides comme la coloration syntaxique et l'auto-complétion.

5. Éléments, balises et attributs HTML

- Une **balise** sert à entourer un contenu pour lui donner un sens.
- Un **élément HTML** = balise + contenu.
- Les **attributs** donnent des informations supplémentaires à une balise.

Exemple :

- `<a href="https://example.com">Lien</a>`
href est un attribut.

Balise orpheline

Certaines balises n'ont pas de contenu :

- `<br>` ou `<br />`
 - `<img>`
-

6. Structure minimale d'une page HTML

Une page HTML doit respecter une structure de base :

- `<!DOCTYPE html>` : indique au navigateur que le document est en HTML
 - `<html>` : englobe tout le code HTML
 - `<head>` : contient les informations non visibles
 - `<title>` : titre affiché dans l'onglet du navigateur
 - `<meta charset="utf-8">` : définit l'encodage des caractères
 - `<body>` : contient le contenu visible de la page
-

7. Création d'une page HTML

- Créer un nouveau fichier

- Lui donner l'extension `.html`
 - Écrire la structure HTML minimale
-

8. Importance de l'indentation et des commentaires

- **L'indentation** rend le code lisible et compréhensible.
- Les **commentaires** servent à expliquer le code sans être affichés.

Types de commentaires :

- Monoligne
- Multiligne

Syntaxe :

```
<!-- Ceci est un commentaire -->
```

9. Titres et paragraphes

- `<h1>` à `<h6>` : titres, du plus important au moins important
 - `<p>` : paragraphe de texte
-

10. Espaces et retours à la ligne

- `<br>` : retour à la ligne
 - `<pre>` : conserve exactement les espaces et retours à la ligne écrits dans le code
-

11. Gérer l'importance du texte

- `<strong>` : texte important (souvent en gras)
 - `<em>` : texte mis en valeur (souvent en italique)
 - `<mark>` : surlignage du texte
-

12. Les listes en HTML

Les listes permettent d'organiser le contenu.

a) Liste non ordonnée

- `<ul>` : liste
- `<li>` : élément de liste

b) Liste ordonnée

- `<ol>` : liste ordonnée
- Attribut `type` pour changer le style (chiffres, lettres, chiffres romains)

c) Liste de description

- `<dl>` : liste
- `<dt>` : terme
- `<dd>` : description

d) Liste imbriquée

Une liste peut contenir une autre liste.

13. Les liens en HTML

La balise `<a>` permet de créer des liens.

- **Lien externe** : vers un autre site
 - **Lien interne** : vers une autre page ou section du même site
-

14. Autres utilisations de `<a>`

- Lien ancre : navigation dans la même page
 - Lien email : `mailto:`
 - Lien de téléchargement : lien vers un fichier
-

COURS 2 : HTML AVANCÉ ET INTRODUCTION AU CSS

1. Insertion des images

- Balise `<img>` (orpheline)

- `src` : chemin de l'image
 - `alt` : description de l'image (accessibilité et SEO)
-

2. Validation du code HTML et entités HTML

Les entités HTML permettent d'afficher des caractères réservés :

- `<` pour `<`
 - `>` pour `>`
 - `&` pour `&`
-

3. Découvrir le CSS

Un **sélecteur** CSS permet de cibler un élément HTML pour lui appliquer un style.

Où écrire le CSS ?

1. Dans l'attribut `style` (non recommandé)
2. Dans une balise `<style>` (peu recommandé)
3. Dans un fichier CSS externe (méthode recommandée)

Lien avec HTML :

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

4. Commenter le code CSS

Syntaxe :

```
/* commentaire */
```

Fonctionne pour une ligne ou plusieurs lignes.

5. Les sélecteurs CSS

- Sélecteur simple : cible une balise (`p`, `h1`)
- Groupage de sélecteurs
- `id` : unique, ciblé avec `#`
- `class` : réutilisable, ciblée avec `.`

Héritage

Certains styles sont transmis des parents vers les enfants.

6. Types d'éléments HTML

- **Block** : prend toute la largeur (`div`, `p`, `h1`)
 - **Inline** : prend seulement l'espace nécessaire (`span`, `a`)
-

7. Div et span

- `div` : élément block, sert de conteneur
- `span` : élément inline, sert à styliser une partie du texte
Ils n'ont pas de sens sémantique.

1. Les couleurs en CSS

La propriété `color` permet de définir la **couleur du texte**.

La propriété `background-color` permet de définir la **couleur de fond** d'un élément.

Différentes façons de définir une couleur

- **Nom de couleur** : `red`, `blue`, `black`
- **Hexadécimal** : `#ff0000`
- **RGB** : `rgb(255, 0, 0)`
- **RGBA** : `rgba(255, 0, 0, 0.5)` (avec transparence)

Exemple :

```
p {  
  color: blue;  
  background-color: lightgray;  
}
```

2. L'opacité

La propriété `opacity` permet de gérer le **niveau de transparence d'un élément**.

- Valeur comprise entre **0 et 1**
 - 0 : totalement transparent

- 1 : totalement visible

Important : `opacity` s'applique à **tout l'élément et à ses enfants** (texte, image, fond).

Exemple :

```
div {  
  opacity: 0.5;  
}
```

Pour rendre seulement le fond transparent sans affecter le texte, on utilise plutôt `rgba()` avec `background-color`.

3. Les propriétés de texte

a) Alignement du texte

`text-align` permet d'aligner le texte horizontalement.

- `left`
- `right`
- `center`
- `justify`

Exemple :

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

b) Décoration du texte

`text-decoration` sert à ajouter ou supprimer une décoration.

- `underline` : souligné
- `overline` : ligne au-dessus
- `line-through` : barré
- `none` : aucune décoration

Souvent utilisé pour enlever le soulignement des liens.

c) Transformation du texte

text-transform modifie l'apparence des lettres.

- `uppercase` : tout en majuscules
 - `lowercase` : tout en minuscules
 - `capitalize` : première lettre en majuscule
 - `none` : aucune transformation
-

d) Espacement entre les mots

word-spacing permet d'augmenter ou réduire l'espace entre les mots.

Exemple :

```
p {  
  word-spacing: 5px;  
}
```

e) Espacement entre les lettres

letter-spacing permet de régler l'espace entre les caractères.

Exemple :

```
h1 {  
  letter-spacing: 3px;  
}
```

4. Les propriétés de mise en page (box model)

En CSS, chaque élément est considéré comme une **boîte**. Les propriétés suivantes servent à gérer cette boîte.

a) Margin

margin définit l'espace **extérieur** autour d'un élément (espace entre les éléments).

Exemple :


```
div {  
  margin: 20px;  
}
```

b) Padding

padding définit l'espace **intérieur** entre le contenu et la bordure.

Exemple :

```
div {  
  padding: 15px;  
}
```

c) Border

border permet d'ajouter une bordure autour d'un élément.

Elle est composée de :

- épaisseur
- style (solid, dashed, dotted)
- couleur

Exemple :

```
div {  
  border: 2px solid black;  
}
```

d) Width (largeur)

width définit la largeur d'un élément.

Valeurs courantes :

- px
- %
- auto

Exemple :

```
div {  
  width: 300px;  
}
```

e) Height (hauteur)

height définit la hauteur d'un élément.

Exemple :

```
div {  
  height: 200px;  
}
```

5. Récapitulatif logique pour ton résumé

Couleurs et transparence

- `color`
- `background-color`
- `opacity`
- `rgba()`

Propriétés de texte

- `text-align`
- `text-decoration`
- `text-transform`
- `word-spacing`
- `letter-spacing`

Mise en page

- `margin`
 - `padding`
 - `border`
 - `width`
 - `height`
-