

---

## **Activité**

### **Découverte HTML et CSS**

---

## **Exercice**

1. **Mettez en place tous les éléments HTML et CSS présentés dans un nouveau document HTML et CSS (à vous de créer les document HTML et CSS).**
2. **Prenez note de chaque élément et de leur implication.**

## **Introduction**

### **Présentation du HTML comme langage de structure et du CSS comme langage de style**

Le HTML (Hypertext Markup Language) est un langage de balisage utilisé pour structurer le contenu d'une page web.

Il permet de définir la structure sémantique d'un document, en organisant les éléments tels que les titres, paragraphes, liens et images.

Le HTML utilise des balises, identifiées par des chevrons (<>), pour marquer et délimiter ces différents éléments.

Le CSS (Cascading Style Sheets), quant à lui, est un langage de style qui complète le HTML. Il est utilisé pour définir l'apparence et la mise en forme des éléments HTML.

Le CSS permet de contrôler des aspects visuels tels que les couleurs, les polices, les marges, et le positionnement des éléments sur la page. La séparation entre HTML (structure) et CSS (style) présente plusieurs avantages :

1. **Clarté du code** : Le HTML se concentre sur le contenu et sa structure, tandis que le CSS gère l'apparence.
2. **Maintenance facilitée** : Les modifications de style peuvent être effectuées indépendamment de la structure.
3. **Réutilisation** : Un même fichier CSS peut être appliqué à plusieurs pages HTML.

En résumé, le HTML fournit la structure et le contenu de base d'une page web, tandis que le CSS permet de la styliser et de la rendre visuellement attrayante.

# 1. La partie HTML

## 1.1 Structure de base d'un document HTML

La structure de base d'un document HTML se compose des éléments suivants :

1. Déclaration du type de document : `<!DOCTYPE html>`

Cette déclaration indique qu'il s'agit d'un document HTML.

2. Élément racine `<html>` :

Il englobe tout le contenu de la page et peut inclure l'attribut `lang` pour spécifier la langue du document

3. En-tête `<head>` :

Contient les métadonnées et informations sur le document, notamment :

`<title>` : Définit le titre de la page affiché dans l'onglet du navigateur

`<meta charset="utf-8">` : Spécifie l'encodage des caractères

`<meta name="viewport" content="width=device-width">` : Assure un bon rendu sur les appareils mobiles

4. Corps `<body>` :

Contient tout le contenu visible de la page web

*Voici un exemple de structure de base d'un document HTML :*

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Titre de la page</title>
</head>
<body>
  <!-- Contenu de la page -->
</body>
</html>
```

**Note :** La balise `<!-- commentaire -->` représente un commentaire, c'est-à-dire une indication technique ou fonctionnelle. Elle ne sera en aucun cas interprétée et donc affichée par le navigateur Web.

## 1.2 Suite du chapitre HTML

La suite du chapitre présente de façon non exhaustive des balises à utiliser afin de mieux structurer les informations que vous voulez afficher dans une page Web.

Vous pouvez aussi consulter ce site pour en savoir plus :

[https://developer.mozilla.org/fr/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/HTML_basics)

## 1.3 Titres (<h1> à <h6>)

Les titres en HTML sont utilisés pour structurer le contenu d'une page web et hiérarchiser les informations. Ils vont de <h1> à <h6>, chaque niveau représentant une importance décroissante. Voici un aperçu de ces balises :

### Balises de titre

#### 1. `<h1>`

- **Description** : C'est le titre principal de la page. Il doit être unique par page et représente le sujet principal.
- **Utilisation** : Utilisé pour le titre principal, souvent visible en haut de la page.
- **Exemple** :

xml

```
<h1>Bienvenue sur mon site web</h1>
```



#### 2. `<h2>`

- **Description** : Utilisé pour les sous-titres ou les sections principales de la page.
- **Utilisation** : Indique les sections importantes qui relèvent du titre principal.
- **Exemple** :

xml

```
<h2>À propos de nous</h2>
```



## 1.4 Paragraphes (<p>)

Les paragraphes en HTML sont définis par la balise <p>, qui est utilisée pour structurer et organiser le texte dans un document. Voici un aperçu détaillé de l'utilisation des paragraphes :

### Balise de paragraphe <p>

- **Description** : La balise <p> est utilisée pour encapsuler un bloc de texte qui constitue un paragraphe. Elle permet de séparer le contenu textuel en sections logiques et lisibles.
- **Utilisation** : Chaque balise <p> crée un nouveau paragraphe, avec une marge par défaut qui aide à distinguer visuellement les blocs de texte.

### Exemple d'utilisation

Voici un exemple simple d'un document HTML utilisant des paragraphes :

```
xml

<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Exemple de Paragraphes</title>
</head>
<body>
  <h1>Bienvenue sur mon site web</h1>

  <p>Nous sommes ravis de vous accueillir sur notre site. Notre objectif
est de fournir des informations utiles et intéressantes sur divers sujets.</
p>

  <p>Notre équipe est composée de professionnels passionnés qui
travaillent dur pour vous offrir le meilleur contenu possible.</p>

  <p>Merci de votre visite et n'hésitez pas à explorer nos différentes
sections !</p>
</body>
</html>
```

## 1.5 Mise en forme de texte : gras (<strong>) et italique (<em>)

La mise en forme du texte est essentielle pour améliorer la lisibilité et la compréhension du contenu sur une page web. En HTML, les balises <strong> et <em> sont couramment utilisées pour mettre en valeur du texte. Voici un aperçu de ces balises :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Mise en forme de texte</title>
</head>
<body>
  <h1>Bienvenue sur notre site</h1>

  <p>Pour réussir, il est <strong>important</strong> de suivre les étapes
correctement.</p>

  <p>Nous vous recommandons de lire ce guide <em>avec attention</em>
avant de commencer.</p>

  <p><strong><em>Note :</em></strong> Assurez-vous d'avoir tous les
matériaux nécessaires avant de débiter.</p>
</body>
</html>
```

## 1.6 Liens hypertexte (<a href="">)

Les liens hypertexte en HTML sont créés à l'aide de la balise <a>, qui permet de naviguer d'une page à une autre ou de se connecter à des ressources externes. Voici un aperçu détaillé de leur utilisation.

### Syntaxe

La syntaxe de base pour créer un lien hypertexte est la suivante :

xml

```
<a href="URL">Texte du lien</a>
```



- `href` : Attribut qui spécifie l'URL de destination du lien.
- `Texte du lien` : Le texte cliquable visible par l'utilisateur.

### Types de liens

#### 1. Liens vers des pages web externes :

Ces liens redirigent vers d'autres sites web.

xml

```
<a href="https://www.geeksforgeeks.org">Visitez GeeksforGeeks</a>
```



#### 2. Liens vers des documents internes :

Ils pointent vers d'autres pages au sein du même site.

xml

```
<a href="page2.html">Aller à la page 2</a>
```



## 1.7 Intégration d'images (<img src="">)

L'intégration d'images dans une page web est une compétence essentielle en HTML, réalisée principalement à l'aide de la balise <img>. Voici un aperçu détaillé de son utilisation.

### Syntaxe de base

La balise `<img>` est utilisée pour afficher des images sur une page web. Sa syntaxe de base est la suivante :

xml

```

```



- `src` : Attribut obligatoire qui spécifie l'URL ou le chemin de l'image à afficher.
- `alt` : Attribut obligatoire qui fournit un texte alternatif descriptif de l'image. Ce texte est affiché si l'image ne peut pas être chargée et est également utilisé par les lecteurs d'écran pour l'accessibilité.

**Note :** Vous remplacerez « URL\_de\_l\_image » par un nom de fichier représentant une image. Vous veillerez à ne pas oublier l'extension du nom de fichier (Ex : jpg, png etc...)



## 1.8 Divisions (<div>)

La balise <div> en HTML est un élément fondamental utilisé pour structurer et organiser le contenu d'une page web. Voici un aperçu détaillé de son utilisation, de ses caractéristiques et de son importance.

### Qu'est-ce qu'un <div> ?

La balise <div> est un conteneur de niveau bloc qui permet de regrouper d'autres éléments HTML. Elle ne possède pas de style ou de signification sémantique intrinsèque, mais elle est souvent utilisée en combinaison avec CSS pour appliquer des styles et structurer le contenu.

### Caractéristiques du <div>

- **Élément de bloc** : Par défaut, un <div> occupe toute la largeur disponible et commence sur une nouvelle ligne, ce qui signifie qu'il crée des sauts de ligne avant et après lui.
- **Pas de style par défaut** : Contrairement à d'autres éléments HTML qui peuvent avoir des styles par défaut, un <div> est neutre jusqu'à ce qu'il soit stylisé avec CSS.
- **Conteneur flexible** : Vous pouvez placer presque n'importe quel type d'élément HTML à l'intérieur d'un <div>, y compris d'autres <div>, ce qui permet une grande flexibilité dans la mise en page.

## 2. La partie CSS

### 2.1 Création d'un fichier CSS externe

Pour créer un fichier CSS externe et l'intégrer dans un document HTML, suivez ces étapes :

#### Étape 1 : Créer le fichier CSS

1. **Ouvrez un éditeur de texte** (notepad++)
2. **Écrivez vos règles CSS** dans le fichier. Par exemple :

CSS

```
/* mystyle.css */
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}

p {
    font-size: 18px;
    line-height: 1.6;
}
```

3. **Enregistrez le fichier** avec l'extension .css, par exemple mystyle.css.

## Étape 2 : Lier le fichier CSS au document HTML

1. **Ouvrez ou créez votre fichier HTML** dans l'éditeur de texte.
2. **Ajoutez la balise <link>** dans la section <head> pour lier le fichier CSS. Voici un exemple :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Mon Site Web</title>
  <link rel="stylesheet" href="mystyle.css"> <!-- Lien vers le fichier
CSS -->
</head>
<body>
  <h1>Bienvenue sur mon site</h1>
  <p>Ceci est un paragraphe de texte sur mon site web.</p>
</body>
</html>
```

## 2.2 Sélecteurs de base (éléments, classes, ID)

Les sélecteurs CSS sont des outils essentiels pour appliquer des styles à des éléments HTML. Voici un aperçu des sélecteurs de base, y compris les sélecteurs d'éléments, de classes et d'ID.

### 1. Sélecteur d'élément

Le sélecteur d'élément cible tous les éléments d'un type spécifique dans le document HTML. Par exemple, pour styliser tous les éléments `<p>` (paragraphes), vous pouvez utiliser :

```
CSS

p {
    color: blue; /* Change la couleur du texte en bleu */
}
```

### 2. Sélecteur de classe

Les sélecteurs de classe ciblent les éléments qui ont un attribut class spécifique. Pour créer une classe, vous devez commencer le nom par un point (.) dans votre feuille de style CSS. Par exemple, pour styliser tous les éléments ayant la classe `highlight`, vous écrivez :

```
CSS

.highlight {
    background-color: yellow; /* Change le fond en jaune */
}
```

## 2.3 Couleur de fond (background-color)

La propriété CSS `background-color` est utilisée pour définir la couleur de fond d'un élément HTML. Cette propriété peut être appliquée à presque tous les éléments, y compris les `<div>`, `<p>`, `<h1>`, et même au `<body>` d'une page. Voici un guide détaillé sur son utilisation.

### CSS

```
élément {  
    background-color: couleur;  
}
```

### Types de couleurs

Vous pouvez spécifier la couleur de plusieurs manières :

1. **Noms de couleurs** : Utilisez des noms de couleurs prédéfinis en CSS (comme ``red``, ``blue``, ``green``, etc.).

#### CSS

```
p {  
    background-color: lightblue; /* Couleur de fond bleu clair */  
}
```

2. **Valeurs hexadécimales** : Utilisez des codes hexadécimaux pour définir des couleurs.

#### CSS

```
div {  
    background-color: #ffcc00; /* Couleur jaune */  
}
```

3. **Valeurs RGB** : Utilisez la notation RGB pour définir des couleurs en termes de rouge, vert et bleu.

#### CSS

```
h1 {  
    background-color: rgb(255, 0, 0); /* Couleur rouge */  
}
```

## 2.4 Police de caractère (font-family)

La propriété CSS `font-family` est essentielle pour définir la police de caractère utilisée pour afficher le texte dans un document HTML. Elle permet de spécifier une liste de polices, avec des options de repli au cas où la première police ne serait pas disponible sur le système de l'utilisateur. Voici un aperçu détaillé de son utilisation.

### Définition et Utilisation

#### Syntaxe

La syntaxe de base pour utiliser `font-family` est la suivante :

CSS

```
élément {  
    font-family: "Nom de la police", famille-générique;  
}
```

- ``Nom de la police`` : Le nom spécifique d'une police. Si le nom contient des espaces, il doit être entouré de guillemets.
- ``famille-générique`` : Un nom générique qui décrit le type de police, utilisé comme dernier recours.

### Types de familles

Il existe deux types principaux de familles dans `font-family` :

1. **Noms de polices spécifiques** : Par exemple, `"Arial"`, `"Times New Roman"`, `"Courier New"`.
2. **Familles génériques** : Ces catégories incluent :
  - ``serif`` : Polices avec des empattements (ex. : Times New Roman).
  - ``sans-serif`` : Polices sans empattements (ex. : Arial).
  - ``monospace`` : Polices où chaque caractère occupe le même espace (ex. : Courier).
  - ``cursive`` : Polices imitant l'écriture manuscrite.
  - ``fantasy`` : Polices décoratives.

## 2.5 Alignement de texte (text-align)

La propriété CSS `text-align` est utilisée pour contrôler l'alignement horizontal du texte à l'intérieur d'un élément de bloc. Voici un aperçu détaillé de son utilisation, de ses valeurs et d'exemples pratiques :

### Valeurs de `text-align`

La propriété `text-align` accepte plusieurs valeurs :

1. `left` : Aligne le texte à gauche. C'est la valeur par défaut si la direction du texte est de gauche à droite (LTR).

```
css
p {
  text-align: left; /* Texte aligné à gauche */
}
```

2. `right` : Aligne le texte à droite.

```
css
p {
  text-align: right; /* Texte aligné à droite */
}
```

3. `center` : Centre le texte dans l'élément.

```
css
p {
  text-align: center; /* Texte centré */
}
```

4. `justify` : Justifie le texte, ce qui signifie que les lignes sont étendues pour que chaque ligne ait la même largeur, alignant ainsi le texte à la fois à gauche et à droite.

```
css
p {
  text-align: justify; /* Texte justifié */
}
```

## 2.6 Centrage d'éléments (margin: auto)

La propriété CSS `margin: auto` est utilisée pour centrer des éléments horizontalement dans leur conteneur. Voici un aperçu de son utilisation :

### Syntaxe

La syntaxe de base pour centrer un élément avec `margin: auto` est la suivante :

CSS

```
élément {  
  width: largeur; /* Largeur spécifiée de l'élément */  
  margin: auto; /* Centre l'élément horizontalement */  
}
```



### Comment cela fonctionne ?

- **Éléments de bloc** : Pour que `margin: auto` fonctionne, l'élément doit être un élément de bloc (comme un `<div>`, `<p>`, ou `<h1>`) et avoir une largeur définie. Lorsque vous appliquez `margin: auto`, le navigateur répartit l'espace restant entre les marges gauche et droite, ce qui centre l'élément dans son conteneur.
- **Largeur spécifiée** : Si l'élément n'a pas de largeur définie, il prendra toute la largeur disponible, rendant le centrage impossible.

**Note** : le mot *largeur* sera à remplacer par une valeur en pixels (Ex : 155 px)



## 2.7 Div personnalisées

Une div est un élément HTML générique utilisé pour regrouper d'autres éléments. Elle agit comme un conteneur sans signification sémantique particulière.

Utilisation des classes

Les classes CSS permettent d'appliquer des styles spécifiques à ces div :

1. Définissez une classe dans votre CSS :

CSS

```
.ma-classe {  
  background-color: skyblue;  
  padding: 20px;  
}
```

Ici le nom de la classe est *ma-classe*.

2. Appliquez la classe à votre div :

Dans le fichier HTML :

```
<div class="ma-classe">Contenu personnalisé</div>
```