

Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

# PROGRAMMATION WEB

Pr. OMARI Kamal

ENSA KHOURIBGA

20 février 2025

Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

## Chapitre 2 : Exploration du Langage HTML

- 1 Introduction au langage HTML
- 2 Histoire et Évolution du HTML
- 3 Exploiter un environnement de développement
- 4 Définir les éléments basiques d'une page HTML
- 5 Créer des formulaires HTML

## Objectifs du cours

Dans ce chapitre, nous allons explorer le langage HTML, qui est essentiel pour la création de pages web. Il sera possible de se familiariser avec la structure d'un document HTML, ainsi que les éléments et balises fondamentaux qui le composent. À l'issue de ce chapitre, il sera possible de créer des pages web simples, d'insérer des images, d'établir des liens, et d'adopter de bonnes pratiques pour garantir l'accessibilité et la validation du code HTML.

Introduction au langage HTML

Histoire et Évolution du HTML

Exploiter un environnement de développement

Définir les éléments basiques d'une page HTML

Créer des formulaires HTML

Présentation générale du HTML

Importance du HTML dans le Développement Web

# Présentation générale du HTML

Le HTML (HyperText Markup Language) est le langage de balisage standard utilisé pour créer des pages web. Il structure le contenu à l'aide de balises, indiquant comment les éléments doivent être affichés dans un navigateur.

Introduction au langage HTML

Histoire et Évolution du HTML

Exploiter un environnement de développement

Définir les éléments basiques d'une page HTML

Créer des formulaires HTML

Présentation générale du HTML

Importance du HTML dans le Développement Web

# Importance du HTML dans le Développement Web

Le HTML est fondamental dans le développement web, car il constitue la base de toute page web. Sans HTML, il serait impossible de structurer et d'afficher des informations. Il sert également de fondement pour des technologies comme le CSS pour le style et JavaScript pour l'interactivité, garantissant ainsi que les sites sont fonctionnels, bien structurés et accessibles.

# Origines du HTML

Le HTML a été créé au début des années 1990 par Tim Berners-Lee, un ingénieur du CERN, pour faciliter le partage d'informations entre chercheurs. Bien que la première version ait été développée en 1991, elle n'a été officiellement publiée qu'en 1993 sous la forme de la spécification HTML 1.0 par le World Wide Web Consortium (W3C).

Ce langage était principalement destiné à structurer des documents textuels et à intégrer des liens hypertextes, permettant ainsi de naviguer d'un document à un autre. Le concept d'hypertexte a été au cœur de cette invention, ouvrant la voie à ce que nous connaissons aujourd'hui comme le World Wide Web.

# Évolution des Versions de HTML

Depuis ses débuts, HTML a connu plusieurs évolutions significatives :

- HTML 4.01 (1999) : Cette version a introduit de nombreuses améliorations, notamment la prise en charge des feuilles de style CSS, permettant ainsi une séparation entre le contenu et la présentation. HTML 4.01 a également renforcé l'accessibilité, permettant aux développeurs de rendre leurs sites plus utilisables pour les personnes handicapées.

# Évolution des Versions de HTML

- XHTML (2000) : Une reformulation de HTML 4.01, XHTML (Extensible Hypertext Markup Language) a imposé des règles plus strictes concernant la syntaxe, combinant les caractéristiques de HTML et XML (Extensible Markup Language). Bien qu'elle ait apporté des avantages en matière de rigueur syntaxique, elle n'a pas gagné une adoption massive.



# Évolution des Versions de HTML

- HTML5 (2014) : La version la plus récente et la plus utilisée du HTML a marqué un tournant dans le développement web. HTML5 a introduit de nouvelles balises et fonctionnalités, telles que les balises `<video>` et `<audio>` pour intégrer des médias directement dans les pages web, ainsi que des éléments comme `<section>`, `<article>`, et `<header>`, qui permettent une meilleure structuration du contenu. Cette version a également mis l'accent sur la compatibilité avec les appareils mobiles et l'accessibilité.

## Le rôle du W3C

Le W3C (World Wide Web Consortium) est un organisme international à but non lucratif.

La mission du W3C est de définir des standards techniques liés au web afin d'assurer l'accessibilité, l'interopérabilité et la pérennité des documents Web. En l'absence de ces standards, chaque navigateur aurait ses propres règles d'interprétation des données Web. Ainsi, le rôle du W3C consiste à standardiser les technologies du web et à harmoniser les pratiques entre les navigateurs et les développeurs, en établissant un ensemble de règles à respecter par tous.

# Propriétés des navigateurs

## Le rôle d'un navigateur

Récupérer le code (généralement écrit en HTML, PHP, JavaScript ou d'autres langages de programmation Web), à partir du serveur, puis l'interpréter et le restituer sous forme de page Web graphique pour l'affichage.

**Interpréter** l'adresse d'un site web (URL) et gérer les requêtes client/Serveur.

Faciliter le développement et le test des pages Web.

# Exemple

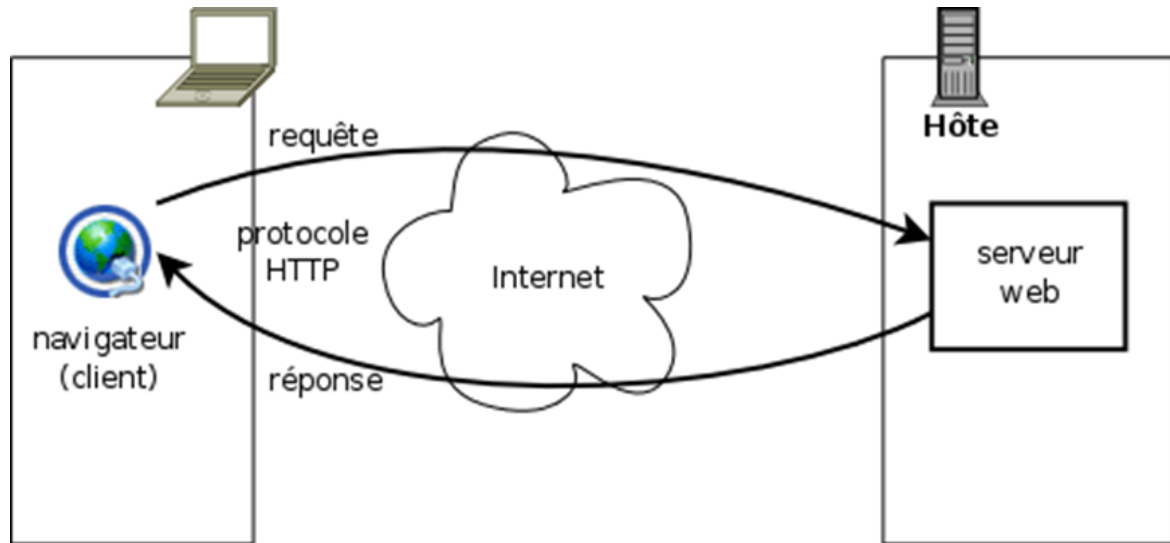


Figure 1 – Communication client-serveur web en HTTP

# Explication

## Le fonctionnement d'un navigateur

En utilisant le navigateur Web, l'utilisateur envoie une requête dite "Requête HTTP" au serveur qui héberge le site web.

**Le serveur Web** reçoit la requête de l'utilisateur qui contient l'adresse IP du serveur, le port sur lequel tourne le serveur Web, la ressource demandée et la méthode avec laquelle on exploite cette ressource.

**Le navigateur** reçoit la page web demandée sous forme de code appelé : "code HTML" et l'interprète pour afficher des objets visuels.

# Le logiciel Visual studio Code

Visual Studio Code est un logiciel gratuit qui permet l'édition, la correction et le débogage du code source dans plusieurs langages informatiques : Visual Basic, JavaScript, XML, Python, HTML, CSS, ....

VS code offre :

- Une présentation sophistiquée du code.
- Une auto-complétion du code source.
- Un ensemble d'extensions permettant de simplifier la programmation.
- La détection du langage de programmation par l'extension du fichier.

Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Utiliser Visual Studio Code comme éditeur de code  
Installation des extensions de VS Code

## Environnement de travail de VS Code

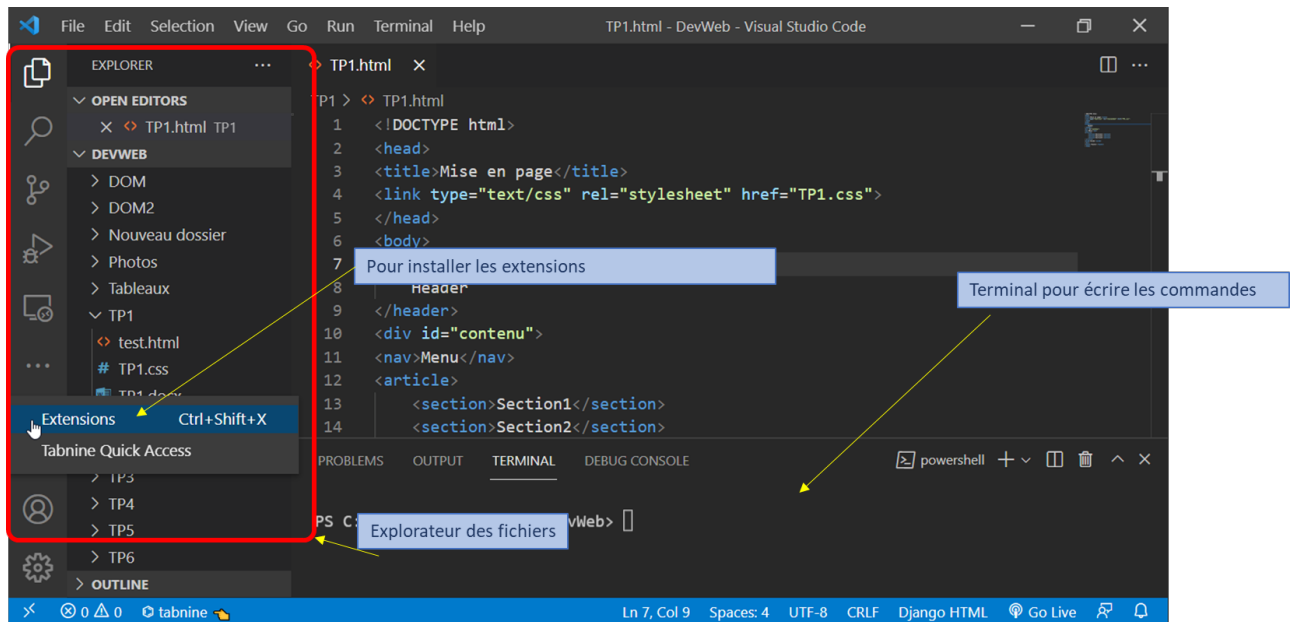


Figure 2 – Enter Caption

Introduction au langage HTML

Histoire et Évolution du HTML

**Exploiter un environnement de développement**

Définir les éléments basiques d'une page HTML

Créer des formulaires HTML

Utiliser Visual Studio Code comme éditeur de code  
Installation des extensions de VS Code

## Les extensions dans VS Code

Les extensions de Visual Studio Code permettent d'accélérer le développement et favorisent le partage entre les membres de l'équipe de développement. Le modèle d'extensibilité riche de VS Code permet aux «auteurs d'extensions» ainsi qu'aux utilisateurs d'intégrer facilement à l'environnement de développement de VS code.

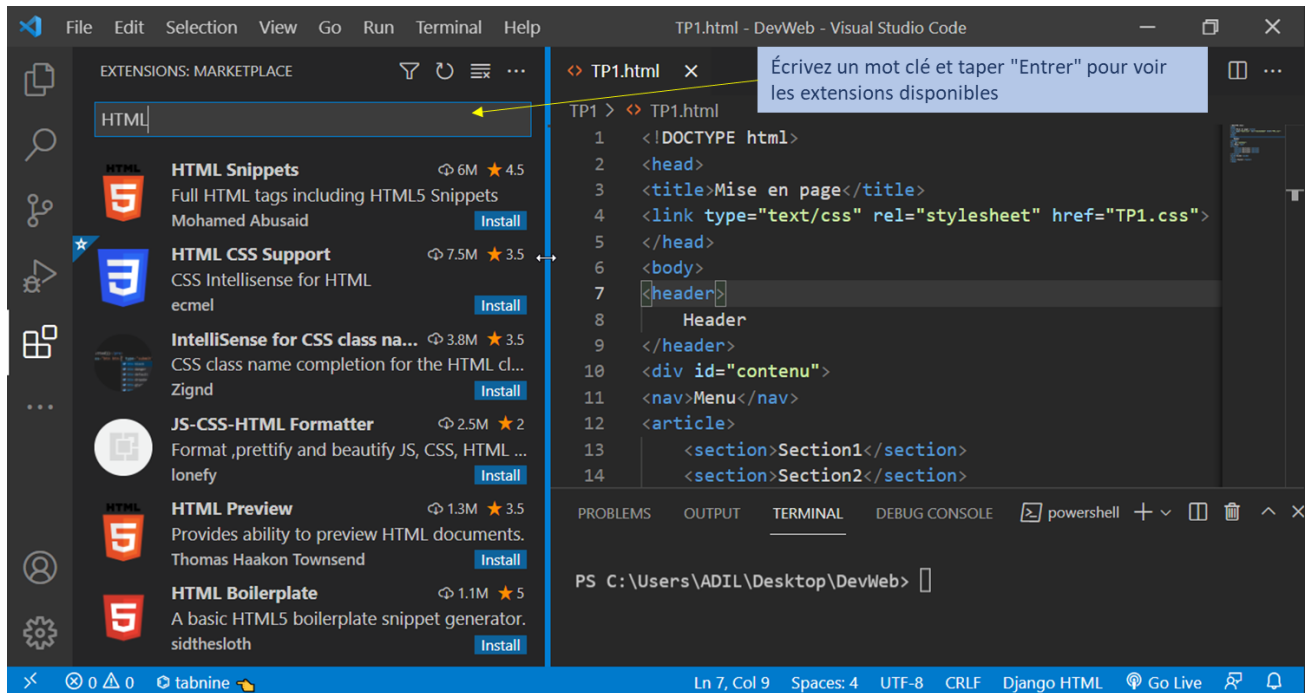


Introduction au langage HTML  
Histoire et Évolution du HTML

Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Utiliser Visual Studio Code comme éditeur de code  
Installation des extensions de VS Code

# Installation des extensions dans VS Code



## Les normes W3C

**W3C** est chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XML, RDF, CSS, PNG, SVG, ...  
Parmi les normes W3C concernant HTML on cite :

- Les balises `<html>`, `<head>`, `<title>`, et `<body>` sont obligatoires
- Les éléments doivent être correctement imbriqués
- Les éléments doivent toujours être fermés
- Les éléments HTML doivent être en minuscules
- Les documents HTML doivent avoir un élément racine
- Les noms d'attributs doivent être en minuscules
- Les valeurs des attributs doivent être indiqués

# Squelette d'une page HTML

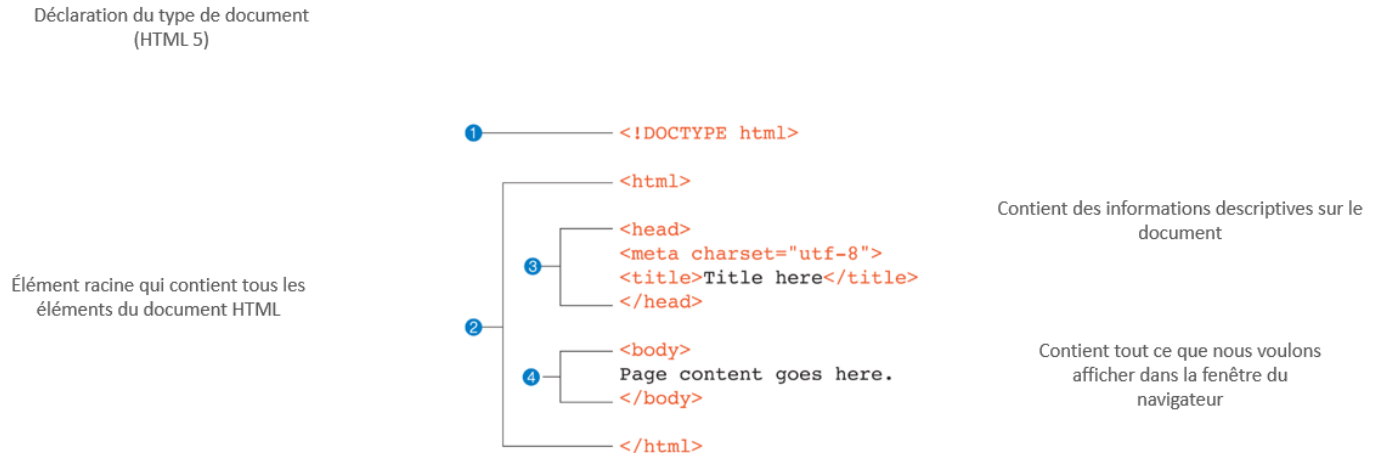


Figure 4 – Squelette d'une page HTML

# Notion de balise

- Une balise est un élément de texte encadrée par les caractères inférieur (<) et supérieur (>)
- Une balise correspond à un élément du code HTML d'une page web (un objet, une mise en forme, ..)
- Une balise a généralement la forme suivante :

# La forme d'une balise HTML

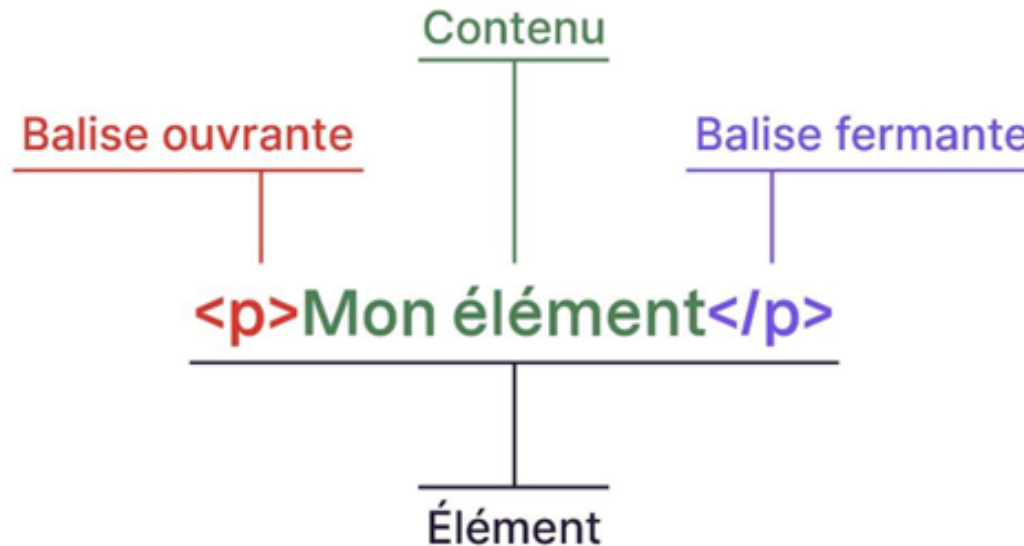


Figure 5 – Forme d'une balise HTML

## Notion d'encapsulation

En HTML, l'encapsulation est le concept d'empaqueter ou de regrouper des éléments à l'intérieur d'autres éléments pour structurer le contenu de manière logique et ordonnée. Contrairement à la programmation orientée objet (où l'encapsulation est utilisée pour restreindre l'accès aux données internes d'un objet), en HTML, l'encapsulation sert principalement à organiser et hiérarchiser les éléments pour une meilleure lisibilité et pour faciliter la mise en page.

# Notion d'encapsulation

```
<div>
  <h1>titre</h1>
  <h2>un sous titre</h2>
  <p>un paragraphe avec
  <em>des mots importants</em>
  pour terminer avec une image
  
  </p>
</div>
```

Figure 6 – Encapsulation des balises HTML

# Les types d'éléments HTML

Les éléments HTML peuvent être classés en plusieurs types, selon leur comportement et leur rôle dans la structure d'une page web. Voici les principaux types d'éléments HTML :

## 1-Éléments en bloc (Block-level elements)

- **Caractéristiques** : Les éléments en bloc occupent tout l'espace disponible en largeur et commencent sur une nouvelle ligne. Ils créent une sorte de "bloc" qui sépare leur contenu du reste de la page.
- **Exemples** : `<div>`, `<p>`, `<h1>` à `<h6>`, `<section>`, `<article>`, `<header>`, `<footer>`, `<nav>`, `<ul>`, `<ol>`, `<li>`, `<table>`.
- **Utilisation** : Ils sont utilisés pour structurer des sections de contenu plus importantes, telles que des paragraphes, des titres, des sections ou des listes.



Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Les normes W3C  
Squelette d'une page HTML  
Les couleurs en HTML  
Les listes en HTML  
Les liens en HTML  
Les balises multimédia (images, audio, vidéo)  
Les tableaux avec du HTML  
Les balises de structuration

# Les types d'éléments HTML

## Exemple.html

```
<div>  
<h1>Titre</h1>  
<p>Ceci est un paragraphe qui prend toute la largeur  
disponible.</p>  
</div>
```

# Les types d'éléments HTML

## 2-Éléments en ligne (Inline elements)

- **Caractéristiques** : Contrairement aux éléments en bloc, les éléments en ligne n'interrompent pas le flux du document. Ils ne commencent pas sur une nouvelle ligne et n'occupent que la largeur nécessaire à leur contenu.
- **Exemples** : `<span>`, `<a>`, `<strong>`, `<em>`, `<img>`, `<input>`, `<label>`, `<abbr>`.
- **Utilisation** : Ils sont souvent utilisés pour styliser ou intégrer des parties du texte ou des petites sections de contenu sans interrompre le flux global du document.

Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Les normes W3C  
Squelette d'une page HTML  
Les couleurs en HTML  
Les listes en HTML  
Les liens en HTML  
Les balises multimédia (images, audio, vidéo)  
Les tableaux avec du HTML  
Les balises de structuration

# Les types d'éléments HTML

## Exemple.html

```
<p>Voici un texte avec un <strong>élément en  
ligne</strong> au milieu.</p>
```

# Les attributs HTML

Les attributs HTML sont des propriétés ajoutées aux balises HTML pour fournir des informations supplémentaires sur les éléments. Ils modifient le comportement ou l'apparence de l'élément auquel ils sont appliqués.

A blue rounded rectangle containing the text "<balise attribut='valeur'>". The word "balise" is in yellow, "attribut=" is in orange, and "valeur" is in red.

**<balise attribut="valeur">**

Figure 7 – Les attributs HTML

# Les balises de base générale d'une page HTML

- `<!DOCTYPE html>`

**Description :** Cette déclaration informe le navigateur sur le type de document qu'il va traiter. Ici, elle indique que le document est un document HTML5. Cela permet au navigateur de rendre correctement la page en suivant les normes HTML.

- `<html>`

**Description :** L'élément `<html>` est l'élément racine de tout document HTML. Il contient tous les autres éléments HTML, tels que `<head>` et `<body>`. C'est le premier élément que le navigateur rencontre.

- `<head>`

**Description :** La balise `<head>` contient des métadonnées (informations sur les données) qui ne sont pas directement affichées sur la page. Ces métadonnées peuvent inclure des liens vers des fichiers CSS, des scripts JavaScript, des balises `<meta>`, et le titre de la page. Elles sont essentielles pour le référencement et le bon fonctionnement de la page.

# Les balises de base générale d'une page HTML

## ● `<meta>`

**Description :** Les balises `<meta>` fournissent des informations supplémentaires sur le document, telles que l'encodage des caractères (`<meta charset="UTF-8">`), la description de la page pour les moteurs de recherche (`<meta name="description" content="Description de la page">`), les mots-clés (`<meta name="keywords" content="mot-clé1, mot-clé2">`), et d'autres instructions qui aident à la gestion de la page. Ces balises sont invisibles sur la page mais cruciales pour le SEO (optimisation pour les moteurs de recherche).

## ● `<title>`

**Description :** La balise `<title>` définit le titre de la page qui apparaît dans l'onglet du navigateur et est également utilisée par les moteurs de recherche pour référencer la page. C'est un élément clé pour l'identification de la page par les utilisateurs.

# Les balises de base générale d'une page HTML

- **<body>**

**Description :** L'élément **<body>** contient tout le contenu visible de la page web, y compris les textes, images, vidéos, liens, et autres éléments interactifs. C'est la partie de la page avec laquelle les utilisateurs interagissent et qu'ils voient directement.

- **<h1>**

**Description :** La balise **<h1>** est utilisée pour définir un titre principal ou un en-tête de premier niveau sur la page. C'est généralement le titre le plus important d'une section ou d'un document. Elle aide à structurer le contenu et à indiquer aux moteurs de recherche de quoi parle la page.

- **<p>**

**Description :** La balise **<p>** est utilisée pour créer un paragraphe de texte. Elle permet d'organiser le texte en blocs lisibles et contribue à la mise en forme du contenu. Chaque balise **<p>** commence un nouveau paragraphe, ce qui facilite la lecture et la compréhension.

*Cette structure de balises est fondamentale pour créer des pages web et comprendre comment elles sont organisées.*

# Les couleurs en HTML

En HTML, on peut spécifier les couleurs de trois manières différentes :

- **En utilisant la valeur Hex :** HEX est un système de numérotation hexadécimale des couleurs (basé sur les chiffres décimaux de 0 à 9 et les lettres A à F). La conception Web utilise le code de couleur hexadécimal sous la forme #RRGGBB, où RR représente le rouge, GG le vert et BB le bleu. Chaque composant de couleur est compris entre 00 et FF.



# Les couleurs en HTML

- **En utilisant les noms des couleurs :** Les noms des couleurs doivent être spécifiés en anglais (par exemple, blue, red, etc.).
- **En utilisant la valeur RGB (Red, Green, Blue) :** Le code couleur RGB est représenté sous la forme de trois valeurs numériques comprises entre 0 et 255. Ces trois valeurs représentent respectivement le dosage du rouge, du vert et du bleu.

**Exemples de sites pour obtenir les codes des couleurs :**

- [https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)
- <https://htmlcolorcodes.com/>

## Les couleurs en HTML (exemples)

### Exemple.html

```
<p style="color:red;">Ceci est un texte rouge</p>  
<p style="color:#FF0000;">Ceci est un texte rouge</p>  
<p style="color:rgb(255,0,0);">Ceci est un texte  
rouge</p>
```

Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Les normes W3C  
Squelette d'une page HTML  
**Les couleurs en HTML**  
Les listes en HTML  
Les liens en HTML  
Les balises multimédia (images, audio, vidéo)  
Les tableaux avec du HTML  
Les balises de structuration

## En résumé

En HTML, la couleur peut être spécifiée de plusieurs façons selon le contexte et les besoins : noms de couleurs prédéfinis, code hexadécimal ou RGB. Il est essentiel de choisir le format qui convient le mieux à vos besoins et de veiller à l'accessibilité.

# Les listes en HTML

En HTML, les listes permettent de structurer et de présenter des informations sous forme d'éléments regroupés. Il existe trois types principaux de listes en HTML :

## 1. Liste ordonnée (<ol>)

- Une liste ordonnée affiche les éléments avec un ordre numéroté ou avec des lettres romaines ou alphabétiques. Chaque élément de la liste est défini par une balise <li> (list item).
- Attributs supplémentaires : Type (pour les listes ordonnées) : L'attribut type permet de spécifier le type de numérotation (nombres, lettres, etc.). *exemple : type="A".*

Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Les normes W3C  
Squelette d'une page HTML  
Les couleurs en HTML  
**Les listes en HTML**  
Les liens en HTML  
Les balises multimédia (images, audio, vidéo)  
Les tableaux avec du HTML  
Les balises de structuration

# Exemple

## Exemple.html

```
<ol>  
<li>Premier élément</li>  
<li>Deuxième élément</li>  
<li>Troisième élément</li>  
</ol>
```

# Les listes en HTML

## 2. Liste non ordonnée (<ul>)

- Une liste non ordonnée affiche des éléments sans numérotation, mais avec des puces (ou un autre style de marquage). Les éléments sont également définis avec la balise <li>.

### Exemple.html

```
<ul>  
<li>Premier élément</li>  
<li>Deuxième élément</li>  
<li>Troisième élément</li>  
</ul>
```

*NB : On peut ajouter l'attribut type="square" dans une liste non ordonnée (<ul>) pour changer l'apparence des puces en carrés au lieu de puces classiques.*

# Frame Title

## 3. Liste de définition (<dl>)

- Les listes de définition sont utilisées pour des paires de termes et de descriptions. Elles sont constituées de la balise <dl> (definition list), <dt> pour le terme (definition term), et <dd> pour la définition (definition description).

### Exemple.html

```
<dl>
<dt>HTML</dt>
<dd>Langage de balisage pour créer des pages web.</dd>
<dt>CSS</dt>
<dd>Langage utilisé pour styliser les pages web.</dd>
</dl>
```

Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Les normes W3C  
Squelette d'une page HTML  
Les couleurs en HTML  
**Les listes en HTML**  
Les liens en HTML  
Les balises multimédia (images, audio, vidéo)  
Les tableaux avec du HTML  
Les balises de structuration

## En résumé

Les listes en HTML sont un moyen efficace d'organiser des données de manière structurée, que ce soit avec un ordre numérique ou sous forme de définitions. Elles améliorent la lisibilité et la présentation des informations sur une page web.



# Les liens en HTML

Les liens en HTML permettent de naviguer d'une page web à une autre ou de rediriger vers des ressources externes. Ils sont créés à l'aide de la balise `<a>`, qui signifie "ancree". L'attribut principal pour définir un lien est `href`, qui indique l'URL vers laquelle le lien pointe.

## Syntaxe de base :

```
<a href="https://www.example.com">Cliquez ici</a>
```

Attributs courants des liens :

- **href** : Spécifie l'URL de destination.
- **target** : Détermine où ouvrir le lien. Par exemple, `target="_blank"` ouvre le lien dans un nouvel onglet.
- **title** : Fournit une information supplémentaire sur le lien, souvent affichée au survol du curseur.
- **rel** : Définit la relation entre la page actuelle et la page liée (par exemple, `rel="noopener"` pour des raisons de sécurité).

# Balises pour les Images

Balise `<img>` : Cette balise est utilisée pour insérer des images dans une page web. Elle ne nécessite pas de balise de fermeture et permet d'afficher visuellement des contenus.

## Syntaxe de base :

```

```

## Attributs principaux :

- **src** : spécifie le chemin vers le fichier image. Il peut s'agir d'un chemin relatif (par rapport à la page web) ou d'une URL absolue.
- **alt** : fournit une description alternative de l'image. Cet attribut est essentiel pour l'accessibilité, car il permet aux utilisateurs de comprendre le contenu de l'image lorsqu'elle ne peut pas être affichée.
- **width et height** : définissent la taille de l'image en pixels. Cela permet de contrôler la mise en page de la page web.

# Balises pour l'Audio

Balise `<audio>` : Cette balise est utilisée pour intégrer des fichiers audio dans une page web. Elle permet aux utilisateurs de lire, mettre en pause et contrôler le volume de l'audio directement à partir de la page.

## Syntaxe de base :

```
<audio controls> <source src="chemin/vers/audio.mp3" type="audio/mpeg"> Votre navigateur ne prend pas en charge l'élément audio. </audio>
```

Attributs principaux :

- **controls** : cet attribut ajoute des contrôles natifs au lecteur audio, permettant aux utilisateurs de lire, mettre en pause et régler le volume.
- **<source>** : cette sous-balise permet d'ajouter plusieurs formats audio, ce qui améliore la compatibilité entre les différents navigateurs. Il est courant d'inclure des fichiers en différents formats (comme MP3, OGG) pour garantir une lecture fluide.

## Balises pour la Vidéo

Balise `<video>` : Cette balise est utilisée pour intégrer des fichiers vidéo. Elle offre des fonctionnalités similaires à la balise audio, permettant d'inclure des vidéos interactives sur une page web.

### Syntaxe de base :

```
<video width="640" height="360" controls> <source  
src="chemin/vers/video.mp4" type="video/mp4"> Votre  
navigateur ne prend pas en charge l'élément vidéo.  
</video>
```

Attributs principaux :

- **width et height** : ces attributs définissent la taille de la vidéo en pixels, permettant un meilleur contrôle de l'affichage et de l'intégration visuelle dans la page.
- **controls** : en ajoutant cet attribut, des contrôles de lecture sont affichés, permettant aux utilisateurs de gérer la lecture de la vidéo (lecture, pause, volume, etc.).
- **<source>** : comme pour l'audio, cette sous-balise permet d'inclure plusieurs formats vidéo pour maximiser la compatibilité avec les navigateurs.

## En résumé

Les balises multimédia en HTML permettent d'enrichir les pages web avec des éléments visuels et sonores, offrant une expérience utilisateur plus interactive :

- **Images** : La balise `<img>` permet d'intégrer des images, essentielles pour illustrer des contenus.
- **Audio** : La balise `<audio>` intègre des fichiers audio avec des contrôles pour une écoute interactive.
- **Vidéo** : La balise `<video>` permet d'inclure des vidéos, offrant des fonctionnalités de lecture directement sur la page.

Ces balises sont essentielles pour créer des sites web modernes et interactifs, engageant les utilisateurs avec divers types de contenu multimédia.

# Les tableaux avec du HTML

Les tableaux en HTML sont des structures essentielles permettant d'organiser et de présenter des données de manière tabulaire. Grâce à la balise `<table>`, les développeurs peuvent créer des tableaux composés de lignes et de colonnes, facilitant la visualisation et la comparaison d'informations. Les tableaux sont souvent utilisés pour afficher des données comme des horaires, des statistiques, des catalogues de produits, ou tout autre type d'information nécessitant une présentation claire et ordonnée.

## Éléments de base d'un tableau

- **<table>** : Définit le début d'un tableau.
- **<tr>** (table row) : Définit une ligne du tableau. Chaque ligne peut contenir plusieurs cellules.
- **<td>** (table data) : Définit une cellule de données dans une ligne. C'est ici que le contenu du tableau est affiché.
- **<th>** (table header) : Définit une cellule d'en-tête, généralement utilisée pour le titre des colonnes. Les cellules d'en-tête sont souvent affichées en gras et centrées par défaut.

Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Les normes W3C  
Squelette d'une page HTML  
Les couleurs en HTML  
Les listes en HTML  
Les liens en HTML  
Les balises multimédia (images, audio, vidéo)  
**Les tableaux avec du HTML**  
Les balises de structuration

# Exemple de tableau en HTML

## Exemple.html

```
<table border="1">
<caption>Exemple de tableau de produits</caption>
<tr>
<th>Produit</th>
<th>Prix</th>
<th>Quantité</th>
</tr> <tr>
<td>Chaussures</td>
<td>500 MAD</td>
<td>10</td>
</tr> <tr>
<td>T-shirt</td>
<td>200 MAD</td>
<td>15</td>
</tr> <tr>
<td>Pantalon</td>
<td>400 MAD</td>
<td>8</td>
</tr>
</table>
```



## Espacement des cellules

- L'attribut "cellspacing" permet de définir l'espacement entre les cellules du tableau.
- l'attribut "cellpadding" permet de définir l'espacement entre le contenu de la cellule et ses bords.

### Exemple.html

```
<table border="1" cellspacing=10 cellpadding=5>
```

## Division des lignes et des colonnes (fusion des cellules)

Les attributs **colspan** et **rowspan** indiquent sur combien de colonnes et de lignes s'étend la cellule.

***Exemple :** On souhaite que la première cellule du tableau prenne toute la largeur de la ligne. Elle doit donc déborder sur 3 cellules horizontales.*

### Exemple de tableau en HTML

```
<table width="60%" border="1">
<tr>
<td colspan="3">cellule 1</td>
</tr>
<tr>
<td width="33%">cellule 1</td>
<td width="33%">cel 2</td>
<td width="34%">3</td>
</tr>
</table>
```

# Les balises de structuration

Les balises de structuration en HTML sont utilisées pour organiser et donner une structure logique au contenu d'une page web. Elles permettent de définir les différentes sections et éléments d'une page afin de la rendre plus compréhensible, aussi bien pour les navigateurs que pour les utilisateurs. Elles se divisent en deux parties : les balises sémantiques et non sémantiques. Voici quelques balises essentielles de structuration :

# Les balises sémantiques

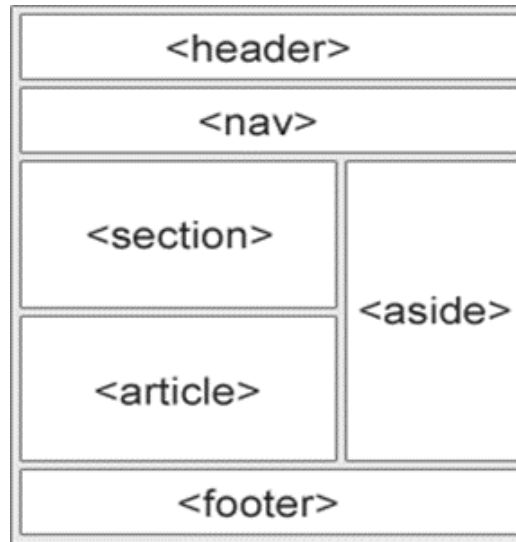


Figure 8 – Les éléments sémantiques

# Balises Sémantiques

Ces balises décrivent clairement leur signification de manière compréhensible aussi bien pour les humains que pour les machines.

**<header>** : Définit un en-tête de page ou de section, souvent utilisé pour inclure des titres, logos ou éléments de navigation.

## Exemple

```
<header>  
<h1>Bienvenue sur Mon Site</h1>  
</header>
```

# Balises Sémantiques

**<nav>** : Définit une section de liens de navigation.

## Exemple

```
<nav>
<ul>
<li><a href="home">Accueil</a></li>
<li><a href="about">À propos</a></li>
</ul>
</nav>
```

# Balises Sémantiques

**<article>** :Définit une portion de contenu autonome, comme un article de blog.

## Exemple

```
<article>  
<h2>Titre de l'article</h2>  
<p>Contenu de l'article...</p>  
</article>
```

# Balises Sémantiques

**<section>** : Définit une section du document, généralement accompagnée d'un titre.

## Exemple

```
<section>  
<h2>Titre de la section</h2>  
<p>Détails de cette section...</p>  
</section>
```



# Balises Sémantiques

**<footer>** : Définit le pied de page d'un document ou d'une section, contenant souvent des informations comme les droits d'auteur ou les contacts.

## Exemple

```
<footer>  
<p>© 2024 Mon Site. Tous droits réservés.</p>  
</footer>
```

# Balises Sémantiques

**<aside>** : Définit un contenu en dehors du contenu principal, comme une barre latérale.

## Exemple

```
<aside>
<h3>Liens utiles</h3>
<ul>
<li><a href="#">Lien 1</a></li>
</ul>
</aside>
```

## Balises Non Sémantiques

Ces balises ne donnent pas d'information spécifique sur la structure ou la signification du contenu.

**<div>** : Un conteneur générique pour regrouper des éléments. Utilisé souvent avec CSS pour la mise en page.

### Exemple

```
<div class="content">  
<p>Contenu...</p>  
</div>
```

# Balises Non Sémantiques

**<span>** : Un conteneur en ligne générique pour du texte ou d'autres éléments en ligne. Souvent utilisé pour appliquer des styles.

## Exemple

```
<span style="color:red;">Texte en rouge</span>
```

## Balise <form>

Un formulaire est une section d'un document web qui permet à un utilisateur de fournir des informations via des contrôles interactifs

L'élément HTML <form> permet de déclarer un formulaire

L'attribut "action" spécifie la ressource qui va recevoir les données du formulaire pour les traiter (un script).

L'attribut "method" spécifie la méthode d'envoi des données. On distingue deux méthodes :

- GET : permet d'envoyer les données sous forme de chaîne de paramètres à travers l'URL. Les données sont visibles dans la barre d'adresse.
- POST : permet d'envoyer les données (invisibles) à travers une requête HTTP.

Introduction au langage HTML  
Histoire et Évolution du HTML

Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Balise <form>

Champs de saisie (Input, TextArea)

Balises de choix (RadioButton, CheckBox, Select)

Boutons

## Exemple

### Exemple

```
<form action="authentification.php" method="post">  
<!-- Ici on insère les éléments du formulaire-->  
</form>
```

## Champs de saisie (Input, TextArea)

L'élément <input> Permet de coder différents types d'entrée pour un formulaire, selon la valeur de l'attribut type.

### Exemple

```
<form action="authentification.php" method="post">  
<input type="text" id="firstname" name="firstname">  
</form>
```

# Types d'Input HTML

```
<input type="button">  
<input type="checkbox">  
<input type="color">  
<input type="date">  
<input type="datetime-local">  
<input type="email">  
<input type="file">  
<input type="hidden">
```

```
<input type="image">  
<input type="month">  
<input type="number">  
<input type="password">  
<input type="radio">  
<input type="range">  
<input type="reset">  
<input type="search">
```

```
<input type="submit">  
<input type="tel">  
<input type="text">  
<input type="time">  
<input type="url">  
<input type="week">
```

Figure 9 – Types d'Input HTML



# Champ de saisie <input>

## Type texte

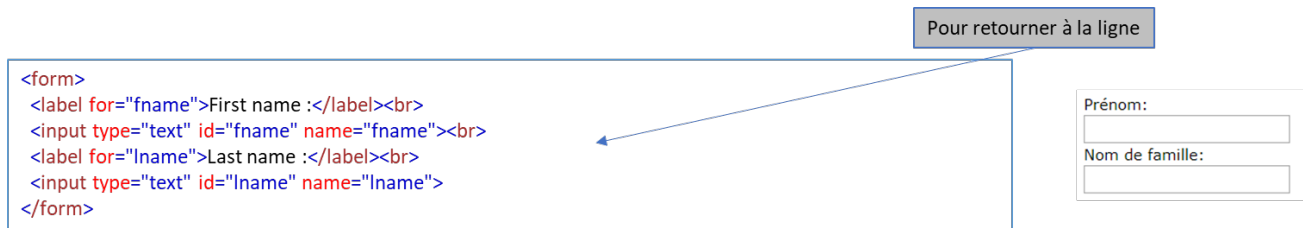
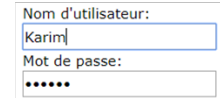


Figure 10 – Champ de saisie <input>

# Type password

```
<form>  
<label for="username">Username :</label><br>  
<input type="text" id="username" name="username"><br>  
<label for="pwd">Password :</label><br>  
<input type="password" id="pwd" name="pwd">  
</form>
```



Nom d'utilisateur:  
Karim

Mot de passe:  
\*\*\*\*\*

Figure 11 – Type password

## Champ de saisie <textarea>

L'éléments <textarea> permet d'afficher et saisir un texte en plusieurs ligne. Exemple :

```
<textarea>L'Office de la formation professionnelle et de la promotion du travail (OFPPT) est un opérateur public en formation professionnelle.  
</textarea>
```

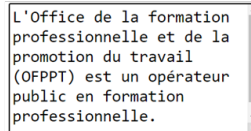


Figure 12 – Champ de saisie <textarea>

## Type radio

Les boutons radio permettent à un utilisateur d'en sélectionner SEULEMENT UN parmi un nombre limité de choix :

```
<form>  
<input type="radio" id="male" name="sexe" value="Masculin">  
<label for="male">Masculin</label><br>  
<input type="radio" id="female" name="sexe" value="Féminin">  
<label for="female">Féminin</label><br>  
</form>
```

☐ Masculin  
☐ Féminin

Figure 13 – Type radio

# Type number

```
<form>  
<label for="quantité">Quantité :</label>  
<input type="number" id="quantity" name="quantity" min="0" max="100" step="10" value="30">  
</form>
```

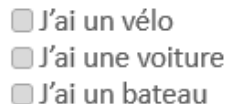
Quantité:

Figure 14 – Type number

## Type checkbox

<input type="checkbox"> définit une case à cocher . Les cases à cocher permettent à un utilisateur de sélectionner ZERO ou PLUS d'options d'un nombre limité de choix.

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> J'ai un vélo</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2">J'ai une voiture</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> J'ai un bateau</label>
</form>
```



☐ J'ai un vélo  
☐ J'ai une voiture  
☐ J'ai un bateau

## Type select

L'élément HTML <select> représente un contrôle qui fournit un menu d'options :

- L'attribut id associe un identifiant unique à la liste
- L'attribut name représente le nom du point de données associé lors de la soumission au serveur.
- L'attribut multiple spécifie si plusieurs options peuvent être sélectionnées, et size spécifie combien d'options doivent être affichées à la fois.
- L'élément <option> définit une option de la liste.
- Chaque élément <option> doit avoir un attribut value contenant la valeur de données à soumettre au serveur lorsque cette option est sélectionnée. L'attribut selected sur un élément <option> le rend sélectionné par défaut lors du premier chargement de la page.

## Type select

```
<label for="langage">Sélectionner un langage de programmation:</label>

<select name="lg" id="langage">
  <option value="">--Choisir --</option>
  <option value="Python">Python</option>
  <option value="CSharp">C#</option>
  <option value="java">java</option>
</select>
```

Figure 16 – Type select



## Type button

<input type="button"> définit un bouton :

```
<input type="button" onclick="alert('Hello World!)" value="Click Me!">
```

### Type color

```
<form>  
<label for="favcolor">Select your favorite color :</label>  
<input type="color" id="favcolor" name="favcolor">  
</form>
```

### Type date

```
<form>  
<label for="birthday">Birthday :</label>  
<input type="date" id="birthday" name="birthday">  
</form>
```

Figure 17 – Type button

## Type datetime-local

Le type datetime-local dans un élément <input> HTML permet de créer un champ de saisie où l'utilisateur peut entrer à la fois une date et une heure, sans fuseau horaire. Cela fournit une interface conviviale pour sélectionner ces informations via une interface de sélection de date et d'heure (par exemple, un calendrier et un sélecteur d'heure).

## Type datetime-local

Dans cet exemple, l'utilisateur pourra choisir à la fois la date et l'heure d'un rendez-vous, par exemple.

Voici un exemple d'utilisation du type datetime-local

```
<label for="appointment">Choisir une date et heure :</label>  
<input type="datetime-local" id="appointment" name="appointment">
```

## Type datetime-local

Attributs supplémentaires possibles :

- min et max : pour définir des limites de date et d'heure.
- step : pour spécifier l'incrément (en secondes) entre les valeurs disponibles.

### Exemple avec des attributs supplémentaires

```
<input type="datetime-local" id="meeting" name="meeting"  
min="2024-10-01T00:00" max="2024-12-31T23:59">
```

Introduction au langage HTML

Histoire et Évolution du HTML

Exploiter un environnement de développement

Définir les éléments basiques d'une page HTML

Créer des formulaires HTML

Balise <form>

Champs de saisie (Input, TextArea)

Balises de choix (RadioButton, CheckBox, Select)

Boutons

## Type email

Le type email dans un élément <input> HTML est utilisé pour créer un champ de saisie où l'utilisateur peut entrer une adresse e-mail. Ce type de champ vérifie automatiquement que l'entrée correspond au format d'une adresse e-mail valide (par exemple, contenant un symbole @ et un domaine valide).

## Type email

Voici un exemple d'utilisation du type email

```
<form action="/submit_form" method="POST">  
<label for="email">Adresse e-mail :</label>  
<input type="email" id="email" name="email">  
<input type="submit" value="Envoyer">  
</form>
```

## Type email

Explication des attributs :

- `type = "email"` : indique que le champ est pour une adresse e-mail.
- `id = "email"` et `name = "email"` : identifiant et nom associés au champ, nécessaires pour lier l'étiquette et traiter les données du formulaire.

Ce type de champ facilite l'expérience utilisateur en validant les entrées et en fournissant des suggestions automatiques pour les navigateurs et appareils compatibles.

Introduction au langage HTML

Histoire et Évolution du HTML

Exploiter un environnement de développement

Définir les éléments basiques d'une page HTML

Créer des formulaires HTML

Balise <form>

Champs de saisie (Input, TextArea)

Balises de choix (RadioButton, CheckBox, Select)

Boutons

## Type file

Le type file dans un élément <input> HTML permet à l'utilisateur de sélectionner un ou plusieurs fichiers à télécharger depuis son ordinateur ou appareil. Il est couramment utilisé dans les formulaires de téléchargement de fichiers.



Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Balise <form>  
Champs de saisie (Input, TextArea)  
Balises de choix (RadioButton, CheckBox, Select)  
Boutons

## Type file

Voici un exemple d'utilisation du type file

```
<form action="/upload" method="POST"  
enctype="multipart/form-data">  
<label for="fileUpload">Choisissez un fichier :</label>  
<input type="file" id="fileUpload" name="fileUpload" required>  
<input type="submit" value="Télécharger">  
</form>
```

## Type file

Explication des attributs :

- `type="file"` : indique que ce champ permet de sélectionner un fichier.
- `id="fileUpload"` et `name="fileUpload"` : identifiant et nom associés au champ, utilisés pour relier l'étiquette et traiter le fichier dans le backend.
- `required` : rend le téléchargement de fichier obligatoire avant la soumission du formulaire.
- `enctype="multipart/form-data"` : nécessaire pour permettre le transfert des fichiers via le formulaire.

## Type file

Attributs supplémentaires :

- accept : permet de restreindre les types de fichiers que l'utilisateur peut sélectionner (par exemple, uniquement les images ou les documents).
- multiple : permet la sélection de plusieurs fichiers à la fois.

Voici un exemple d'utilisation du type file

```
<form action="/upload" method="POST"
enctype="multipart/form-data">
<label for="fileUpload">Choisissez des fichiers d'image :</label>
<input type="file" id="fileUpload" name="fileUpload[]"
accept="image/*" multiple>
<input type="submit" value="Télécharger">
</form>
```

## Type time

Le type time dans un élément <input> HTML permet à l'utilisateur de sélectionner une heure (sans date). Il est généralement utilisé dans les formulaires pour demander une heure précise, comme pour la planification d'un rendez-vous ou d'un événement.

Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Balise <form>  
Champs de saisie (Input, TextArea)  
Balises de choix (RadioButton, CheckBox, Select)  
Boutons

## Type time

Voici un exemple d'utilisation du type time

```
<form action="/submit_time" method="POST">  
<label for="appt">Choisissez une heure :</label>  
<input type="time" id="appt" name="appt" required>  
<input type="submit" value="Envoyer">  
</form>
```

## Type time

Attributs supplémentaires utiles :

- min : permet de définir l'heure minimale qui peut être sélectionnée.
- max : permet de définir l'heure maximale qui peut être sélectionnée.
- step : permet de définir l'intervalle de temps (en secondes).

## Type time

### Exemple avec les attributs min, max et step

```
<form action="/submit_time" method="POST">  
<label for="appt">Choisissez une heure (entre 09:00 et 18:00)  
:</label>  
<input type="time" id="appt" name="appt" min="09:00" max="18:00"  
step="900" required>  
<input type="submit" value="Envoyer">  
</form>
```

## Frame Title

Explication :

- min="09 :00" : l'heure minimale est 9h00.
- max="18 :00" : l'heure maximale est 18h00.
- step="900" : l'intervalle est de 900 secondes, soit 15 minutes.

Cela permet de restreindre la sélection à une plage d'heures et de proposer des intervalles de temps spécifiques.



## Type url

Le type url dans un élément <input> HTML est utilisé pour demander à l'utilisateur de saisir une adresse web valide. Le navigateur effectue une validation automatique pour s'assurer que la valeur entrée est bien une URL au bon format, mais ne vérifie pas si l'URL existe réellement.

Introduction au langage HTML  
Histoire et Évolution du HTML  
Exploiter un environnement de développement  
Définir les éléments basiques d'une page HTML  
Créer des formulaires HTML

Balise <form>  
Champs de saisie (Input, TextArea)  
Balises de choix (RadioButton, CheckBox, Select)  
Boutons

## Type url

Voici un exemple d'utilisation du type url

```
<form action="/submit_url" method="POST">  
<label for="website">Votre site web :</label>  
<input type="url" id="website" name="website" required>  
<input type="submit" value="Envoyer">  
</form>
```

## Type url

Explication des attributs :

- `type="url"` : indique que le champ est destiné à recevoir une URL.
- `id="website"` et `name="website"` : identifiant et nom associés au champ, utilisés pour lier l'étiquette et traiter les données dans le backend.
- `required` : rend ce champ obligatoire, empêchant l'envoi du formulaire si l'URL n'est pas saisie ou si elle est mal formée.

Le type url aide à garantir que l'utilisateur entre une adresse web complète, généralement commençant par `"http ://"` ou `"https ://"`.

## Type range

Le type range dans un élément <input> HTML permet de créer un curseur interactif pour sélectionner une valeur numérique dans une plage spécifiée. Il est utilisé pour offrir une interface utilisateur simple pour la sélection de valeurs numériques dans une plage définie (par exemple, volume, luminosité, etc.).

## Type range

Voici un exemple d'utilisation du type range

```
<form action="/submit_range" method="POST">  
<label for="volume">Régler le volume :</label>  
<input type="range" id="volume" name="volume" min="0" max="100"  
step="1" value="50">  
<input type="submit" value="Envoyer">  
</form>
```

## Type range

Attributs couramment utilisés :

- min : définit la valeur minimale du curseur, la valeur minimale est 0.
- max : définit la valeur maximale du curseur, la valeur maximale est 100.
- step : définit l'intervalle entre chaque valeur sélectionnable, le curseur se déplace par incréments de 1.
- value : définit la valeur initiale du curseur, la valeur initiale affichée est 50.

## Type range

### Exemple avec étiquette pour afficher la valeur sélectionnée

```
<form>
<label for="brightness">Luminosité :</label>
<input type="range" id="brightness" name="brightness" min="0"
max="100" value="50" oninput="this.nextElementSibling.value =
this.value">
<output>50</output>
</form>
```

Dans cet exemple, le champ output affiche la valeur actuelle du curseur en temps réel grâce à l'attribut oninput.

## Type search

Le type search dans un élément <input> HTML est similaire au type text, mais il est optimisé pour la saisie de requêtes de recherche. Ce type peut être utilisé dans les barres de recherche pour permettre à l'utilisateur de saisir et de soumettre une requête de recherche.



# Type search

Voici un exemple d'utilisation du type search

```
<form action="/search_results" method="GET">  
<label for="search">Rechercher :</label>  
<input type="search" id="search" name="query" placeholder="Tapez  
votre recherche..." required>  
<input type="submit" value="Rechercher">  
</form>
```

## Type search

Explication des attributs :

- `type="search"` : indique que le champ est destiné à saisir une requête de recherche.
- `id="search"` et `name="query"` : identifiant et nom associés au champ, utilisés pour traiter les données saisies lors de l'envoi du formulaire.
- `placeholder="Tapez votre recherche..."` : affiche un texte indicatif dans le champ avant que l'utilisateur ne commence à saisir sa recherche.
- `required` : oblige l'utilisateur à entrer une requête avant de pouvoir soumettre le formulaire.

## L'attribut value

L'attribut value dans un élément <input> HTML spécifie la valeur initiale ou prédéfinie du champ de saisie. Lorsque le formulaire est soumis, la valeur actuelle de cet attribut est envoyée au serveur. Il est utilisé dans différents types d'entrées pour définir la valeur par défaut ou pour remplir un champ automatiquement avec une donnée.

# L'attribut value

## Utilisation de l'attribut value

```
<input type="radio" name="genre" value="male"> Homme
```

```
<input type="radio" name="genre" value="female"> Femme
```

```
<input type="checkbox" name="newsletter" value="yes"> S'abonner à  
la newsletter
```

## Utilisation de l'attribut value

L'attribut value est extrêmement utile lorsque vous souhaitez :

- Pré-remplir des champs dans des formulaires basés sur des informations connues.
- Fournir une valeur par défaut que l'utilisateur peut modifier ou non.
- Définir la valeur qui sera envoyée lors de la soumission du formulaire pour des éléments invisibles (hidden) ou visibles.

L'utilisateur peut généralement modifier cette valeur dans le champ, sauf pour certains types d'entrées comme les boutons ou les champs cachés (hidden).

## L'attribut readonly

L'attribut readonly dans HTML est utilisé pour rendre un champ de formulaire non modifiable, tout en permettant à l'utilisateur de voir et de sélectionner le contenu du champ. Contrairement à l'attribut disabled, le champ reste actif, et sa valeur sera envoyée lorsque le formulaire est soumis.

# L'attribut readonly

## Utilisation de l'attribut readonly

```
<form action="/submit">  
<label for="username">Nom d'utilisateur :</label>  
<input type="text" id="username" name="username"  
value="Utilisateur123" readonly> <br>  
<label for="email">Email :</label>  
<input type="email" id="email" name="email"  
value="exemple@domaine.com" readonly> <br>  
<input type="submit" value="Envoyer">  
</form>
```

## L'attribut readonly

Caractéristiques de l'attribut readonly :

- Le champ est visible : Contrairement à disabled, un champ readonly est toujours visible et son contenu peut être sélectionné et copié.
- Non modifiable : L'utilisateur ne peut pas changer le contenu du champ. Cependant, il peut voir le texte et le sélectionner.
- Soumis avec le formulaire : Contrairement aux champs disabled, les champs marqués avec readonly sont inclus lors de la soumission du formulaire, ce qui signifie que leurs valeurs sont envoyées avec les autres données.



## Différences avec disabled

Un champ avec disabled est grisé et non soumis lors de la soumission du formulaire, tandis qu'un champ readonly est visible, non modifiable mais sera soumis.

### Utilisation de l'attribut readonly

```
<input type="text" name="disabled_field" value="Non modifiable"  
disabled>
```

## L'attribut readonly

Quand utiliser readonly :

- Lorsque vous voulez afficher une valeur non modifiable (comme un identifiant utilisateur, un code généré automatiquement, etc.), mais que vous voulez toujours soumettre cette valeur avec le formulaire.
- Pour verrouiller un champ temporairement tout en permettant à l'utilisateur de voir son contenu.

### Utilisation de l'attribut readonly

```
<label for="orderId">Numéro de commande :</label>  
<input type="text" id="orderId" name="orderId" value="123456"  
readonly>
```

Introduction au langage HTML

Histoire et Évolution du HTML

Exploiter un environnement de développement

Définir les éléments basiques d'une page HTML

Créer des formulaires HTML

Balise <form>

Champs de saisie (Input, TextArea)

Balises de choix (RadioButton, CheckBox, Select)

Boutons

## L'attribut pattern

L'attribut pattern dans HTML permet de définir une expression régulière (ou regex) que l'entrée d'un utilisateur doit respecter dans un champ de formulaire. Il est couramment utilisé avec les types de champs tels que text, email, password, etc., pour imposer des formats spécifiques, comme des formats de numéros de téléphone, des adresses e-mail, des codes postaux, etc.

# L'attribut pattern

## Syntaxe de base

```
<input type="text" name="field_name"  
pattern="expression_régulière" title="Message d'erreur  
personnalisé">
```

- pattern : définit l'expression régulière que l'entrée de l'utilisateur doit respecter.
- title (optionnel) : permet d'afficher un message personnalisé si l'utilisateur ne respecte pas le format défini.

# L'attribut pattern

## Exemple d'utilisation

```
<form action="/submit_phone">  
<label for="phone">Numéro de téléphone :</label>  
<input type="tel" id="phone" name="phone" pattern="[0-9]10"  
title="Veuillez entrer un numéro de téléphone à 10 chiffres"  
required>  
<input type="submit" value="Envoyer">  
</form>
```

- `pattern="[0-9]10"` : L'expression régulière exige une chaîne composée exactement de 10 chiffres.
- `title` : affiche un message d'erreur personnalisé lorsque le format n'est pas respecté.

# L'attribut pattern

## Exemple d'utilisation

```
<form action="/submit_password">  
<label for="password">Mot de passe :</label>  
<input type="password" id="password" name="password"  
pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}" title="Le mot  
de passe doit contenir au moins 8 caractères, dont une majuscule,  
une minuscule et un chiffre" required>  
<input type="submit" value="Envoyer">  
</form>
```

- `pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}"` : Cette expression régulière oblige le mot de passe à contenir au moins un chiffre ( `\d` ), une lettre minuscule (`[a-z]`), une lettre majuscule (`[A-Z]`), et au moins 8 caractères au total (`.{8,}`).

## L'attribut pattern

Résumé des avantages :

- Validation côté client : pattern permet de valider les données avant même l'envoi du formulaire.
- Messages d'erreur personnalisés : En ajoutant l'attribut title, vous pouvez guider l'utilisateur avec des messages explicites sur le format attendu.
- Contrôle précis : Les expressions régulières permettent un contrôle très fin des formats acceptés dans les champs de formulaire.

En utilisant pattern, vous pouvez garantir que les données saisies correspondent exactement à ce que vous attendez, offrant une meilleure expérience utilisateur et des données plus propres.

## L'attribut placeholder

L'attribut placeholder en HTML est utilisé pour fournir un texte indicatif à l'intérieur d'un champ de saisie (<input> ou <textarea>). Ce texte s'affiche dans le champ tant que l'utilisateur n'a pas saisi de valeur. L'attribut placeholder est particulièrement utile pour donner des exemples de formats attendus ou pour informer l'utilisateur sur le type de données qu'il doit entrer.



# L'attribut placeholder

## Exemple d'utilisation

```
<form action="/submit">  
<label for="username">Nom d'utilisateur :</label>  
<input type="text" id="username" name="username"  
placeholder="Entrez votre nom d'utilisateur">  
<input type="submit" value="Envoyer">  
</form>
```

Dans cet exemple, "Entrez votre nom d'utilisateur" s'affiche dans le champ tant que l'utilisateur n'a pas commencé à saisir un texte.

## L'attribut autofocus

L'attribut autofocus en HTML est utilisé pour spécifier qu'un champ de formulaire (comme un <input>, <textarea>, ou un <select>) doit recevoir le focus automatiquement lorsque la page est chargée. Cela signifie que l'utilisateur peut commencer à saisir des données immédiatement sans avoir à cliquer sur le champ.

# L'attribut autofocus

## Exemple d'utilisation

```
<form action="/submit">  
<label for="username">Nom d'utilisateur :</label>  
<input type="text" id="username" name="username" autofocus>  
<input type="submit" value="Envoyer">  
</form>
```

Dans cet exemple, le champ "Nom d'utilisateur" sera automatiquement sélectionné lorsque la page se charge, permettant à l'utilisateur de commencer à taper immédiatement.

## Type submit

Le type submit est utilisé dans les éléments <input> et <button> en HTML pour créer un bouton qui soumet un formulaire lorsque l'utilisateur clique dessus. C'est un élément essentiel dans les formulaires web, car il permet d'envoyer les données saisies par l'utilisateur au serveur pour traitement.

# Type submit

## Syntaxe de base

```
<form action="/submit">  
<input type="submit" value="Texte du bouton">  
<button type="submit">Texte du bouton</button>  
</form>
```

## Exemple d'utilisation

### Exemple d'utilisation

```
<form action="/submit_form" method="post">  
<label for="name">Nom :</label>  
<input type="text" id="name" name="name" required> <br>  
<label for="email">Adresse e-mail :</label>  
<input type="email" id="email" name="email" required> <br>  
<input type="submit" value="Envoyer">  
</form>
```

Dans cet exemple, lorsque l'utilisateur clique sur le bouton "Envoyer", le formulaire est soumis à l'URL spécifiée dans l'attribut action du <form>, en utilisant la méthode spécifiée (dans ce cas, post).

## Type reset

Le type reset est utilisé dans les éléments <input> et <button> en HTML pour créer un bouton qui réinitialise tous les champs d'un formulaire à leurs valeurs par défaut lorsque l'utilisateur clique dessus. C'est un outil utile pour permettre aux utilisateurs de revenir rapidement à l'état initial du formulaire, en effaçant toutes les entrées qu'ils ont pu faire.

Introduction au langage HTML

Histoire et Évolution du HTML

Exploiter un environnement de développement

Définir les éléments basiques d'une page HTML

Créer des formulaires HTML

Balise <form>

Champs de saisie (Input, TextArea)

Balises de choix (RadioButton, CheckBox, Select)

Boutons

# Type reset

## Syntaxe de base

```
<form action="/submit">  
<input type="reset" value="Réinitialiser">  
<button type="reset">Réinitialiser</button>  
</form>
```



## Exemple d'utilisation

### Exemple d'utilisation

```
<form action="/submit_form" method="post">  
<label for="name">Nom :</label>  
<input type="text" id="name" name="name" required> <br>  
<label for="email">Adresse e-mail :</label>  
<input type="email" id="email" name="email" required> <br>  
<input type="submit" value="Envoyer">  
<input type="reset" value="Réinitialiser">  
</form>
```

Dans cet exemple, lorsque l'utilisateur clique sur le bouton "Réinitialiser", tous les champs du formulaire reviendront à leur état initial (par exemple, effaçant le texte saisi et restaurant les sélections par défaut).