# Cours d'initiation au HTML : Histoire, Structure, Syntaxe et Balises HTML courantes

## Histoire du HTML et d'Internet

Pour comprendre le HTML (HyperText Markup Language), il est important de connaître un peu l'histoire d'Internet. Dans les années 1960, le département de la Défense des États-Unis a créé ARPANET, un réseau de communication utilisé principalement par les universités et les institutions de recherche. C'est le prédécesseur d'Internet tel que nous le connaissons aujourd'hui.

Au fur et à mesure que le nombre d'utilisateurs d'ARPANET augmentait, il est devenu nécessaire de développer un moyen de partager des informations et de les présenter de manière structurée. C'est ainsi qu'est né le HTML. En 1991, Tim Berners-Lee a créé le premier navigateur Web, le WorldWideWeb, ainsi que le langage HTML. Depuis lors, le HTML a évolué pour devenir un langage essentiel pour la création de pages Web.

## Structure et Syntaxe du HTML

Le HTML est basé sur une structure en balises, qui permet de marquer les éléments et de les organiser dans une page Web. Chaque balise est encadrée par des chevrons angulaires (< et >), et certaines balises peuvent contenir du texte ou d'autres balises. Voici un exemple de structure de base d'une page HTML :

<!DOCTYPE html>

<html>

<head>

<title>Titre de la page</title>

</head>

<body>

Contenu de la page

</body>

</html>



- La balise **<!DOCTYPE html>** définit la version du HTML utilisée (HTML5 dans cet exemple).

- La balise **<html>** est l'élément racine de la page HTML.

- La balise **<head>** contient des informations sur la page, telles que le titre affiché dans la barre de titre du navigateur.

- La balise **<title>** spécifie le titre de la page.

- La balise **<body>** contient le contenu réel de la page.

## Balises HTML courantes

Il existe de nombreuses balises HTML disponibles, mais voici une liste des 12 balises les plus couramment utilisées, avec une description de leurs attributs les plus courants et des exemples d'utilisation :

### Balise header :

**<h1>** à **<h6>** : Ces balises définissent les niveaux de titres. L'attribut `class` peut être utilisé pour appliquer des styles spécifiques.

Exemple : <h1 class="titre">Titre principal</h1`

### Balise paragraphe :

**<p>** : Cette balise est utilisée pour définir un paragraphe de texte.

Exemple : <p>Voici un paragraphe.</p>

### Balise anchor :

**<a>** : Cette balise crée un lien hypertexte vers une autre page ou une ressource. L'attribut href spécifie l'URL de destination.

Exemple : <a href="https://www.example.com">Lien vers Example.com</a>

### Balise image :

**<img>** : Cette balise insère une image sur la page. L'attribut `src` spécifie l'URL de l'image.

Exemple : <img src="image.jpg" alt="Description de l'image">

### Balises liste et liste non ordonnée :

**<li>** et **<ul>** : Ces balises sont utilisées pour créer des listes non ordonnées. <ul> crée la liste et <li> définit chaque élément de la liste.

Exemple :

<ul>

<li>Élément 1</li>

<li>Élément 2</li>

</ul>



### Balise liste et liste ordonnée :

**<li>** et **<ol>** : Ces balises sont utilisées pour créer des listes ordonnées. <ol> crée la liste numérotée et <li> définit chaque élément de la liste.

Exemple :

<ol>

<li>Élément 1</li>

<li>Élément 2</li>

</ol>



### Balise division :

**<div>** : Cette balise est utilisée pour diviser le contenu en sections distinctes et peut être utilisée pour appliquer des styles spécifiques.

Exemple : <div class="section">Contenu de la section</div>

### Balise espace :

**<span>** : Cette balise est utilisée pour appliquer des styles spécifiques à une partie du texte à l'intérieur d'autres balises.

Exemple :

<p>Un texte normal avec du <span style="color: red;">texte rouge</span>.</p>

### Balise formulaire :

**<form>** : Cette balise est utilisée pour créer un formulaire interactif qui peut être soumis à un serveur. L'attribut `action` spécifie l'URL de destination.

Exemple :

<form action="/submit-form" method="post">

<!-- Les éléments du formulaire -->

</form>



### Balise Saisie :

**<input>** : Cette balise est utilisée pour créer des champs de saisie dans un formulaire.

Exemple : <input type="text" name="username" placeholder="Nom d'utilisateur">

### Balises pour tableaux :

**<table>**, **<tr>**, **<th>**, **<td>** : Ces balises sont utilisées pour créer des tableaux. <table> définit le tableau, <tr> définit une ligne, <th> définit une cellule d'en-tête et <td> définit une cellule de données.

Exemple :

<table>

<tr>

<th>En-tête 1</th>

<th>En-tête 2</th>

</tr>

<tr>

<td>Donnée 1</td>

<td>Donnée 2</td>

</tr>

</table>



### Balise break :

**<br>** : Cette balise est utilisée pour insérer un saut de ligne.

Exemple : <p>Ligne 1<br>Ligne 2</p>

Ces balises HTML courantes représentent seulement une petite partie des possibilités offertes par le langage. Il existe de nombreuses autres balises et attributs pour créer des pages Web riches et interactives.

## Frameworks Front-End populaires

Un framework signifie littéralement « cadre de travail ». Il fournit, en effet, une base d'outils et de modules qui peuvent être utilisés sur différents projets. Par exemple, un framework peut inclure des classes, variables et fonctions prédéfinies.

**Bootstrap** - Un framework CSS et JavaScript populaire pour la création de sites Web réactifs et adaptatifs. Il offre une collection de composants prêts à l'emploi, des grilles de mise en page et des styles personnalisables.

Site officiel <https://getbootstrap.com/>

Tutoriel <https://www.tutorialspoint.com/bootstrap/index.htm>

**React** - Une bibliothèque JavaScript développée par Facebook pour la création d'interfaces utilisateur interactives. Il utilise un modèle de composants réutilisables et est largement utilisé pour la construction d'applications Web modernes.

Site officiel <https://reactjs.org/>

Tutoriel <https://reactjs.org/tutorial/tutorial.html>

**Vue.js** - Un framework JavaScript évolutif et progressif pour la construction d'interfaces utilisateur. Il facilite la création de composants réutilisables et offre des fonctionnalités telles que le rendu côté serveur et le routage.

Site officiel <https://vuejs.org/>

Tutoriel <https://vuejs.org/v2/guide/>

**Angular** - Un framework TypeScript développé par Google pour la construction d'applications Web dynamiques et évolutives. Il offre un système complet pour la gestion de l'état, le routage et la création de composants réutilisables.

Site officiel <https://angular.io/>

Tutoriel <https://angular.io/tutorial>

**jQuery** - Une bibliothèque JavaScript populaire qui simplifie la manipulation du DOM, les animations et les requêtes AJAX. Elle permet d'ajouter facilement des fonctionnalités interactives à un site Web existant.

Site officiel <https://jquery.com/>

Tutoriel https://www.w3schools.com/jquery/

# Sources

[HTML (HyperText Markup Language) | MDN](https://developer.mozilla.org/fr/docs/Web/HTML)

[HTML Tutorial](https://www.w3schools.com/html/)

[Introduction · Bootstrap v5.0](https://getbootstrap.com/docs/5.0/getting-started/introduction/)

[Tutorial: Tic-Tac-Toe – React](https://reactjs.org/tutorial/tutorial.html)

[Introduction | Vue.js](https://vuejs.org/v2/guide/)

[Introduction to the Angular docs](https://angular.io/docs)

[jQuery](https://jquery.com/)