

# Intro à HTML

HTML, ou HyperText Markup Language, est le langage de balisage standard pour les documents conçus pour être affichés dans un navigateur web. Il peut être assisté par des technologies telles que les feuilles de style en cascade (CSS) et les scripts côté client (JavaScript).

## Historique des évolutions

1. **HTML (1991)** : Créé par Tim Berners-Lee, ce premier jet de HTML n'était pas une spécification complète mais plutôt un ensemble de règles permettant de créer des pages web simples. Il n'y avait que quelques balises disponibles pour la mise en forme du texte et la création de liens.
2. **HTML 2.0 (1995)** : Cette version a été standardisée par le IETF (Internet Engineering Task Force) et a ajouté des fonctionnalités comme les formulaires.
3. **HTML 3.2 (1997)** : Introduit par le W3C (World Wide Web Consortium), cette version a ajouté des scripts et des feuilles de style à HTML.
4. **HTML 4.0 (1997) et 4.01 (1999)** : Ces versions ont introduit des fonctionnalités comme la séparation du contenu et de la présentation (grâce à CSS), l'accessibilité, et le support pour diverses langues et caractères.
5. **XHTML (2000)** : Une révision de HTML en utilisant les règles de XML. Il était plus strict en termes de syntaxe.
6. **HTML5 (2014)** : Une mise à jour majeure qui a introduit des éléments pour le support de la vidéo et de l'audio, le dessin graphique 2D, le glisser-déposer, et d'autres fonctionnalités pour créer des applications web riches.

## Structure de base d'un document HTML

Un document HTML typique a une structure de base qui ressemble à ceci :

```
<!DOCTYPE html>
<html>
<head>
  <title>Titre de la page</title>
</head>
<body>
  <h1>Mon premier titre</h1>
  <p>Mon premier paragraphe.</p>
</body>
</html>
```

- `<!DOCTYPE html>` : Déclare le type de document et la version HTML.
- `<html>` : L'élément racine qui contient tous les autres éléments.
- `<head>` : Contient des métadonnées et des liens vers des feuilles de style ou des scripts.
- `<title>` : Définit le titre de la page, qui apparaît dans l'onglet du navigateur.

- `<body>` : Contient le contenu visible de la page.
- `<h1>`, `<p>` : Sont des exemples de balises HTML qui définissent le titre et le paragraphe, respectivement.

HTML est un langage incroyablement flexible qui a évolué au fil des ans pour répondre aux besoins croissants des développeurs et des utilisateurs. Il continue d'être un élément fondamental de la toile moderne.

Bien sûr ! Les langages à balises sont des langages de description qui utilisent des balises pour annoter des éléments dans un document. Ces balises permettent de structurer le texte et d'autres types de contenu, et elles peuvent être interprétées par des machines pour formater ou manipuler le document de manière appropriée.

## Caractéristiques des langages à balises

1. **Balilage**: Les balises sont généralement composées de caractères spéciaux comme `<` et `>` qui entourent un mot-clé pour indiquer le début et la fin d'un élément.
2. **Hiérarchie**: Les langages à balises sont souvent hiérarchiques, ce qui signifie que les balises peuvent être imbriquées les unes dans les autres pour créer une structure arborescente.
3. **Métadonnées**: Les balises peuvent contenir des attributs pour fournir des métadonnées supplémentaires sur un élément.
4. **Extensibilité**: Certains langages à balises sont extensibles, ce qui signifie que vous pouvez définir vos propres balises et attributs.

## Exemples de langages à balises

1. **HTML (HyperText Markup Language)**: Utilisé pour créer des pages web. Il est interprété par les navigateurs pour afficher du contenu web.
2. **XML (eXtensible Markup Language)**: Utilisé pour stocker et transporter des données. Il est très extensible et permet aux développeurs de définir leurs propres balises.
3. **SGML (Standard Generalized Markup Language)**: Un métalangage à balises sur lequel HTML et XML sont basés.
4. **Markdown**: Un langage de balisage léger utilisé pour formater du texte. Il est souvent converti en HTML pour l'affichage.
5. **LaTeX**: Utilisé principalement pour la composition de documents scientifiques et techniques, il permet un contrôle précis sur la mise en page et la typographie.
6. **BBCode**: Utilisé dans certains forums en ligne pour formater des messages, il est plus simple que HTML et conçu pour être sûr et rapide à analyser.
7. **JSON (JavaScript Object Notation)**: Bien que techniquement pas un langage à balises, JSON remplit une fonction similaire à XML en tant que format de données léger pour le stockage et le transport d'informations.

## Utilisations

Les langages à balises sont utilisés dans une variété de domaines, notamment:

- Le développement web
- Le stockage et le transfert de données
- La publication électronique
- La documentation technique
- Les interfaces utilisateur
- Les services web

Chaque langage à balises a ses propres règles syntaxiques et ses propres cas d'utilisation, mais ils partagent tous le concept de base de l'utilisation de balises pour structurer et annoter des informations.

Les langages à balises offrent plusieurs avantages qui les rendent utiles dans divers contextes :

## **Structuration des données**

Les balises permettent de structurer les données de manière hiérarchique et logique. Cette structuration facilite le traitement, le stockage et la récupération des données.

## **Interopérabilité**

Les langages à balises comme XML sont souvent utilisés pour échanger des données entre des systèmes hétérogènes. Leur nature textuelle et leur structure bien définie facilitent l'interopérabilité.

## **Flexibilité et extensibilité**

Certains langages à balises sont extensibles, ce qui signifie que vous pouvez définir vos propres balises et attributs pour répondre à des besoins spécifiques. Cela rend ces langages très flexibles.

## **Séparation du contenu et de la présentation**

Dans le cas de HTML et CSS, le langage à balises (HTML) est utilisé pour structurer le contenu, tandis qu'un autre langage (CSS) est utilisé pour la mise en forme. Cette séparation permet une meilleure maintenance et une plus grande réutilisabilité du code.

## **Accessibilité**

Les balises peuvent être utilisées pour fournir des métadonnées sur le contenu, ce qui peut améliorer l'accessibilité. Par exemple, en HTML, des balises comme `<alt>` pour les images et `<header>`, `<nav>`, `<main>`, `<footer>` pour la structure de la page aident à rendre le contenu plus accessible.

## **Facilité d'utilisation**

Les langages à balises sont souvent plus faciles à apprendre et à utiliser que d'autres types de langages de programmation, ce qui les rend accessibles à un plus grand nombre de personnes, y compris celles qui ne sont pas des développeurs de logiciels.

# Représentation visuelle

Dans le cas de HTML, les navigateurs Web interprètent les balises pour créer une représentation visuelle du contenu. Cela permet une grande variété de présentations sans avoir à manipuler le contenu lui-même.

## Recherche et indexation

La structure hiérarchique et les métadonnées fournies par les balises facilitent la recherche et l'indexation du contenu par les moteurs de recherche.

## Portabilité

Les documents basés sur des langages à balises sont généralement indépendants de la plateforme et peuvent être ouverts et manipulés par une variété d'outils et de technologies.

## Automatisation

La structure claire et les règles syntaxiques des langages à balises facilitent l'automatisation du traitement des données, y compris l'analyse, la transformation et la génération de documents.

En somme, les langages à balises offrent une manière structurée et flexible de représenter des données, ce qui les rend utiles dans une grande variété de domaines et d'applications.

# Aide mémoire au Langage HTML

---

## 1. Introduction

HTML (HyperText Markup Language) est le langage de balisage standard pour créer des pages web. Il permet de structurer le contenu sur le web de manière à ce qu'il soit interprété par les navigateurs.

---

## 2. Historique

- HTML a été créé par Tim Berners-Lee en 1991.
  - HTML 2.0, 3.2, 4.01 ont successivement ajouté de nouvelles fonctionnalités.
  - HTML5, la dernière version majeure, a été finalisée en 2014.
- 

## 3. Structure de base d'un document HTML

```
<!DOCTYPE html>  
<html>
```

```
<head>
  <title>Titre de la page</title>
</head>
<body>
  <!-- Contenu de la page -->
</body>
</html>
```

---

## 4. Éléments et balises

- Éléments de titre : `<h1>`, `<h2>`, `<h3>`, etc.
  - Éléments de paragraphe : `<p>`
  - Éléments de division : `<div>`
- 

## 5. Attributs

- `class` et `id` pour la sélection CSS.
  - `src` pour les sources d'images.
  - `href` pour les liens.
- 

## 6. Texte et typographie

- `<strong>` et `<em>` pour le texte en gras et en italique.
  - `<blockquote>` pour les citations.
  - `<pre>` pour le texte préformaté.
- 

## 7. Listes

- Listes à puces : `<ul>`
  - Listes ordonnées : `<ol>`
  - Éléments de liste : `<li>`
- 

## 8. Liens

- Liens hypertexte : `<a href="URL">`
-

## 9. Images

- Élément image : ``
- 

## 10. Tableaux

- `<table>` pour créer un tableau.
  - `<tr>` pour les lignes.
  - `<td>` pour les cellules.
- 

## 11. Formulaires

- `<form>` pour créer un formulaire.
  - `<input>`, `<textarea>`, `<select>` pour les champs de formulaire.
- 

## 12. Éléments sémantiques HTML5

- `<header>`, `<footer>`, `<article>`, `<section>`, etc.
- 

## 13. Intégration avec CSS et JavaScript

- Utilisation de `<link>` pour les feuilles de style CSS.
  - Utilisation de `<script>` pour le JavaScript.
- 

## 14. Validation et bonnes pratiques

- Utilisez le validateur W3C pour vérifier votre code.
- Assurez-vous que votre site est accessible.