Les Fondations : HTML

Bachelier en Développement d'Applications

Le Langage du Web : HTML

Construire le squelette de nos pages

À quoi sert une page web ?

Avant de coder, posons-nous la question : **quel est le but d'une page web ?**

> Le but principal est de **structurer** et de **présenter** de l'information à un utilisateur.

- 📰 Un article de journal
- 🛍️ Une page de produit
- 🎬 Une vidéo avec ses commentaires
- ✉️ Un formulaire de contact

L’Analogie du Squelette

Imaginez que vous construisez un être humain.

HTML

Le Squelette
(La structure)

CSS

L'Apparence
(La peau, les vêtements)

JavaScript / PHP

Le Cerveau
(L'intéraction, la logique)

On commence par la base : **la structure**.

Qu’est-ce que l’HTML ?

HTML = **H**yper**T**ext **M**arkup **L**anguage

HyperText

C’est du texte qui contient des liens (hyperliens) vers d’autres documents. C’est ce qui nous permet de "naviguer" sur le web.

Markup Language

C’est un "langage de balisage". On utilise des **balises** pour "marquer" le contenu et lui donner du sens.

⚠️ **Attention :** HTML n’est **PAS** un langage de programmation !

Il ne contient pas de logique (conditions, boucles…). Il ne fait que décrire la structure d’un document.

Un Peu d’Histoire

De l’HTML à l’HTML5

L’Évolution du Web

L’HTML n’a pas toujours été aussi puissant et structuré qu’aujourd’hui. Son histoire est celle du web lui-même.

- 📜 **1991 : La Naissance**

  Tim Berners-Lee, chercheur au CERN, crée l'HTML pour partager des documents scientifiques. Le but est simple : du texte et des liens.

- ⚔️ **1995-2000 : Les "Guerres des Navigateurs"**

  Netscape et Microsoft ajoutent chacun leurs propres balises HTML. C'est le chaos : un site qui fonctionne sur un navigateur est cassé sur l'autre.

- 🏛️ **2000-2010 : L'Âge de la Standardisation (HTML4 / XHTML)**

  Le W3C (World Wide Web Consortium) met de l'ordre en créant des règles strictes. Le code devient plus propre, mais aussi plus rigide et complexe.

- ✨ **2014 - Aujourd'hui : La Révolution HTML5**

  Une nouvelle version majeure, pensée pour le web moderne : vidéo, audio, interactivité, et une syntaxe plus

Alors, c’est quoi l’HTML5 ?

Un standard "vivant"

Contrairement aux anciennes versions, HTML5 n’est pas figé. C’est un **standard vivant**, continuellement mis à jour par un groupe appelé le WHATWG (Web Hypertext Application Technology Working Group).

On ne parle plus de "HTML6" ou "HTML7". On dit simplement **"HTML"**, en sous-entendant qu’on utilise la dernière version du standard vivant.

HTML5 = HTML (la version moderne)

Les grandes nouveautés

HTML5 a été conçu pour répondre aux besoins d’aujourd’hui, en intégrant nativement des fonctionnalités qui nécessitaient avant des plugins (comme Flash).

- 🎬 Intégration facile de **vidéos** (`<video>`) et de **sons** (`<audio>`).
- 🏛️ De nouvelles **balises sémantiques** pour mieux décrire la structure de la page (`<header>` , `<nav>` , `<footer>` , `<article>`).
- 📝 Des **formulaires** plus intelligents et interactifs.
- 🎨 La possibilité de **dessiner** dans la page (`<canvas>`).
- Et bien plus encore !

Pourquoi est-ce important pour vous ?

Apprendre l'HTML aujourd'hui, c'est directement apprendre l'HTML5. Vous partez sur des bases saines et pérennes.

- ✅ Vous apprenez le **standard actuel**, utilisé par tous les sites web modernes.
- 📱 Vous apprenez des outils pensés pour le **mobile**, la **vidéo** et l'**accessibilité**.
- ✍️ Vous allez apprendre à écrire un code plus **clair** et plus **significatif** grâce aux balises sémantiques.

> En bref, vous apprenez la bonne version, de la bonne manière.

La structure de l’HTML

La Brique de Base : La Balise (Tag)

Une balise HTML est une étiquette que l'on place autour d'un contenu pour dire au navigateur ce que ce contenu représente.

"Hey Navigateur, ce texte est un titre de niveau 1 !"

> Les balises **conteneurs** fonctionnent par paires : une balise **ouvrante** et une balise **fermante**.

```
<p>Ceci est un paragraphe.</p>
```

`<p>`	"Ceci est un paragraphe."	`</p>`
Balise ouvrante	Le Contenu	Balise fermante (notez le `/`)

Notre Première Page : Un Titre

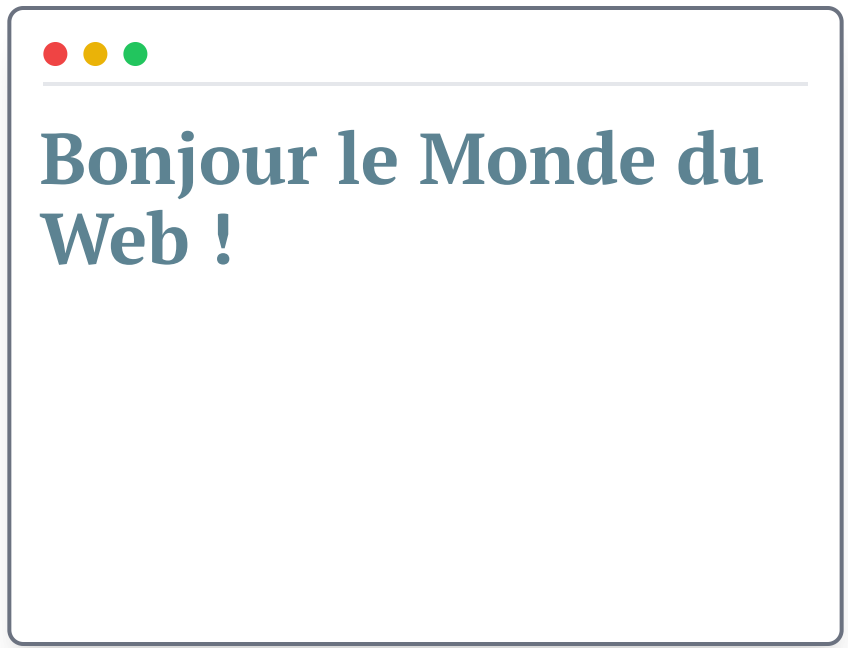
Le Code HTML

On va utiliser la balise `<h1>` qui signifie "Heading 1" (Titre de niveau 1), le plus important de la page.

```
<h1>Bonjour le Monde du Web !</h1>
```

Ici, on indique au navigateur que le texte "Bonjour le Monde du Web !" doit être traité comme le titre principal de notre page.

Le Résultat dans le Navigateur



L’Exception à la Règle : Les Balises Autofermantes

Toutes les balises ne fonctionnent pas par paires.

> Certaines balises sont "vides" : elles n’ont pas de contenu à encadrer. Elles servent à **insérer un élément** directement dans la page. Elles n’ont donc pas besoin de balise fermante.

Exemples courants :

- `<img>` : Pour insérer une image 🖼️
- `<br>` : Pour forcer un retour à la ligne (**br**eak)
- `<hr>` : Pour afficher une ligne de séparation (**h**orizontal **r**ule)

Par exemple, pour sauter une ligne à l'intérieur d'un paragraphe :

```
<p>
  Ceci est la première ligne.<br>Et voici la deuxième, après le saut de ligne.
</p>
```

⚠️ **Attention :** en HTML, le retour à la ligne n'a aucun effet visuel ! C'est uniquement pour la structure et la lisibilité du code HTML.

Une balise autofermante peut contenir un `/` de fermeture.

Exemple : `<br/>`

Afin de ne pas se tromper, nous vous recommandons dans ce cours de ne pas le mettre. La position du `/` dans une balise autofermante est importante !! `</br>` n'est pas valide !!

Donner plus d'informations : Les Attributs

Jusqu'à présent, nos balises donnent la **nature** d'un contenu (`<p>` = paragraphe, `<h1>` = titre).

Mais comment donner des **informations supplémentaires** à ces balises ?

C'est le rôle des **attributs**. Un attribut est une information additionnelle qui configure ou modifie le comportement d'une balise.

Analogie :

- La balise est le **nom** : `Voiture`
- L'attribut est l'**adjectif** : `couleur="rouge"`

La Syntaxe d’un Attribut

Un attribut se place toujours dans la **balise ouvrante**, jamais dans la fermante.

Il est composé d’une paire **nom="valeur"**.

```
<balise nom_attribut="valeur_attribut">Contenu</balise>
```

`nom_attribut`	`=`	`"valeur_attribut"`
Le nom de l'information (ex: `href`, `src`, `class`...)	Le signe "égal"	La valeur, toujours entre guillemets (" ")

⚠️ **Erreur de débutant :** Oublier les guillemets autour de la valeur. Prenez l’habitude de toujours les mettre !

L'Attribut le Plus Important : `href`

Vous vous souvenez de "HyperText" ? C'est ici qu'il prend tout son sens.

La balise `<a>` (ancre, *anchor* en anglais) sert à créer un lien, mais seule, elle ne fait rien. Elle a besoin de l'attribut `href` (**h**ypertext **ref**erence) pour savoir **vers où** pointer.

Code HTML

```
<p>
  Visitez le site de
  <a href="https://www.google.com">Google</a>
  pour faire une recherche.
</p>
```

Résultat dans le navigateur

Visitez le site de Google pour faire une recherche.

Sans l'attribut `href`, le mot "Google" ne serait qu'un simple texte. C'est `href` qui le transforme en un lien cliquable.

Un Autre Exemple Crucial : Les Images

Nous avons vu la balise autofermante `<img>`. Comment sait-elle quelle image afficher ?

Grâce à deux attributs essentiels :

`src` (source)

C'est le plus important ! Il indique le **chemin** (l'URL) vers le fichier de l'image. Sans lui, aucune image ne s'affiche.

`alt` (texte alternatif)

Ce texte est crucial pour l'**accessibilité**. Il est lu par les lecteurs d'écran pour les personnes malvoyantes et s'affiche si l'image ne peut pas être chargée.

Code complet :

```
<img
  src="https://picsum.photos/id/237/400/200"
  alt="Une photo en noir et blanc d'un chiot noir."
>
```

💡 **Bonne pratique :**

Remplissez **TOUJOURS** l'attribut `alt`. C'est une marque de professionnalisme et de respect pour tous vos utilisateurs.

La Structure d’une Page HTML

De la balise simple au document complet

Le Squelette de Base d'une Page

Toutes les balises que nous avons vues doivent être placées dans un "contenant" structuré que le navigateur peut comprendre.

Ce code est le point de départ **OBLIGATOIRE** de tout fichier `.html` que vous créerez.

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Initiation au langage HTML</title>
    </head>
    <body>

    </body>
</html>
```

L’Anatomie d’une Page HTML

Décortiquons ce squelette ligne par ligne pour comprendre le rôle de chaque élément.

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Titre de ma page</title>
    </head>
    <body>
        <!-- Le contenu visible vient ici -->
    </body>
</html>
```

- `<!DOCTYPE html>` : Déclare au navigateur qu’il s’agit d’un document HTML5.
- `<html>` : La balise racine qui englobe **toute** la page.
- `<head>` : Contient les "méta-informations" de la page (non visibles).
- `<body>` : Contient **tout** le contenu visible de la page.

Les Fondations : `<!DOCTYPE>` et `<html>`

`<!DOCTYPE html>`

- C’est une **instruction**, pas une balise. Elle doit toujours être la toute première ligne du fichier.
- Elle informe le navigateur d’utiliser le "mode standard" le plus récent pour interpréter le code, ce qui garantit un affichage cohérent.

`<html lang="fr">`

- C’est la balise "mère", le conteneur principal de la page.
- L’attribut `lang="fr"` est très important : il indique que la langue principale de la page est le français. C’est utile pour :
  - Les moteurs de recherche (SEO) SEO 🔍
  - Les outils d’accessibilité (lecteurs d’écran) ♿
  - Les correcteurs orthographiques du navigateur ✍️

La Tête Pensante : La balise `<head>`

Le contenu de `<head>` n'est **PAS visible** sur la page, mais il est crucial pour son bon fonctionnement. C'est le "cerveau" 🧠 de notre document, il contient des informations sur la page.

- `<meta charset="utf-8">`

  Indispensable pour que les caractères spéciaux et les accents (é, à, ç...) s'affichent correctement. Sans lui, vous pourriez voir des `Ã©` ou des `?`.

- `<meta name="viewport" ...>`

  Dit au navigateur (surtout sur mobile) comment gérer la taille de la page. C'est la première étape pour rendre un site "responsive" (adapté aux mobiles).

- `<title>`

  Définit le titre qui s'affiche dans l'**onglet du navigateur** 📑 et dans les résultats de recherche Google.

Le Corps Visible : La balise `<body>`

C’est ici que la magie opère ! ✨

Tout ce que l’utilisateur voit et avec quoi il interagit doit se trouver à l’intérieur de la balise `<body>`.

C’est là que nous mettrons nos titres `<h1>`, nos paragraphes `<p>`, nos images `<img>`, nos liens `<a>`, etc.

```
<body>
    <h1>Mon Super Titre</h1>
    <p>
        Ceci est mon premier paragraphe avec un
        <a href="page2.html">lien</a>.
    </p>
    <img src="images/logo.png" alt="Mon logo">
</body>
```

Résumé : `<head>` vs `<body>`

🧠 `<head>` (L'invisible)

- Méta-informations pour le navigateur et les moteurs de recherche.
- Encodage des caractères (`charset`).
- Titre de l'onglet (`title`).
- Informations pour le responsive design.
- *(Plus tard, on y liera nos fichiers CSS et JavaScript)*

👀 `<body>` (Le visible)

- Le contenu réel de la page.
- Le texte, les titres.
- Les images, les vidéos.
- Les liens.
- Les formulaires.
- Tout ce que l'utilisateur peut voir et lire.

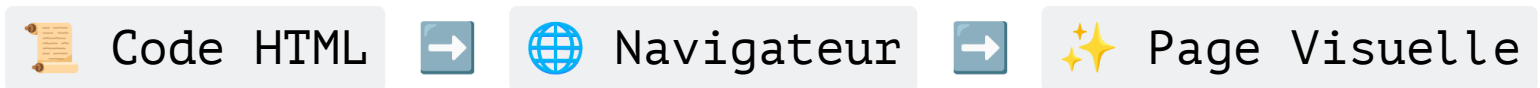
Le Rôle du Navigateur Web

Le traducteur de notre code

Qu’est-ce qu’un navigateur ?

Un navigateur (Chrome, Firefox, Safari, Edge…) est un logiciel **traducteur**.

Son travail principal est de lire du code (HTML, CSS, JavaScript) et de le transformer en une page web visuelle et interactive que nous pouvons utiliser.



Chaque navigateur a son propre "moteur de rendu", mais ils suivent tous les mêmes règles (les standards du W3C) pour :

- **Interpréter** la structure HTML.
- **Appliquer** les styles CSS.
- **Exécuter** le code JavaScript.

Le processus d’interprétation

Quand vous demandez une page web, voici ce que le navigateur fait en quelques millisecondes :

1. **Réception :** Il reçoit le fichier `.html` depuis un serveur web.
2. **Analyse (Parsing) :** Il lit votre code HTML de haut en bas, balise par balise, pour en comprendre la structure et la hiérarchie.
3. **Construction du DOM :** Il construit une représentation interne de votre page, une sorte d'arbre généalogique des balises. C'est le **DOM** (Document Object Model).
4. **Rendu (Rendering) :** Il utilise cet "arbre" pour dessiner chaque élément (titre, paragraphe, image…) à l'écran, en appliquant les styles par défaut.

La structure interne : le DOM

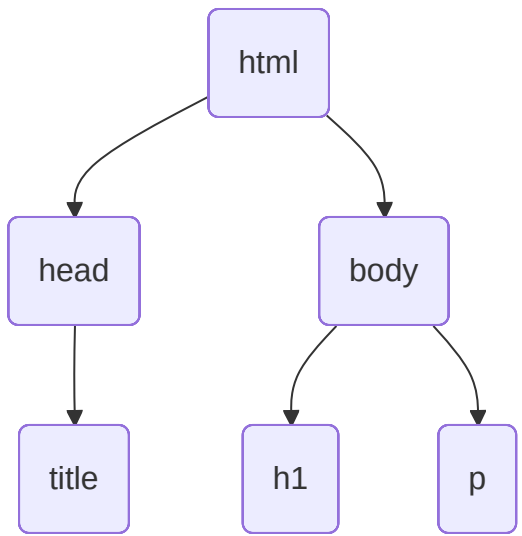
Le Code HTML

Le navigateur lit ce code simple et hiérarchisé.

```
<html>
  <head>
    <title>Ma page</title>
  </head>
  <body>
    <h1>Titre</h1>
    <p>Un paragraphe.</p>
  </body>
</html>
```

L'Arbre DOM

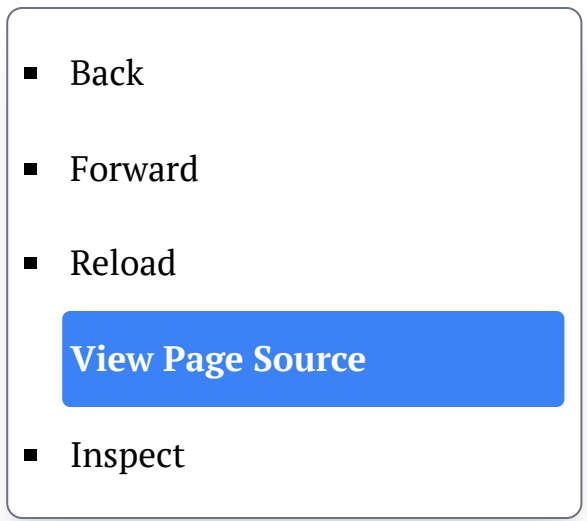
Et le transforme en cette structure logique, qui ressemble à un arbre généalogique. C'est sur cette structure que le CSS et le JavaScript pourront agir.



Votre super-pouvoir : "Afficher le code source"

Chaque navigateur vous donne la possibilité de voir le "squelette" HTML de n’importe quelle page web !

Faites un clic droit sur une page web et cherchez l'option :



C’est un excellent moyen d’apprendre en regardant comment les sites professionnels sont construits.

L’outil indispensable : Les Outils de Développement

Chaque navigateur intègre une suite d’outils puissants pour les développeurs (les "DevTools").

Appuyez sur la touche `F12` sur votre clavier !

L’onglet le plus important pour nous au début est l’onglet **"Éléments"** (ou "Inspecteur").

Il vous montre l'**arbre DOM en direct**. C'est une version interactive du "code source". Vous pouvez déplier les balises, voir leur hiérarchie et même (temporairement) modifier le contenu de la page !

```
<body>
    <title>Mon Titre</title>
    <p>Bonjour</p>
</body>
```

💡 **Habitude à prendre :** Dès qu’un élément ne s’affiche pas comme prévu, ouvrez les DevTools. C’est votre meilleur ami pour comprendre ce qui se passe.

La Boîte à Outils HTML

Les balises essentielles du quotidien

On ne va pas tout voir ! (Et c’est normal)

Il existe plus de **100 balises HTML**, chacune avec son propre rôle. Mémoriser la liste complète est inutile et impossible.

Le vrai travail d’un développeur n’est pas de tout savoir, mais de **savoir où chercher**.

Votre meilleure ressource, votre dictionnaire pour le web, est le **MDN (Mozilla Developer Network)**.

Lien vers la Référence des Éléments HTML du MDN ↗️

Structurer avec les Titres : `<h1>` à `<h6>`

Le Code HTML

Les titres organisent votre contenu de manière hiérarchique. `<h1>` est le plus important, `<h6>` le moins important.

Règle d'or : Il ne doit y avoir qu'**un seul** `<h1>` par page. C'est le titre principal de votre document.

```
<h1>Titre principal de la page</h1>
<h2>Un sous-titre important</h2>
<h3>Un sous-niveau</h3>
<p>Du texte qui suit...</p>
<h4>Un autre sous-niveau</h4>
```

Le Résultat

Le navigateur applique un style par défaut (taille, gras) pour différencier les niveaux.

Le Paragraphe `<p>` et le Saut de Ligne `<br>`

- `<p>` : Utilisé pour grouper des phrases en un **bloc de texte cohérent**. Le navigateur ajoute automatiquement un espace avant et après chaque paragraphe.
- `<br>` : Force un **retour à la ligne** à l'intérieur d'un bloc, sans créer un nouveau paragraphe. C'est une balise autofermante.

```
<p>
  Ceci est un paragraphe complet. Il peut contenir
  plusieurs phrases et s'étendre sur plusieurs lignes
  dans le code.
</p>
<p>
  Voici la première ligne d'un autre paragraphe.<br>
  Grâce à la balise &lt;br&gt;, ceci est la deuxième
  ligne, mais nous sommes toujours dans le même
  paragraphe.
</p>
```

Résultat

Ceci est un paragraphe complet. Il peut contenir plusieurs phrases et s'étendre sur plusieurs lignes dans le code.

Voici la première ligne d'un autre paragraphe.
Grâce à la balise <br>, ceci est la deuxième ligne, mais nous sommes toujours dans le même paragraphe.

Mettre en Évidence : `<strong>` et `<em>`

- `<strong>` : Indique que le texte a une **grande importance**, une urgence ou une gravité. Visuellement, il est affiché en **gras**.
- `<em>` (*emphasis*) : Met l'**accent** sur un mot ou une phrase, changeant le sens de la phrase. Visuellement, il est affiché en *italique*.

```
<p>
  <strong>Attention :</strong> N'oubliez jamais
  de sauvegarder votre travail.
</p>
<p>
  Je ne dis <em>pas</em> que c'est une mauvaise idée,
  je dis juste de faire attention.
</p>
```

Attention : N'oubliez jamais de sauvegarder votre travail.

Je ne dis *pas* que c'est une mauvaise idée, je dis juste de faire attention.

💡 **Important :** Utilisez ces balises pour le **sens**, pas juste pour le style. Si vous voulez juste du gras ou de l'italique pour des raisons

Organiser avec les Listes

Liste non ordonnée `<ul>`

Utilisez `<ul>` (*unordered list*) quand l'ordre des éléments n'a pas d'importance. Chaque élément est un `<li>` (*list item*).

```
<!-- Liste de courses -->
<ul>
  <li>Lait</li>
  <li>Oeufs</li>
  <li>Pain</li>
</ul>
```

- Lait
- Oeufs
- Pain

Liste ordonnée `<ol>`

Utilisez `<ol>` (*ordered list*) pour une séquence, un classement ou des étapes à suivre.

```
<!-- Étapes de la recette -->
<ol>
  <li>Casser les oeufs</li>
  <li>Battre les oeufs</li>
  <li>Cuire dans la poêle</li>
</ol>
```

1. Casser les oeufs
2. Battre les oeufs
3. Cuire dans la poêle

Liens `<a>` et Images `<img>` (Rappel)

On combine souvent ces deux balises ! Par exemple, pour rendre une image cliquable.

- `<a>` : Crée un hyperlien avec l'attribut `href`.
  - `target="_blank"` est un attribut utile pour ouvrir le lien dans un nouvel onglet.
- `<img>` : Affiche une image avec les attributs `src` et `alt`.

```
<!-- Un lien simple vers une autre page -->
<a href="contact.html">Nous contacter</a>

<!-- Une image qui est aussi un lien -->
<a href="https://fr.wikipedia.org" target="_blank">
  <img
    src="images/logo_wikipedia.png"
    alt="Logo de Wikipedia"
    width="150"
  >
</a>
```

Présenter des Données : Les Tableaux

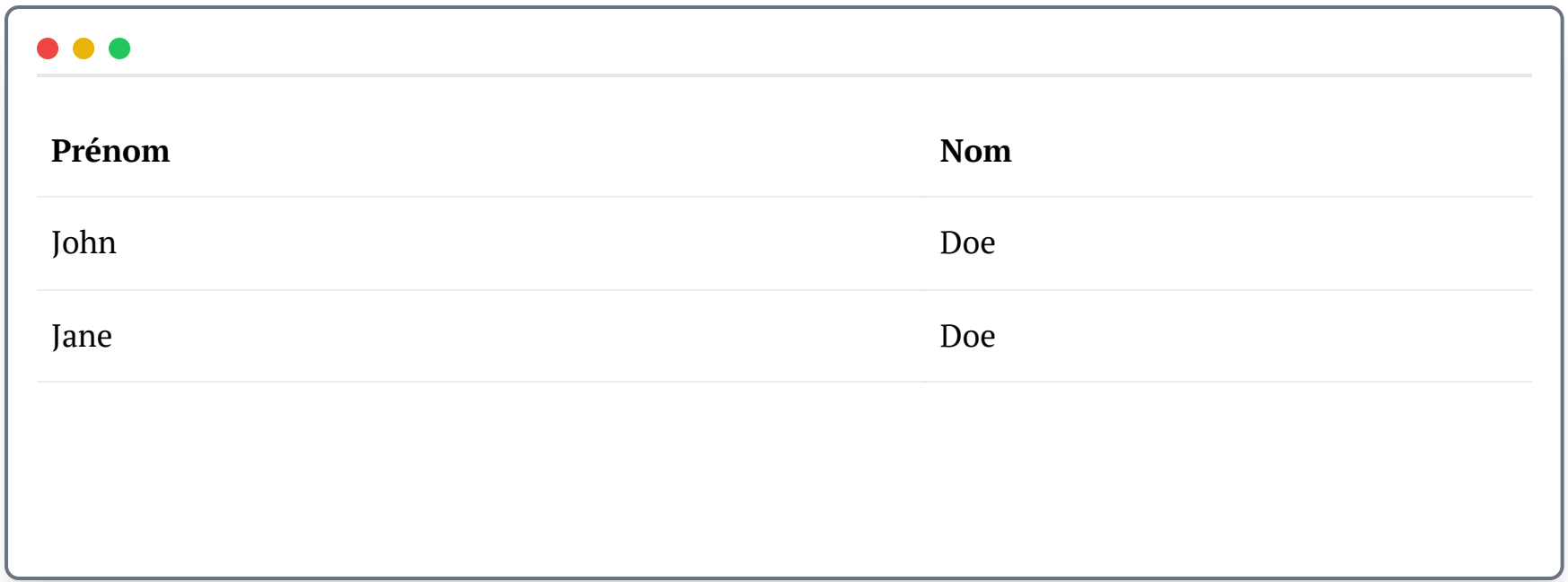
Les tableaux sont utilisés **uniquement pour des données tabulaires** (horaires, statistiques, etc.). N'utilisez **JAMAIS** de tableaux pour la mise en page de votre site !

- `<table>` : Le conteneur du tableau.
- `<tr>` (*table row*) : Définit une ligne.
- `<th>` (*table head*) : Une cellule d'en-tête (en gras et centrée par défaut).
- `<td>` (*table data*) : Une cellule de donnée classique.

```
<table>
  <tr>
    <th>Prénom</th>
    <th>Nom</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
  </tr>
  <tr>
    <td>Jane</td>
    <td>Doe</td>
  </tr>
</table>
```

Le Résultat

Le navigateur applique un style par défaut (taille, gras) pour différencier les niveaux.



Prénom	Nom
John	Doe
Jane	Doe

Les Caractères Spéciaux : Entités HTML

Que faire si vous voulez afficher le caractère `<` ou `>` dans votre page ? Le navigateur va croire que c'est une balise !

Solution : Les **entités HTML**, un code spécial qui représente un caractère.

- Pour afficher `<p>`, on écrira : `<p>`
- Le symbole "copyright" © s'écrit : `&copy;`
- Le symbole "euro" € s'écrit : `&euro;`
- L'esperluette & s'écrit : `&` (pour *ampersand*)

> 💡 Il en existe des centaines. Une recherche rapide "html entity for..." vous donnera toujours celui dont vous avez besoin.