

```
22 <!-- index.html -->
23   <a href="#" alt="Bondi Logo" />
24 </div>
25 <div id="links" class="links">
26   <ul>
27     <li><a href="index.html" class="active">Home</a></li>
28     <li><a href="#services">Services</a></li>
29     <li><a href="#portfolio">Portfolio</a></li>
30     <li><a href="#about">About</a></li>
31     <li><a href="#contact">Contact</a></li>
32   </ul>
33 </div>
34 <div class="search">
35   <button aria-label="Search">
36     <i class="fa-solid fa-magnifying-glass search-icon" data-bbox="125 445 315 555"></i>
37   </button>
38 </div>
39 <div class="login">
40   <button>Login</button>
41 </div>
42 </div>
43 <div class="toggle-menu-container">
44   <i id="toggle-menu" class="fa-solid fa-bars toggle-menu"></i>
45 </div>
46 </div>
47 </div>
48 <!-- End Header -->
49 <!-- Start Hero -->
50 <div class="hero">
51   <div class="container">
52     <div class="content">
```



Hyper Text Markup Language

2025/2026

HTML Introduction

HTML est le langage de balisage standard pour la création de pages Web.

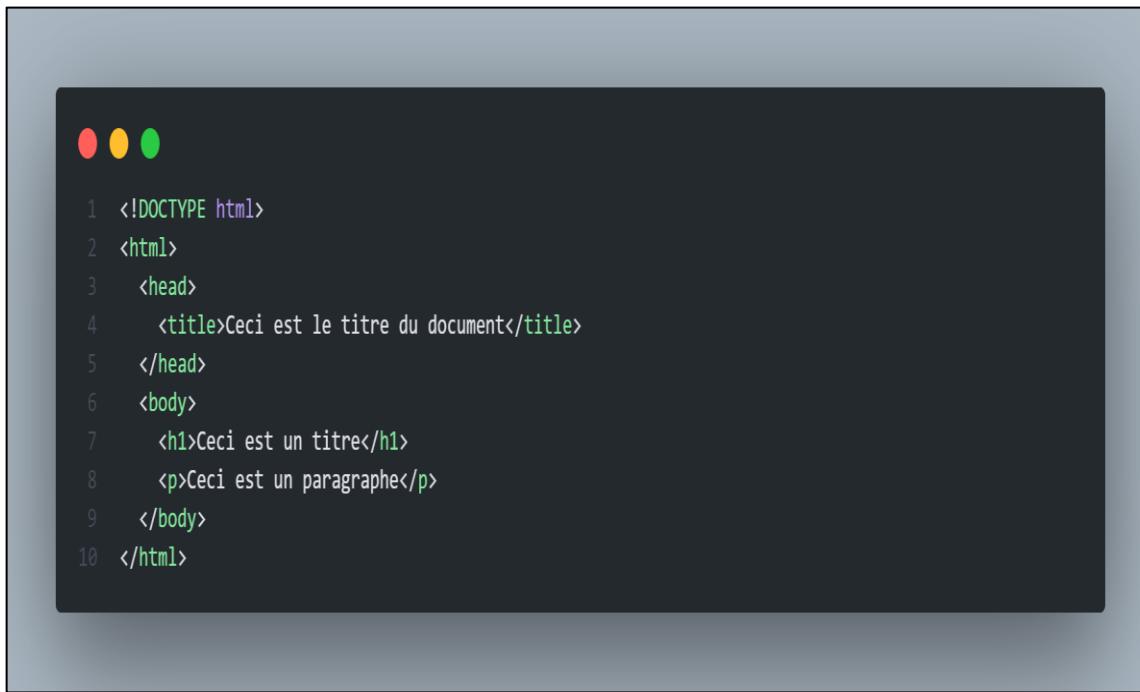
Qu'est-ce que HTML?

- HTML signifie Hyper Text Markup Language(langage de balisage hypertexte)
- HTML est le langage de balisage standard pour la création de pages Web
- HTML décrit la structure d'une page Web
- HTML se compose d'une série d'éléments
- Les éléments HTML indiquent au navigateur comment afficher le contenu
- Les éléments HTML étiquettent des éléments de contenu tels que "ceci est un titre", "ceci est un paragraphe", "ceci est un lien ", etc.

HTML Introduction

Un document HTML simple:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ceci est le titre du document</title>
5   </head>
6   <body>
7     <h1>Ceci est un titre</h1>
8     <p>Ceci est un paragraphe</p>
9   </body>
10 </html>
```



HTML Introduction

Exemple expliqué:

- La déclaration `<!DOCTYPE html>` définit que ce document est un document HTML5
L'élément `<html>` est l'élément racine d'une page HTML
- L'élément `<head>` contient des méta-information sur la page HTML
- L'élément `<title>` spécifie un titre pour la page HTML (qui s'affiche dans la barre de titre du navigateur ou dans l'onglet de la page)
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- L'élément `<h1>` définit un grand titre
- L'élément `<p>` définit un paragraphe

HTML Introduction

Qu'est-ce qu'un élément HTML?

Un élément HTML est défini par une balise de début, du contenu et une balise de fin:

<nomdelabalise>Le contenu va ici...</nomdelabalise>

L'élément HTML comprend tout, de la balise de début à la balise de fin:

<h1>Ceci est un titre</h1>

<p>Ceci est un paragraphe </p>

Balise de début	Contenu de l'élément	Balise de fin
<h1>	Ceci est un titre	</h1>
<p>	Ceci est un paragraphe	</p>
 	aucun	aucune

Remarque: certains éléments HTML n'ont aucun contenu (comme l'élément
). Ces éléments sont appelés éléments vides. Les éléments vides n'ont pas de balise de fin!

HTML Introduction

Navigateurs Web:

Le but d'un navigateur Web (Chrome, Edge, Firefox, Safari) est de lire des documents HTML et de les afficher correctement.

Un navigateur n'affiche pas les balises HTML, mais les utilise pour déterminer comment afficher le document:



HTML Introduction

Structure de la page HTML:

Vous trouverez ci-dessous une visualisation de la structure d'une page HTML:

```
<html>
    <head>
        <title>Ceci est le titre du document</title>
    </head>

    <body>
        <h1>Ceci est un titre</h1>
        <p>Ceci est un paragraphe</p>
        <p>Ceci est un autre paragraphe</p>
    </body>
</html>
```

Remarque: le contenu de la section `<body>` s'affiche dans un navigateur. Le contenu de l'élément `<title>` s'affiche dans la barre de titre du navigateur ou dans l'onglet de la page.

HTML Introduction

Historique du HTML:

Depuis les débuts du World Wide Web, il existe de nombreuses versions du HTML:

Année	Version
1989	Tim Berners-Lee a inventé le www
1991	Tim Berners-Lee a inventé le HTML
1993	Dave Raggett a rédigé le code HTML
1995	Le groupe de travail HTML a défini HTML 2.0
1997	Recommandation W3C : HTML 3.2
1999	Recommandation W3C : HTML 4.01
2000	Recommandation W3C : XHTML 1.0
2008	Première ébauche publique du WHATWG HTML5

HTML Introduction

2012	Norme de vie HTML5 WHATWG
2014	Recommandation W3C: HTML5
2016	Recommandation du candidat W3C: HTML 5.1
2017	Recommandation W3C: HTML5.1 2e édition
2017	Recommandation W3C: HTML5.2

Ce tutoriel suit la dernière norme HTML5.

Éditeurs HTML

Un éditeur de texte est tout ce dont vous avez besoin pour apprendre le HTML.

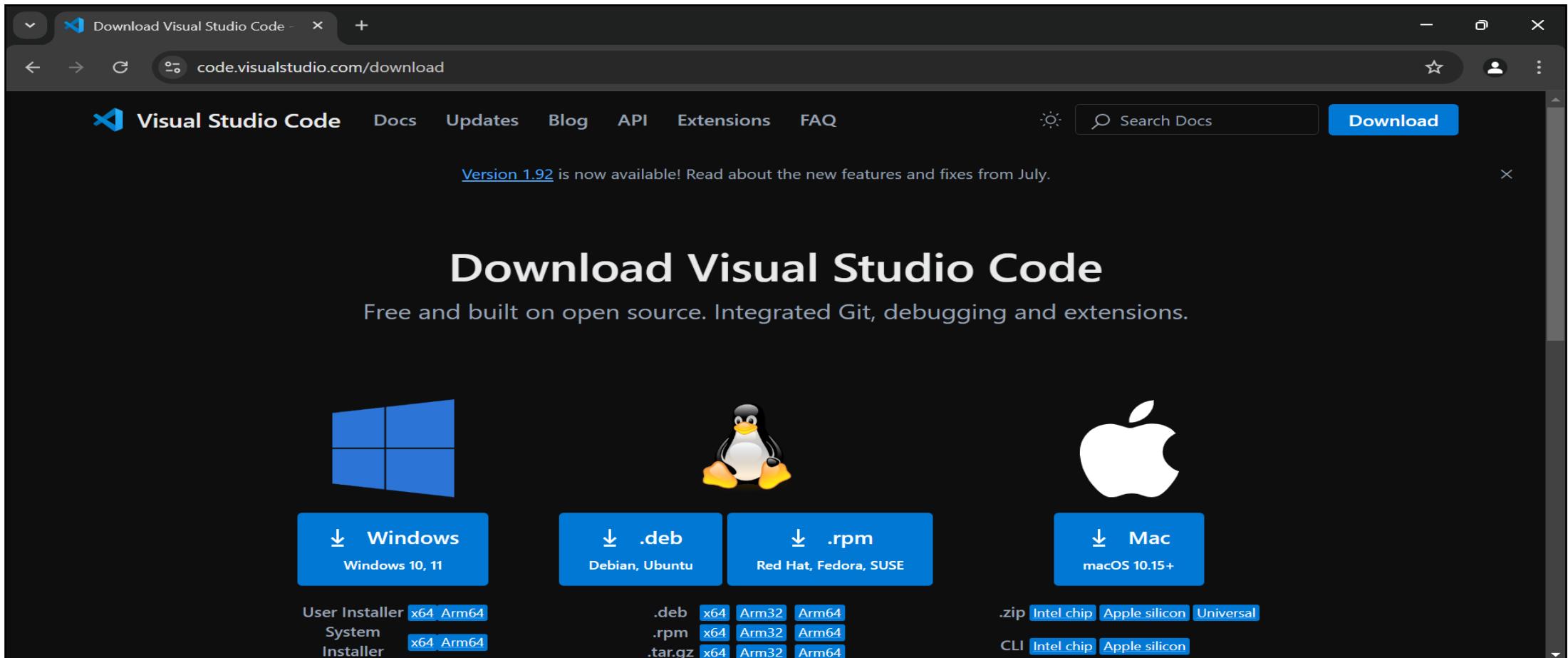
Apprenez le HTML avec VS Code:

Les pages Web peuvent être créées et modifiées à l'aide de divers outils, des éditeurs HTML professionnels comme VS Code, Atom et Sublime Text aux éditeurs de texte simples comme Notepad (sur Windows) ouTextEdit (sur macOS). Dans ce tutoriel, nous allons nous concentrer sur la façon d'écrire et de modifier du code HTML à l'aide de VS Code. Nous avons choisi VS Code car c'est un éditeur de code puissant et polyvalent qui offre des fonctionnalités puissantes.

Suivez les étapes ci-dessous pour créer votre première page Web avec VS Code.

Éditeurs HTML

Étape 1: Téléchargez et installez VS Code depuis
<https://code.visualstudio.com/download>

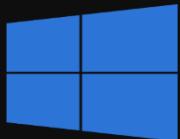


The screenshot shows a dark-themed web browser window displaying the Visual Studio Code download page at code.visualstudio.com/download. The page features a navigation bar with links for Visual Studio Code, Docs, Updates, Blog, API, Extensions, and FAQ. A prominent "Download" button is located in the top right corner. A message通知 Version 1.92 is available. The main section is titled "Download Visual Studio Code" and describes it as free and built on open source, with integrated Git, debugging, and extensions. It provides download links for Windows, Linux, and Mac operating systems.

Version 1.92 is now available! Read about the new features and fixes from July.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

Windows
Windows 10, 11

.deb
Debian, Ubuntu

.rpm
Red Hat, Fedora, SUSE

Mac
macOS 10.15+

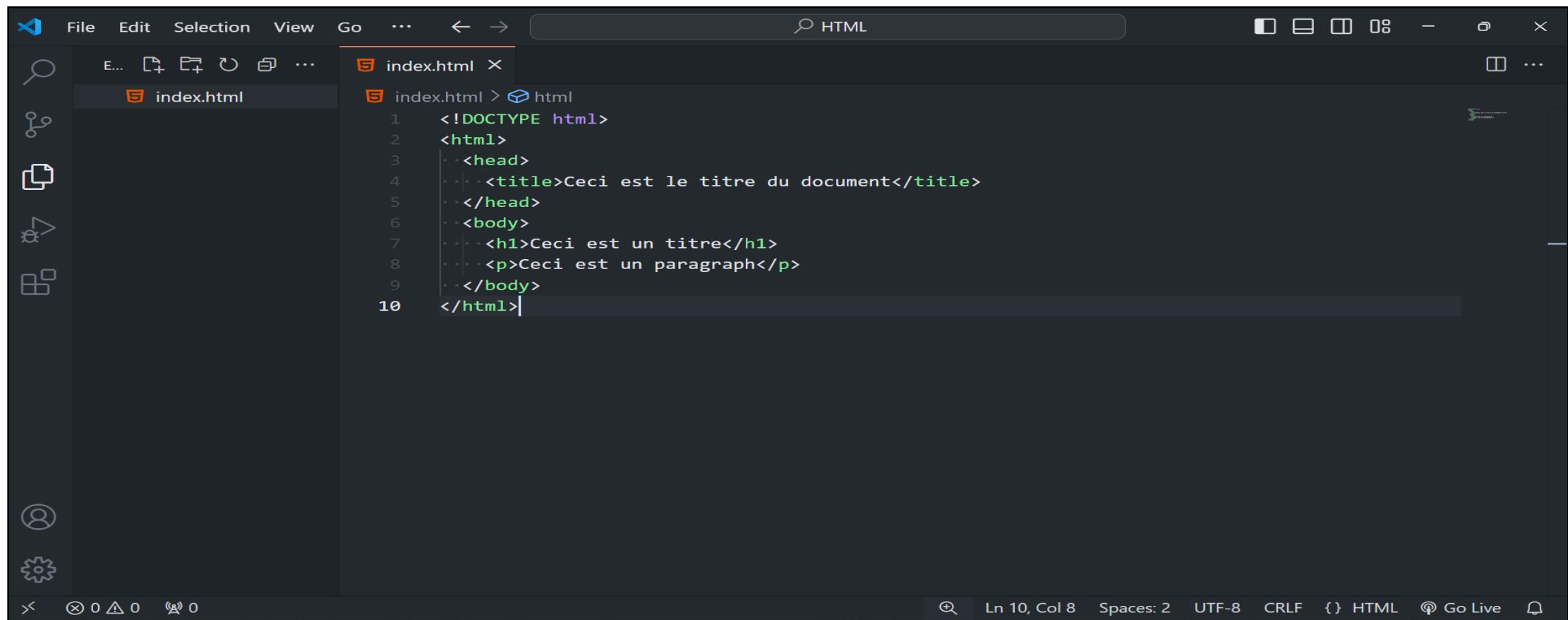
User Installer [x64](#) [Arm64](#)
System Installer [x64](#) [Arm64](#)

[.deb](#) [x64](#) [Arm32](#) [Arm64](#)
[.rpm](#) [x64](#) [Arm32](#) [Arm64](#)
[.tar.gz](#) [x64](#) [Arm32](#) [Arm64](#)

[.zip](#) [Intel chip](#) [Apple silicon](#) [Universal](#)
[CLI](#) [Intel chip](#) [Apple silicon](#)

Éditeurs HTML

Étape 2: ouvrez VS Code et créez un nouveau dossier. Dans ce dossier, créez un fichier nommé index.html (ou index.htm). Écrivez du code HTML dans ce fichier.



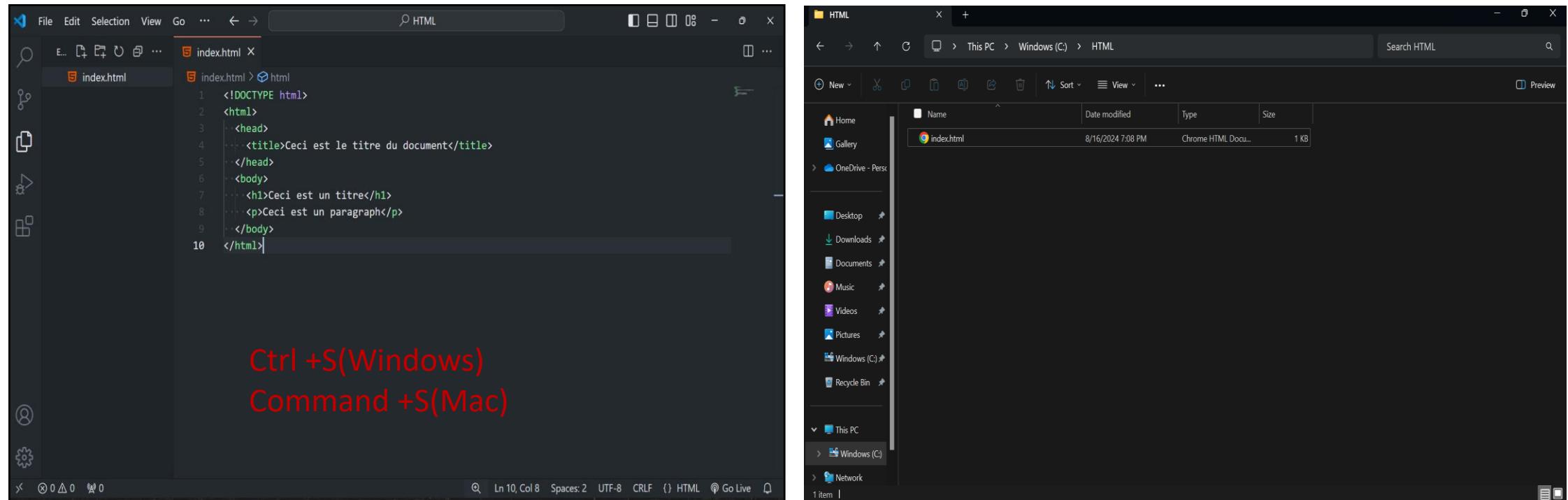
The screenshot shows the Visual Studio Code interface with a dark theme. On the left is the sidebar with icons for search, file, folder, and settings. The main area shows a file list with 'index.html' selected. The code editor displays the following HTML code:

```
1  <!DOCTYPE html>
2  <html>
3  ···<head>
4  ····<title>Ceci est le titre du document</title>
5  ···</head>
6  ···<body>
7  ····<h1>Ceci est un titre</h1>
8  ····<p>Ceci est un paragraphe</p>
9  ···</body>
10 </html>
```

The status bar at the bottom indicates the file is 'index.html', line 10, column 8, with spaces set to 2, encoding to UTF-8, and line endings to CRLF. It also shows icons for HTML, Go Live, and a bell.

Éditeurs HTML

Étape 3: Enregistrez le fichier HTML.



Éditeurs HTML

- **Étape 4:** Affichez la page dans votre navigateur en ouvrant le fichier index.html.



HTML de base

Documents HTML:

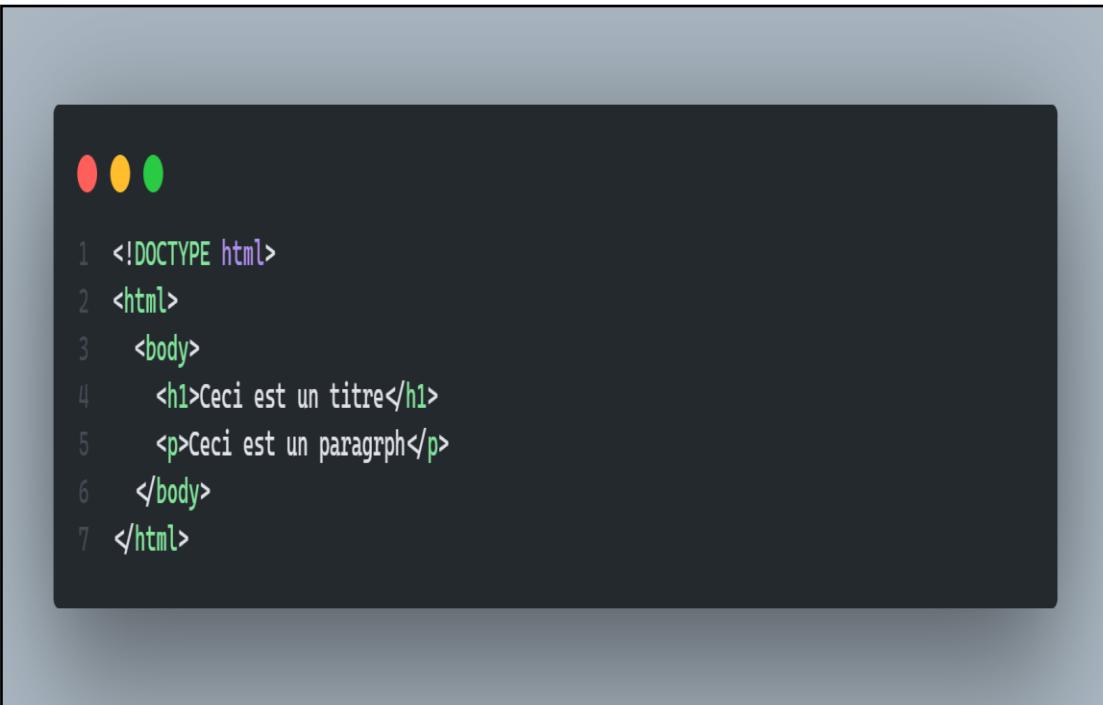
Tous les documents HTML doivent commencer par une déclaration de type de document:
`<!DOCTYPE html>`.

Le document HTML lui-même commence par `<html>` et se termine par `</html>`.

La partie visible du document HTML est comprise entre `<body>` et `</body>`.

Exemple:

HTML de base



A terminal window with a dark background and three colored icons (red, yellow, green) at the top. The window displays the following text:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Ceci est un titre</h1>
5     <p>Ceci est un paragrpah</p>
6   </body>
7 </html>
```



HTML de base

The <!DOCTYPE> Declaration:

La déclaration `<!DOCTYPE>` représente le type de document et aide les navigateurs à afficher correctement les pages Web.

Elle ne doit apparaître qu'une seule fois, en haut de la page (avant toute balise HTML).

La déclaration `<!DOCTYPE>` n'est pas sensible à la casse.

La déclaration `<!DOCTYPE>` pour HTML5 est:



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

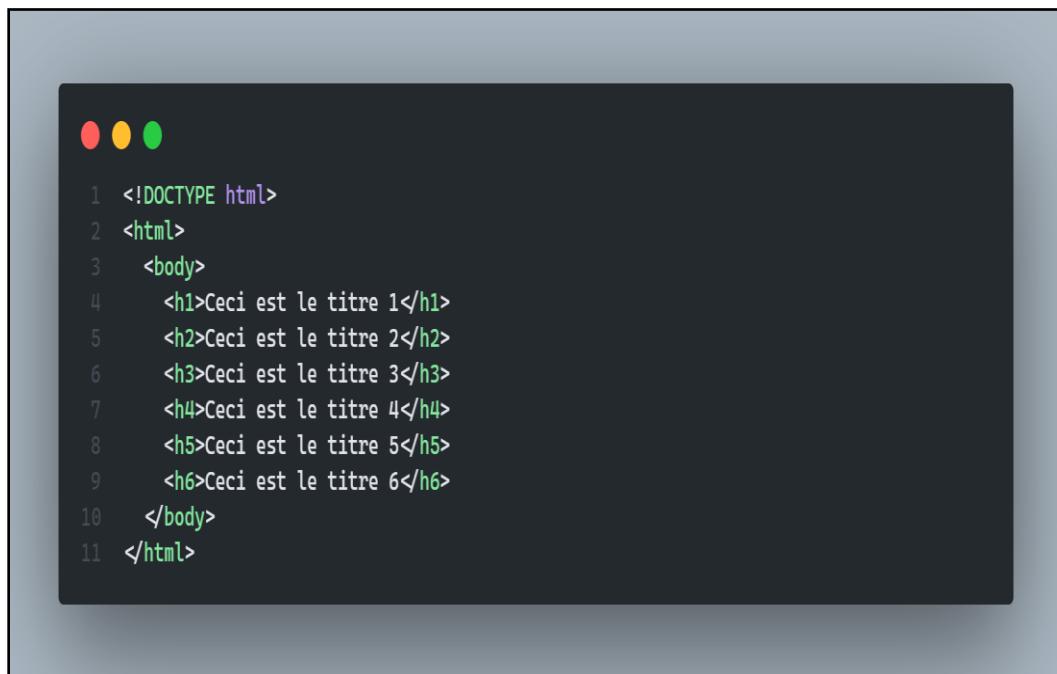
HTML de base

Titres HTML:

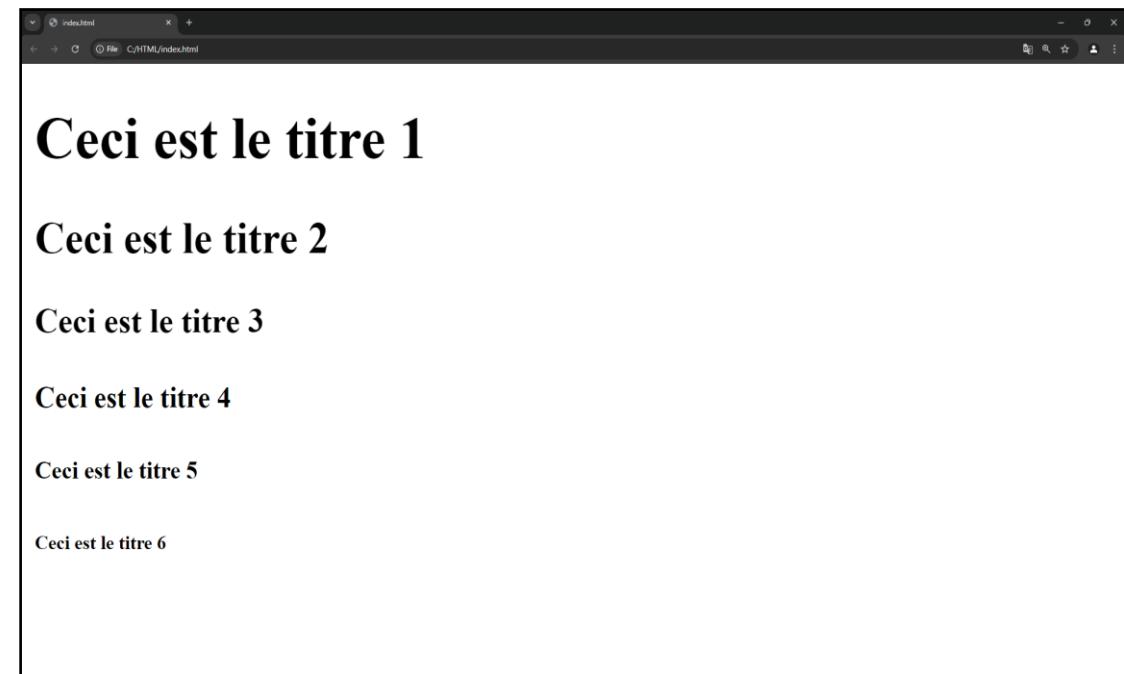
Les titres HTML sont définis avec les balises `<h1>` à `<h6>`.

`<h1>` définit le titre le plus important. `<h6>` définit le titre le moins important:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Ceci est le titre 1</h1>
5     <h2>Ceci est le titre 2</h2>
6     <h3>Ceci est le titre 3</h3>
7     <h4>Ceci est le titre 4</h4>
8     <h5>Ceci est le titre 5</h5>
9     <h6>Ceci est le titre 6</h6>
10    </body>
11  </html>
```

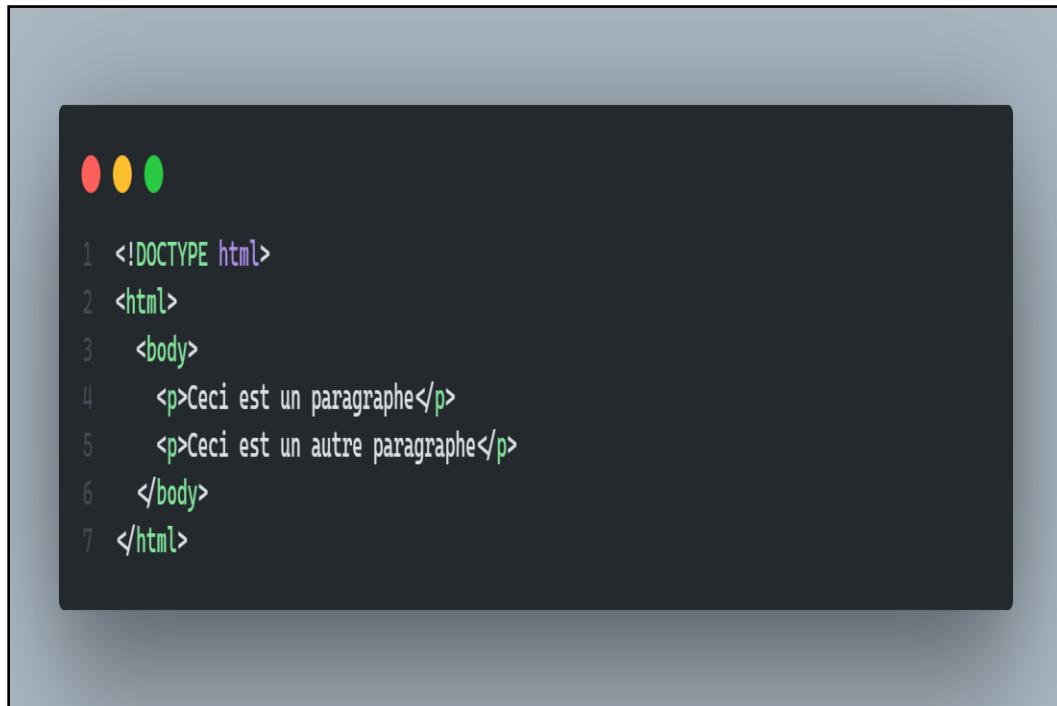


HTML de base

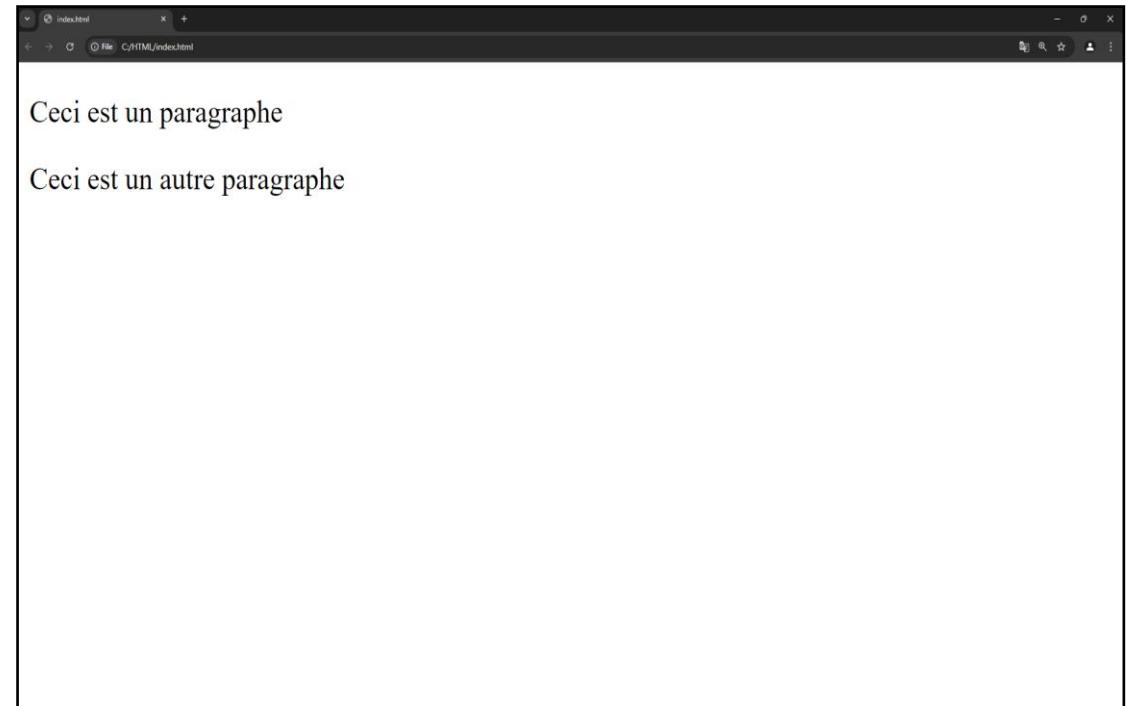
Paragraphes HTML:

Les paragraphes HTML sont définis avec la balise `<p>`:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ceci est un paragraphe</p>
5     <p>Ceci est un autre paragraphe</p>
6   </body>
7 </html>
```



HTML de base

Liens HTML:

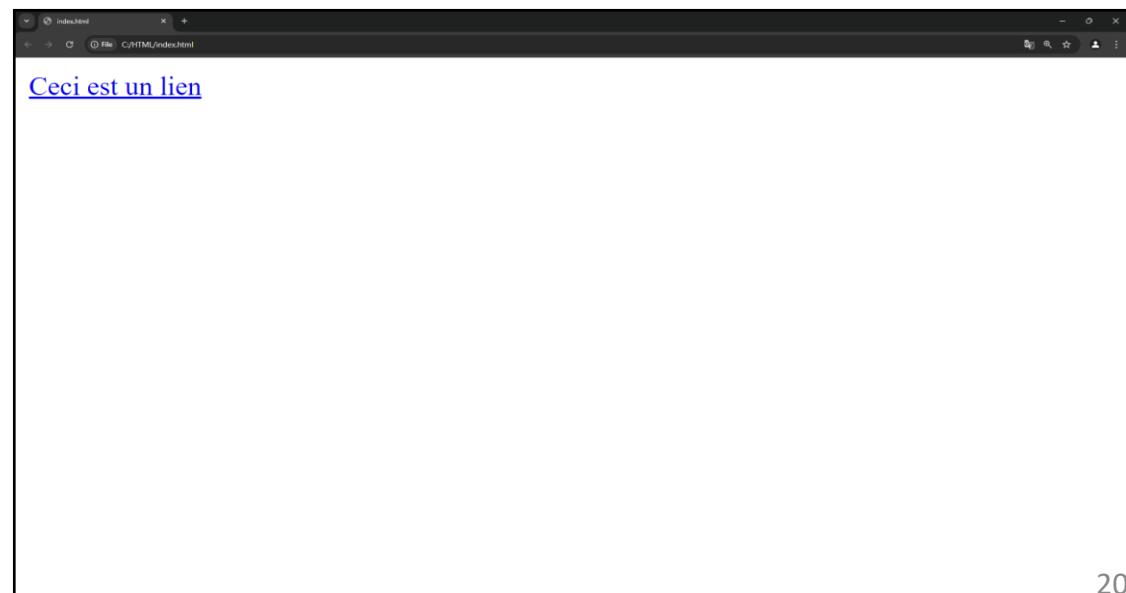
Les liens HTML sont définis avec la balise `<a>`:

La destination du lien est spécifiée dans l'attribut `href`.

Les attributs sont utilisés pour fournir des informations supplémentaires sur les éléments HTML.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <a href="https://www.w3schools.com/html/default.asp">Ceci est un lien</a>
5   </body>
6 </html>
```



HTML de base

HTML Images:

Les images HTML sont définies avec la balise ``.

Le fichier source (`src`), le texte alternatif (`alt`), `width` et `hauteur` sont fournis sous forme d'attributs:

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```



HTML de base

Afficher le code source HTML:

Cliquez sur CTRL + U dans une page HTML ou faites un clic droit sur la page et "sélectionnez Afficher la source de la page". Cela ouvrira un nouvel onglet contenant le code source HTML de la page.

Inspecter un élément HTML:

Cliquez avec le bouton droit de la souris sur un élément (ou une zone vide) et choisissez "Inspecter" pour voir de quoi sont composés les éléments (vous verrez à la fois le code HTML et le code CSS). Vous pouvez également modifier le code HTML ou le code CSS à la volée dans le panneau Éléments ou Styles qui s'ouvre.r un élément HTML:

Éléments HTML

Un élément HTML est défini par une balise de début, du contenu et une balise de fin.

Éléments HTML:

L'élément HTML comprend tout, de la balise de début à la balise de fin:

`<nomdebalise>Le contenu va ici...</nomdebalise>`

Exemples d'éléments HTML:

`<h1>Ceci est un titre</h1>`

`<p>Ceci est un paragraphe<\p>`

Balise de début	Contenu de l'élément	Balise de fin
<code><h1></code>	Ceci est un titre	<code></h1></code>
<code><p></code>	Ceci est un paragraphe	<code></p></code>
<code>
</code>	aucun	aucune

Remarque: certains éléments HTML n'ont aucun contenu (comme l'élément `
`). Ces éléments sont appelés éléments vides. Les éléments vides n'ont pas de balise de fin!

Éléments HTML

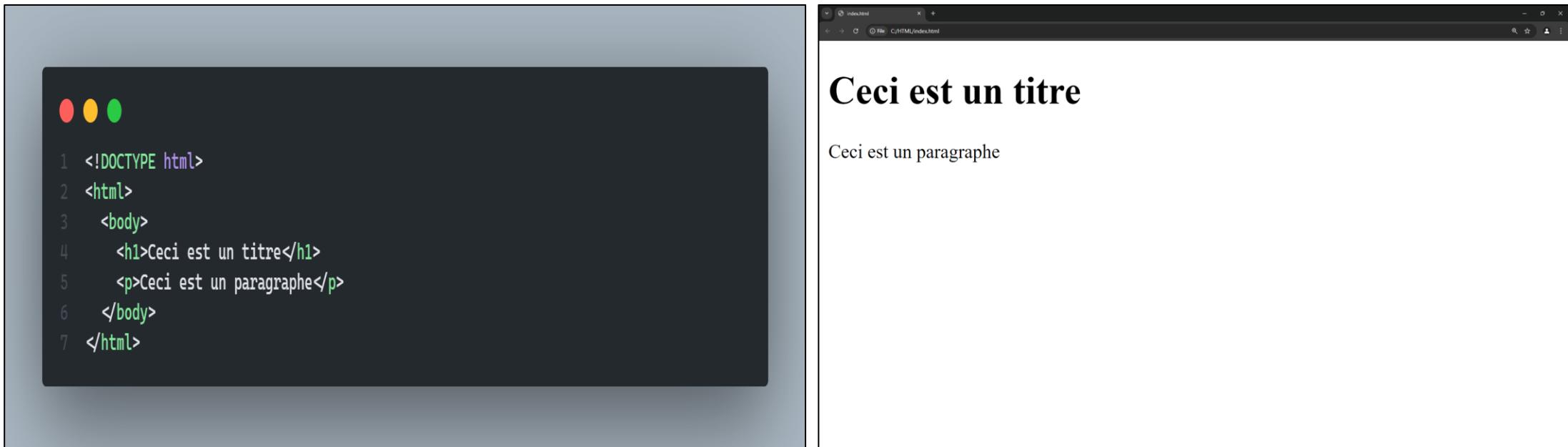
Éléments HTML imbriqués:

Les éléments HTML peuvent être imbriqués (ce qui signifie que les éléments peuvent contenir d'autres éléments).

Tous les documents HTML sont constitués d'éléments HTML imbriqués.

L'exemple suivant contient quatre éléments HTML (`<html>`, `<body>`, `<h1>` et `<p>`):

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Ceci est un titre</h1>
5     <p>Ceci est un paragraphe</p>
6   </body>
7 </html>
```

Éléments HTML

Exemple expliqué:

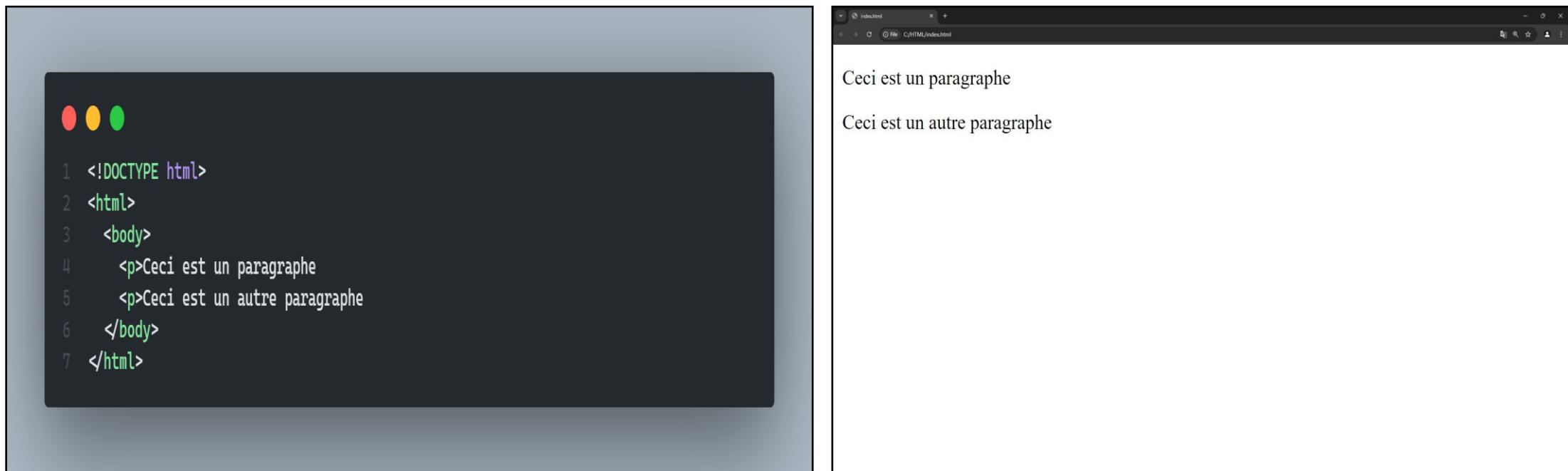
- L'élément `<html>` est l'élément racine et il définit l'ensemble du document HTML.
- Il possède une balise de début `<html>` et une balise de fin `</html>`.
- Ensuite, à l'intérieur de l'élément `<html>`, il y a un élément `<body>`.
- L'élément `<body>` définit le corps du document.
- Il comporte une balise de début `<body>` et une balise de fin `</body>`.
- Ensuite, à l'intérieur de l'élément `<body>`, il y a deux autres éléments: `<h1>` et `<p>`.
- L'élément `<h1>` définit un titre.
- Il possède une balise de début `<h1>` et une balise de fin `</h1>`: L'élément `<h1>` définit un titre.
- Il possède une balise de début `<h1>` et une balise de fin `</h1>`.
- L'élément `<p>` définit un paragraphe.
- Il possède une balise de début `<p>` et une balise de fin `</p>`.

Éléments HTML

Ne sautez jamais la balise de fin:

Certains éléments HTML s'afficheront correctement, même si vous oubliez la balise de fin:

Exemple:



The image shows two side-by-side windows. On the left is a terminal window with a dark background and light text, displaying the following code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ceci est un paragraphe
5     <p>Ceci est un autre paragraphe
6   </body>
7 </html>
```

On the right is a screenshot of a web browser window titled "index.html". The address bar shows "C:\HTML\index.html". The page content is:

Ceci est un paragraphe
Ceci est un autre paragraphe

Mais ne vous fiez jamais à cela! Des résultats inattendus et des erreurs peuvent survenir si vous oubliez la balise de fin!

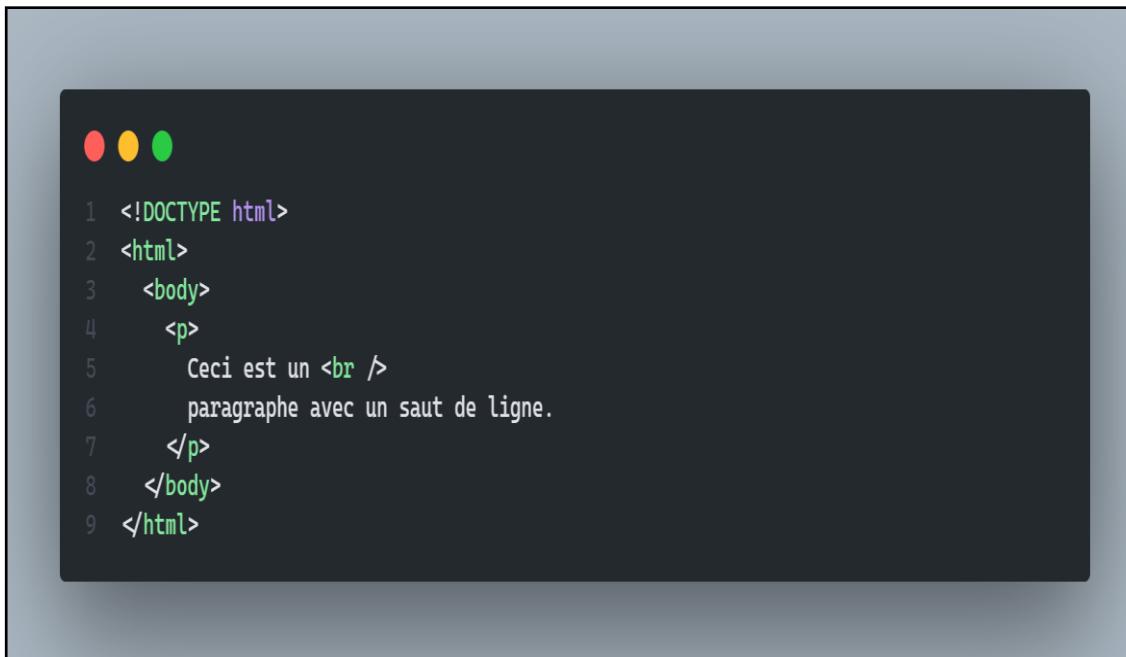
Éléments HTML

Éléments HTML vides:

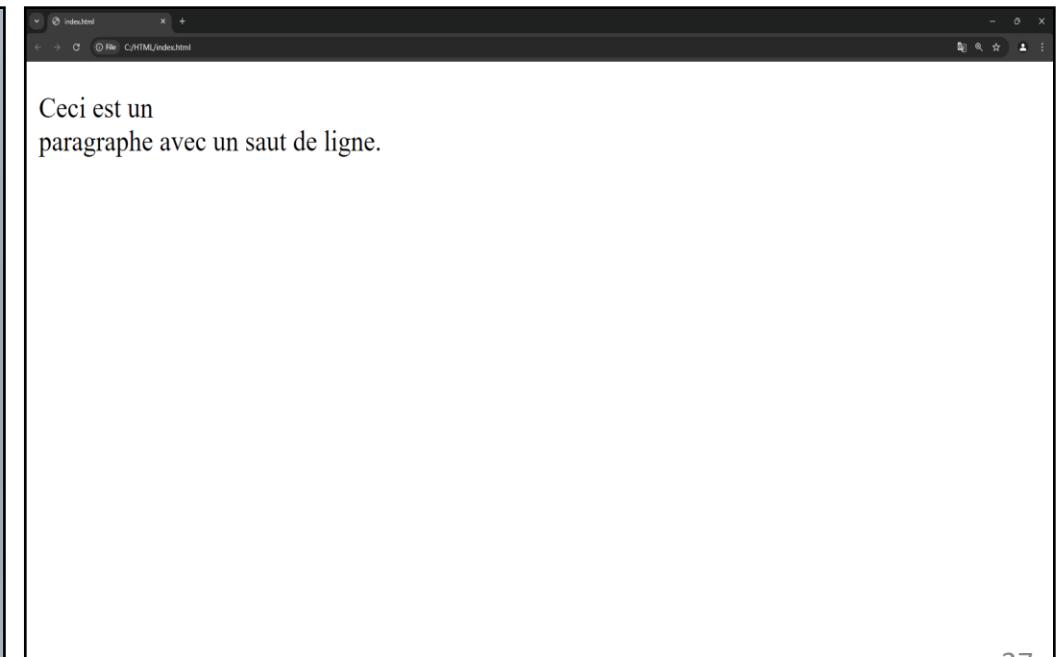
Les éléments HTML sans contenu sont appelés éléments vides.

La balise `
` définit un saut de ligne et est un élément vide sans balise de fermeture:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>
5       Ceci est un <br />
6       paragraphe avec un saut de ligne.
7     </p>
8   </body>
9 </html>
```



Éléments HTML

Le HTML n'est pas sensible à la casse:

Les balises HTML ne sont pas sensibles à la casse:<P> signifie la même chose que <p>.

La norme HTML n'exige pas de balises minuscules, mais le W3C recommande les minuscules en HTML et exige les minuscules pour les types de documents plus stricts comme le XHTML.

Attributs HTML

Les attributs HTML fournissent des informations supplémentaires sur les éléments HTML.

Attributs HTML:

- Tous les éléments HTML peuvent avoir **des attributs**.
- Les attributs fournissent **des informations supplémentaires** sur les éléments.
- Les attributs sont toujours spécifiés dans **la balise de début**.
- Les attributs sont généralement présentés sous forme de paires nom/valeur telles que: **nom="valeur"**.

Attributs HTML

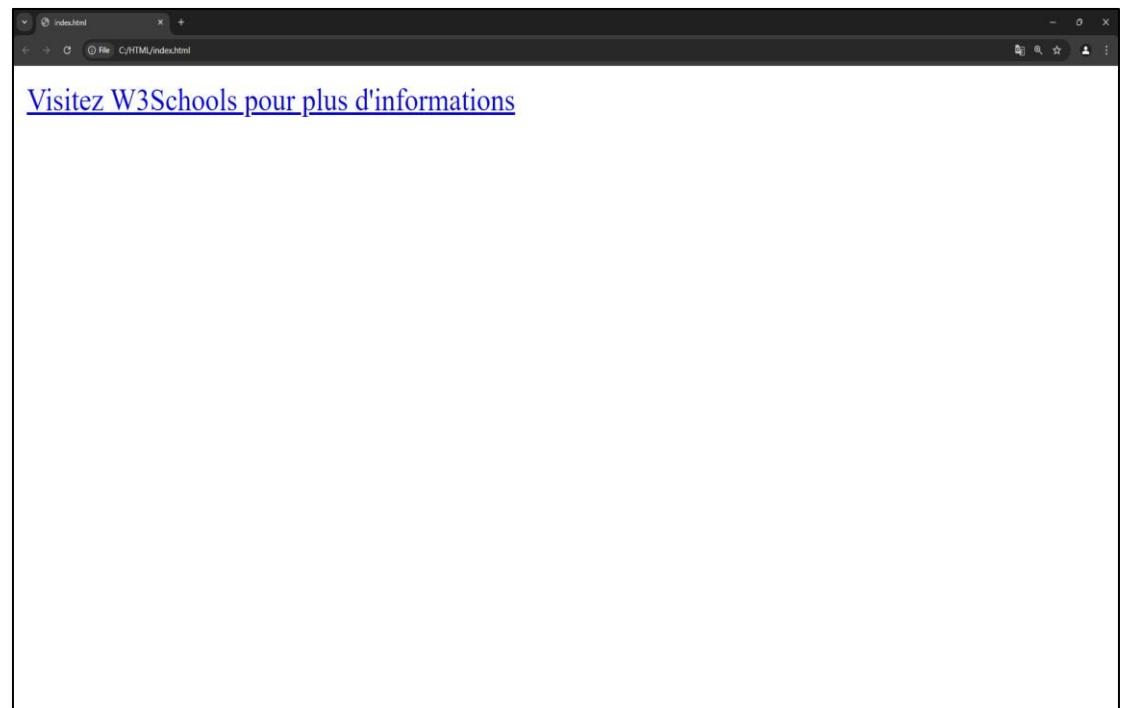
L'attribut href:

La balise `<a>` définit un lien hypertexte. L'attribut `href` spécifie l'URL de la page vers laquelle le lien pointe:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <a href="https://www.w3schools.com/html/default.asp">Visitez W3Schools pour plus d'informations</a>
5   </body>
6 </html>
```



Attributs HTML

L'attribut src:

La balise `` permet d'intégrer une image dans une page HTML. L'attribut `src` spécifie le chemin d'accès à l'image à afficher:

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```



Attributs HTML

Il existe deux manières de spécifier l'URL dans l'attribut `src`:

1. **URL_absolue:** Liens vers une image externe hébergée sur un autre site Web. Exemple: `src="https://www.exemple/images/omage_1.jpg"`.
2. **URL relative:** Liens vers une image hébergée sur le site Web. Ici, l'URL n'inclut pas le nom de domaine. Si l'URL commence sans barre oblique, elle sera relative à la page actuelle. Exemple: `src="HTML-icon.png"`. Si l'URL commence par une barre oblique, elle sera relative au domaine. Exemple: `src="/images/ HTML-icon.png"`.

Remarques: les images externes peuvent être protégées par des droits d'auteur. Si vous n'obtenez pas l'autorisation de les utiliser, vous risquez de violer les lois sur les droits d'auteur. De plus, vous ne pouvez pas contrôler les images externes ; elles peuvent être soudainement supprimées ou modifiées.

Astuce: il est presque toujours préférable d'utiliser des URL relatives. Elles ne seront pas endommagées si vous changez de domaine.

Attributs HTML

Les attributs width et height:

La balise `` doit également contenir les attributs `width` et `height`, qui spécifient la largeur et la hauteur de l'image (en pixels):

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```

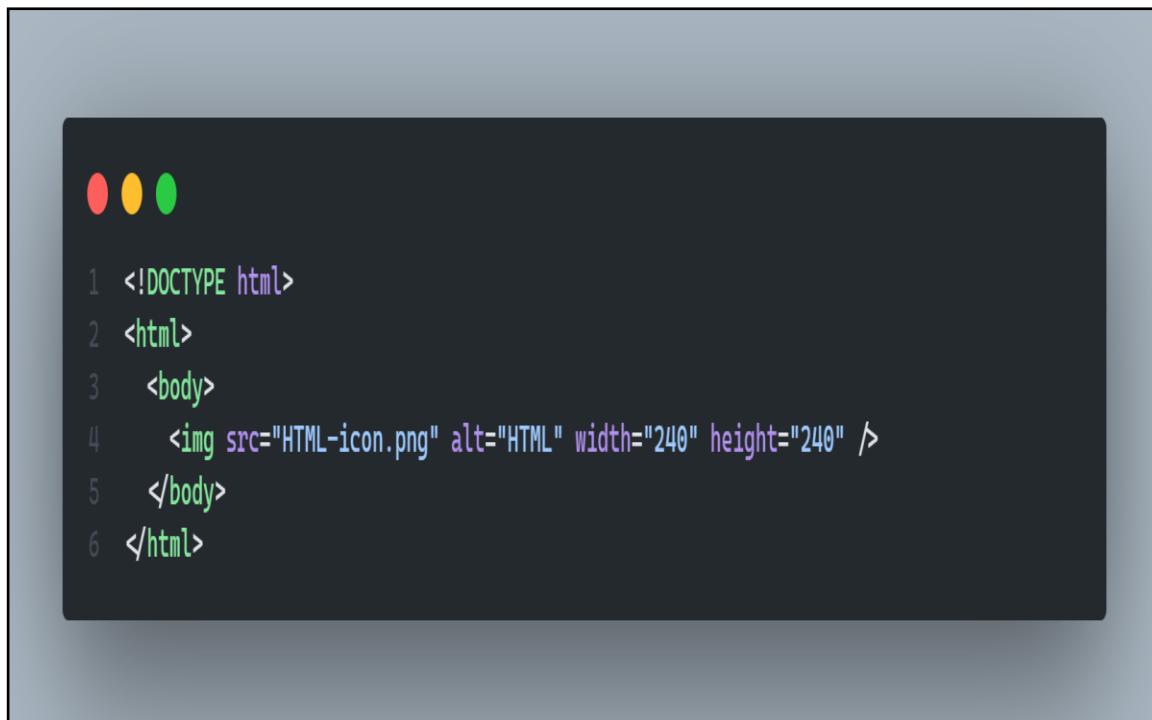


Attributs HTML

L'attribut alt:

L'attribut **alt** requis pour la balise **** spécifie un texte alternatif pour une image, si l'image ne peut pas être affichée pour une raison quelconque. Cela peut être dû à une connexion lente, à une erreur dans l'attribut **src** ou à l'utilisation d'un lecteur d'écran par l'utilisateur.

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```



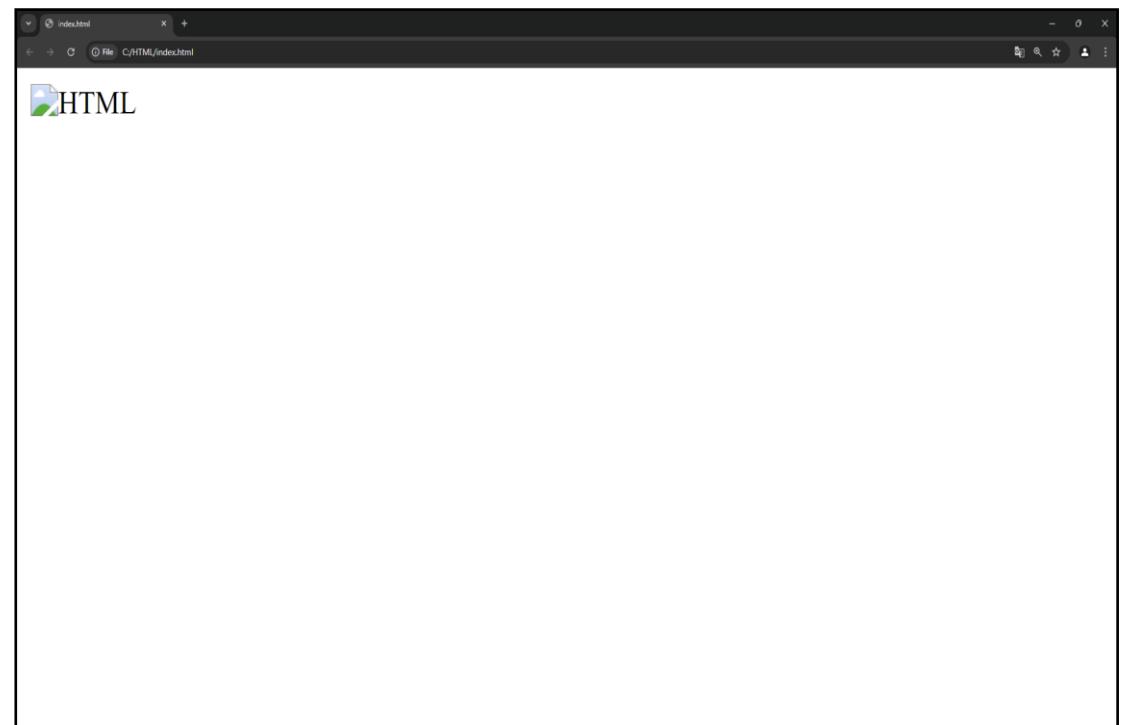
Attributs HTML

Voyez ce qui se passe si nous essayons d'afficher une image qui n'existe pas:

Exemple:



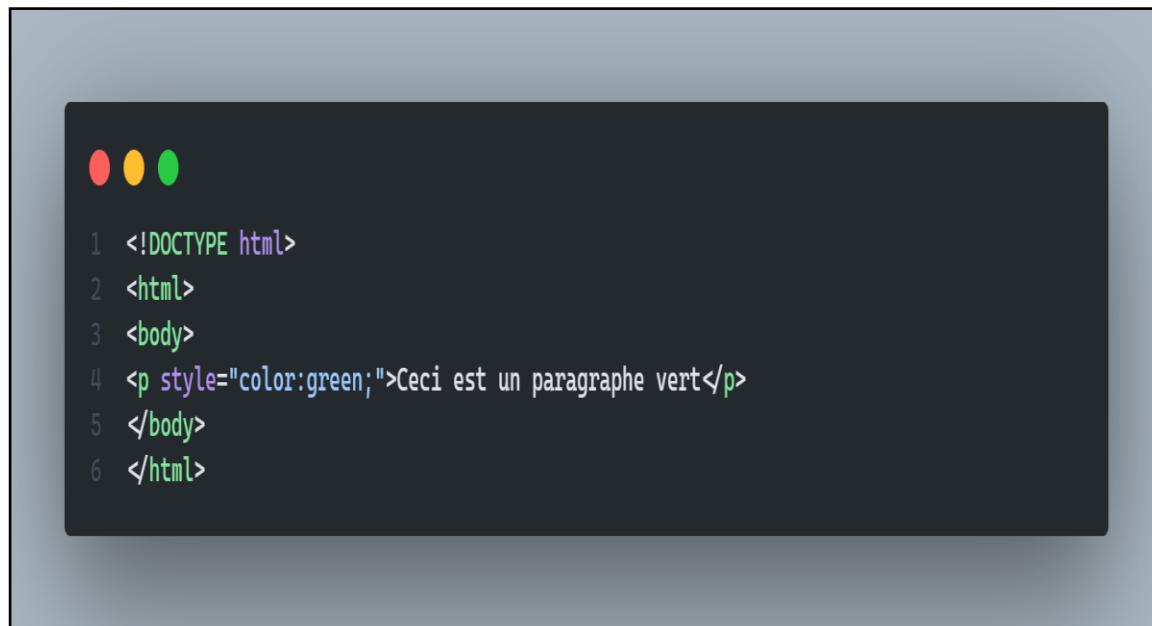
```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```



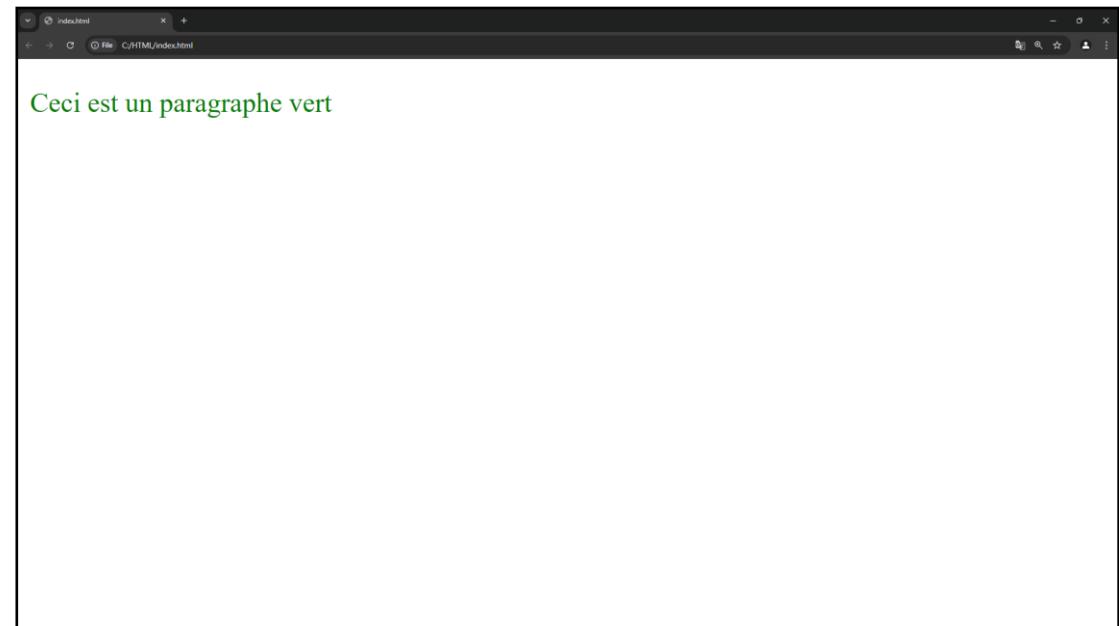
Attributs HTML

L'attribut style:

L'attribut **style** est utilisé pour ajouter des styles à un élément, tels que la couleur, la police, la taille, etc.



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p style="color:green;">Ceci est un paragraphe vert</p>
5 </body>
6 </html>
```



Attributs HTML

L'attribut lang:

Vous devez toujours inclure l'attribut **lang** dans la balise `<html>` pour déclarer la langue de la page Web. Cela a pour but d'aider les moteurs de recherche et les navigateurs.

L'exemple suivant spécifie le français comme langue:

Exemple:



```
1  <!DOCTYPE html>
2  <html lang="fr">
3      <body></body>
4  </html>
```

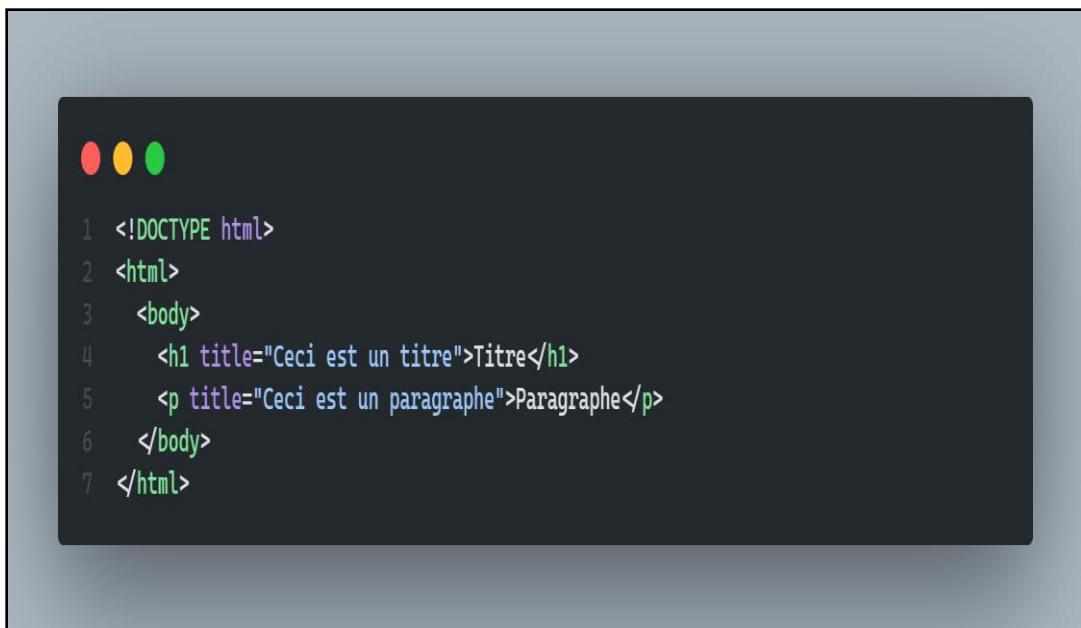
Attributs HTML

L'attribut title:

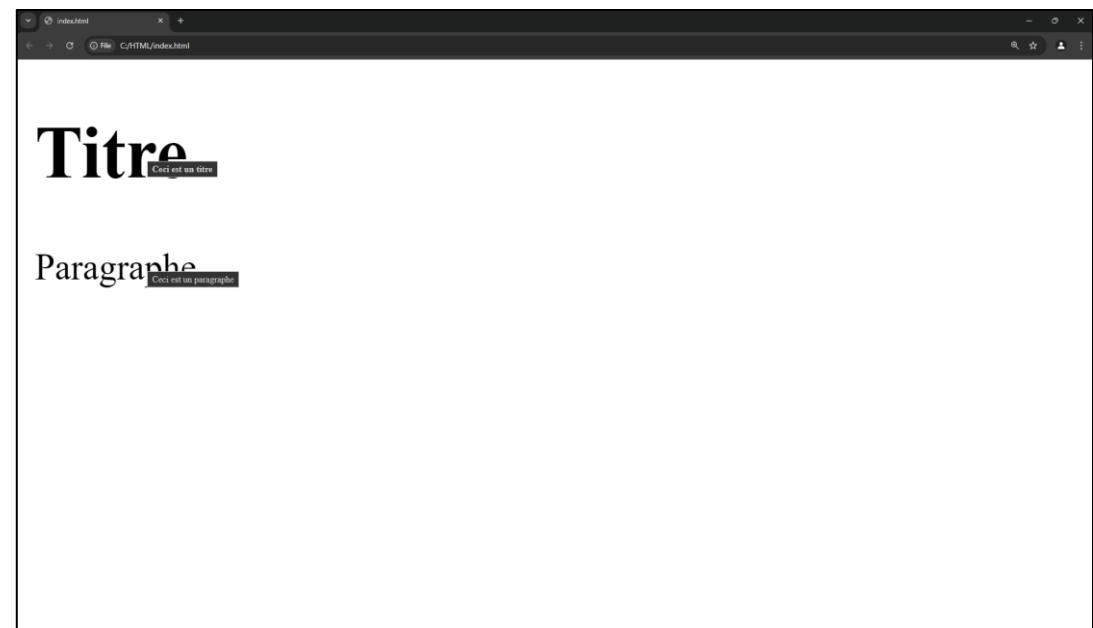
L'attribut **title** définit des informations supplémentaires sur un élément.

La valeur de l'attribut title s'affiche sous forme d'info-bulle lorsque vous passez la souris sur l'élément

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1 title="Ceci est un titre">Titre</h1>
5     <p title="Ceci est un paragraphe">Paragraphe</p>
6   </body>
7 </html>
```



Attributs HTML

Toujours utiliser l'attribut minuscule:

La norme HTML n'exige pas de noms d'attributs en minuscules.

L'attribut `title` (et tous les autres attributs) peut être écrit en majuscules ou en minuscules, comme `title` ou `TITLE`.

Cependant, la meilleure pratique consiste à utiliser des attributs minuscules dans HTML et exige des attributs minuscules pour les types de documents plus stricts comme XHTML.

Attributs HTML

Toujours citer les valeurs d'attribut:

La norme HTML n'exige pas de guillemets autour des valeurs d'attribut.

Cependant, le W3C recommande les guillemets en HTML et les exige pour les types de documents plus stricts comme le XHTML.

Exemple bien:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p title="paragraphe">Paragraphe</p>
5   </body>
6 </html>
```

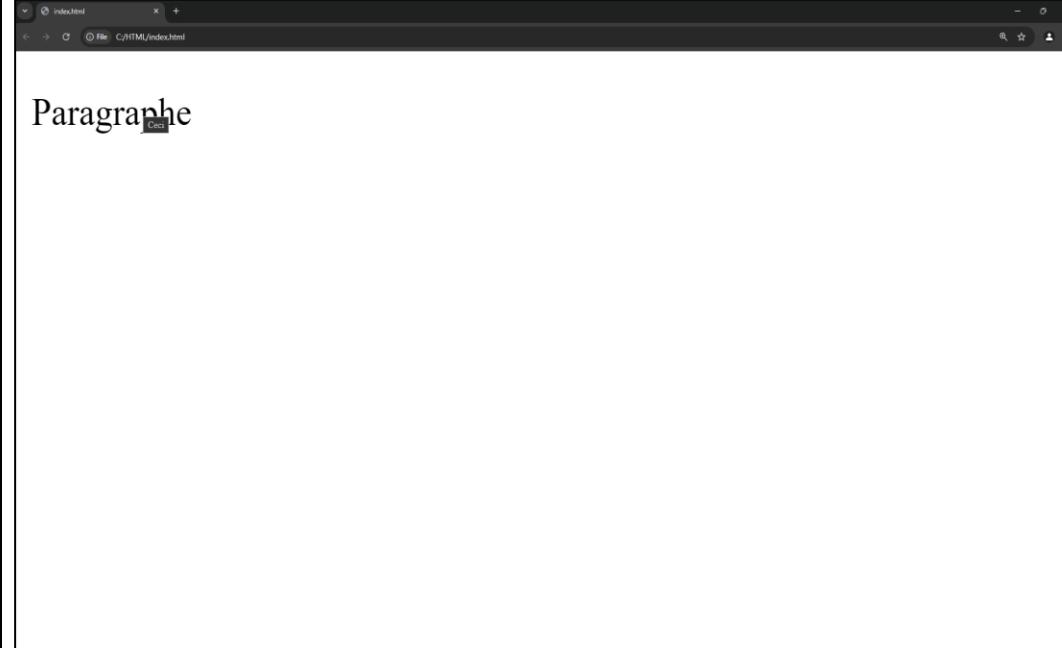
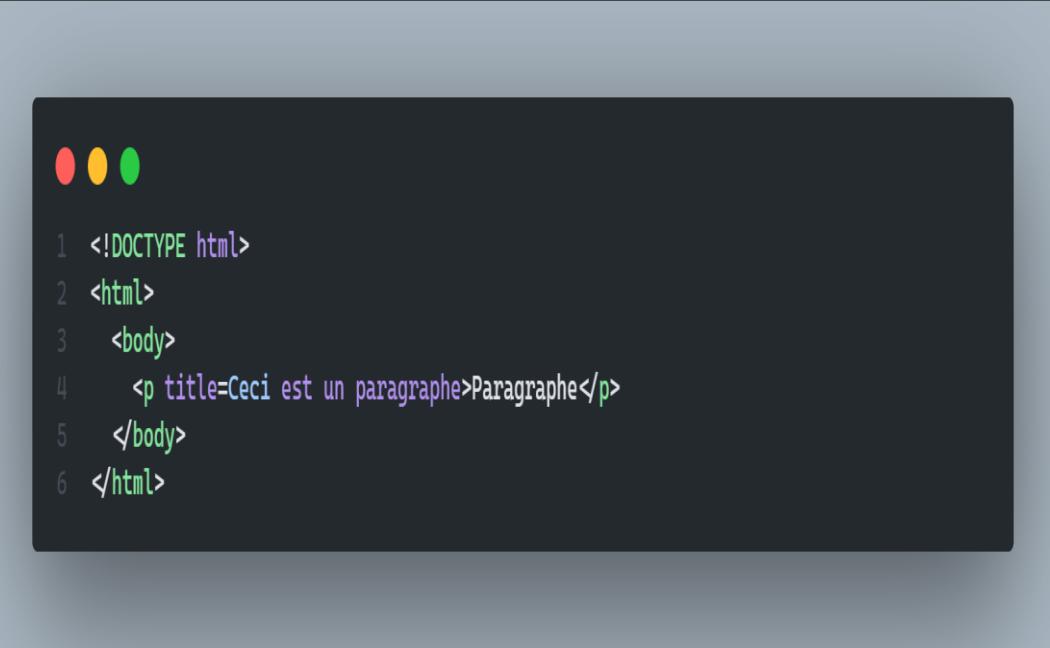
Exemple mauvaise:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p title=paragraphe>Paragraphe</p>
5   </body>
6 </html>
```

Attributs HTML

Il est parfois nécessaire d'utiliser des guillemets. Cet exemple n'affiche pas correctement l'attribut title, car il contient un espace:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p title=Ceci est un paragraphe>Paragraphe</p>
5   </body>
6 </html>
```

The terminal window shows the following code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p title=Ceci est un paragraphe>Paragraphe</p>
5   </body>
6 </html>
```

The browser window shows the output:

Paragraphe

Attributs HTML

Guillemets simples ou doubles?

- Les guillemets doubles autour des valeurs d'attribut sont les plus courants en HTML, mais les guillemets simples peuvent également être utilisés.
- Dans certaines situations, lorsque la valeur d'attribut contient elle-même des guillemets doubles, il est nécessaire d'utiliser des guillemets simples:

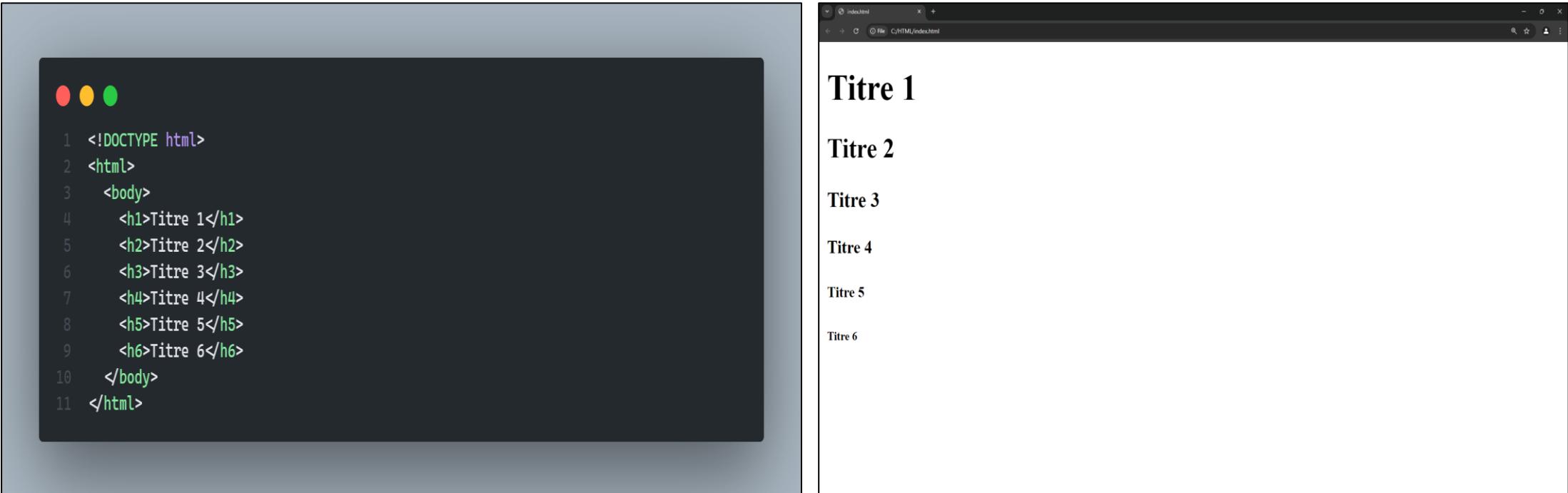
```
● ● ●
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p title="Ceci est un 'paragraphe'">Paragraphe</p>
5   </body>
6 </html>
```

Ou:

```
● ● ●
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p title='Ceci est un "paragraphe"'>Paragraphe</p>
5   </body>
6 </html>
```

Titres HTML

Les en-têtes HTML sont des titres ou des sous-titres que vous souhaitez afficher sur une page Web.



The image shows a terminal window on the left and a browser window on the right. The terminal window displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Titre 1</h1>
5     <h2>Titre 2</h2>
6     <h3>Titre 3</h3>
7     <h4>Titre 4</h4>
8     <h5>Titre 5</h5>
9     <h6>Titre 6</h6>
10   </body>
11 </html>
```

The browser window shows the rendered HTML with the following titles:

- Titre 1
- Titre 2
- Titre 3
- Titre 4
- Titre 5
- Titre 6

Titres HTML

Titres HTML:

Les titres HTML sont définis avec les balises `<h1>` à `<h6>`.

`<h1>` définit le titre le plus important. `<h6>` définit le titre le moins important.

Remarque: les navigateurs ajoutent automatiquement un espace blanc (une marge) avant et après un titre.

Les titres sont importants:

Les moteurs de recherche utilisent les titres pour indexer la structure et le contenu de vos pages Web.

Les utilisateurs parcourent souvent une page en fonction de ses titres. Il est important d'utiliser des titres pour montrer la structure du document.

Les titres `<h1>` doivent être utilisés pour les titres principaux, suivis des titres `<h2>`, puis des titres moins importants `<h3>`, et ainsi de suite.

Remarque: utilisez les titres HTML uniquement pour les titres. N'utilisez pas de titres pour mettre du texte en GRAND ou en gras.

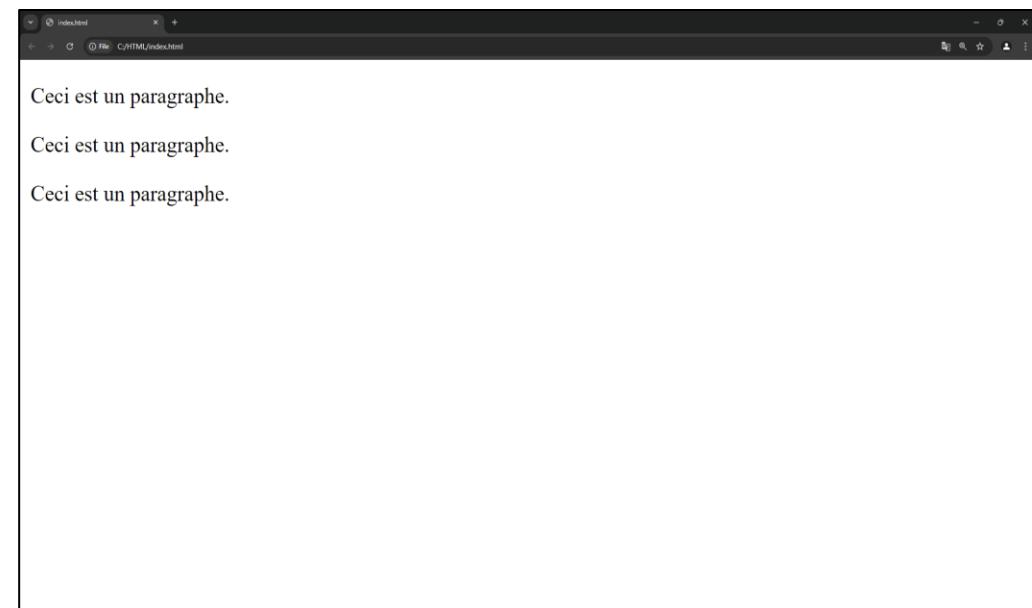
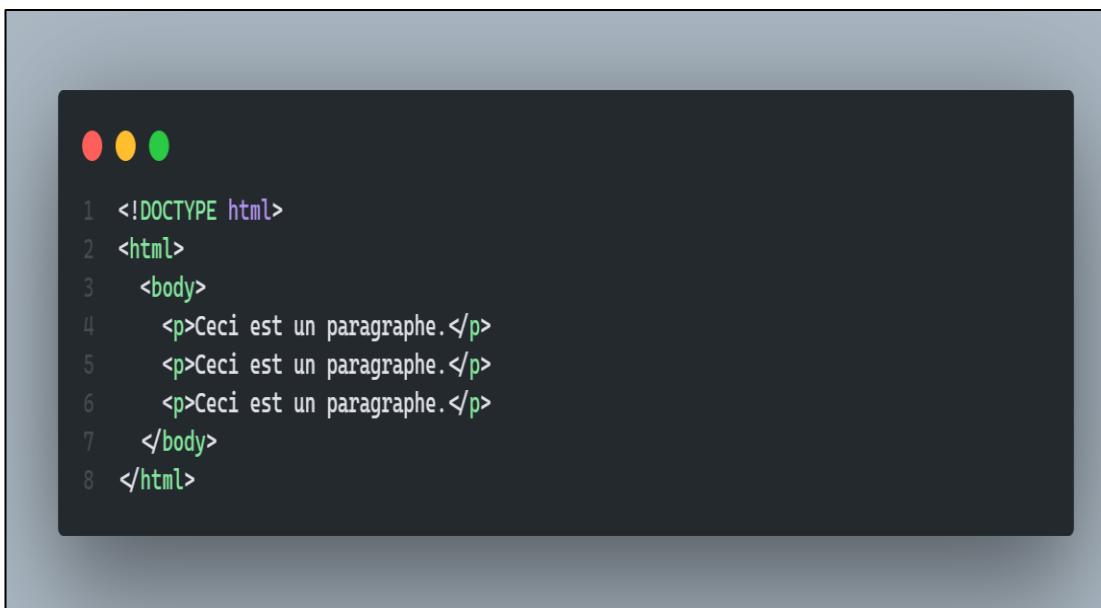
Paragraphes HTML

Un paragraphe commence toujours sur une nouvelle ligne et constitue généralement un bloc de texte.

Paragraphes HTML:

L'élément HTML `<p>` définit un paragraphe.

Un paragraphe commence toujours sur une nouvelle ligne et les navigateurs ajoutent automatiquement un espace blanc (une marge) avant et après un paragraphe.



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ceci est un paragraphe.</p>
5     <p>Ceci est un paragraphe.</p>
6     <p>Ceci est un paragraphe.</p>
7   </body>
8 </html>
```

Ceci est un paragraphe.
Ceci est un paragraphe.
Ceci est un paragraphe.

Paragraphes HTML

Affichage HTML:

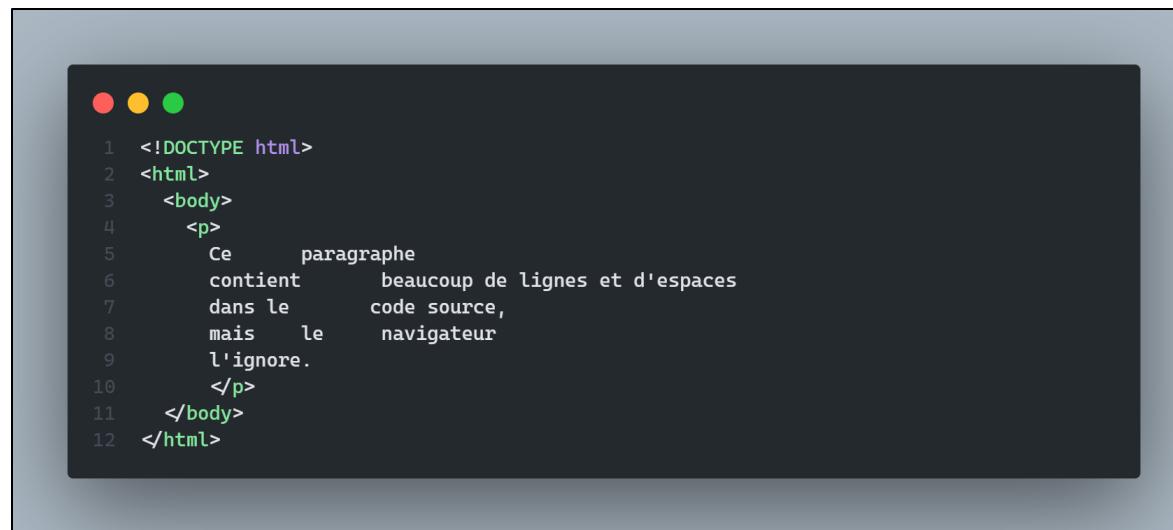
Vous ne pouvez pas être sûr de la manière dont le code HTML s'affichera.

Les écrans grands ou petits et les fenêtres redimensionnées créeront des résultats différents.

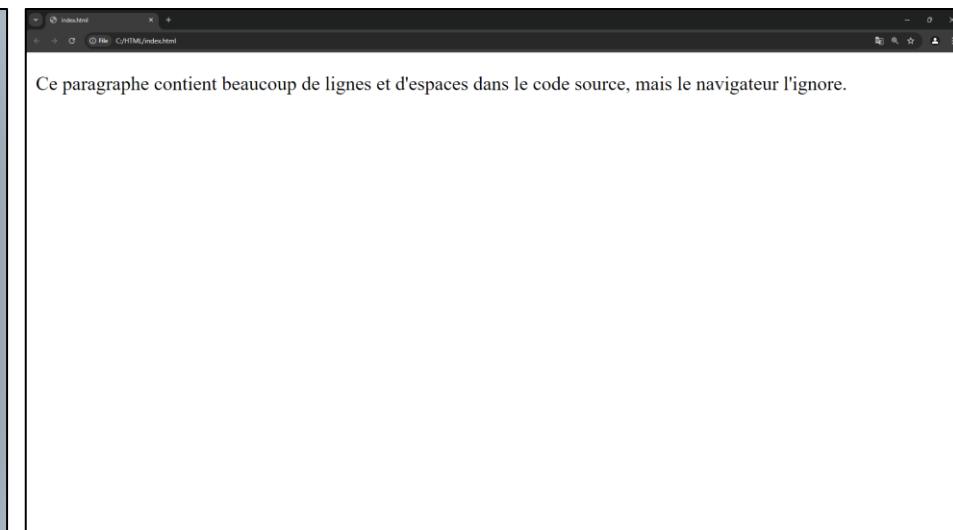
Avec HTML, vous ne pouvez pas modifier l'affichage en ajoutant des espaces ou des lignes supplémentaires dans votre code HTML.

Le navigateur supprimera automatiquement les espaces et les lignes supplémentaires lors de l'affichage de la page:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>
5       Ce      paragraphe
6       contient      beaucoup de lignes et d'espaces
7       dans le      code source,
8       mais      le    navigateur
9       l'ignore.
10    </p>
11  </body>
12 </html>
```



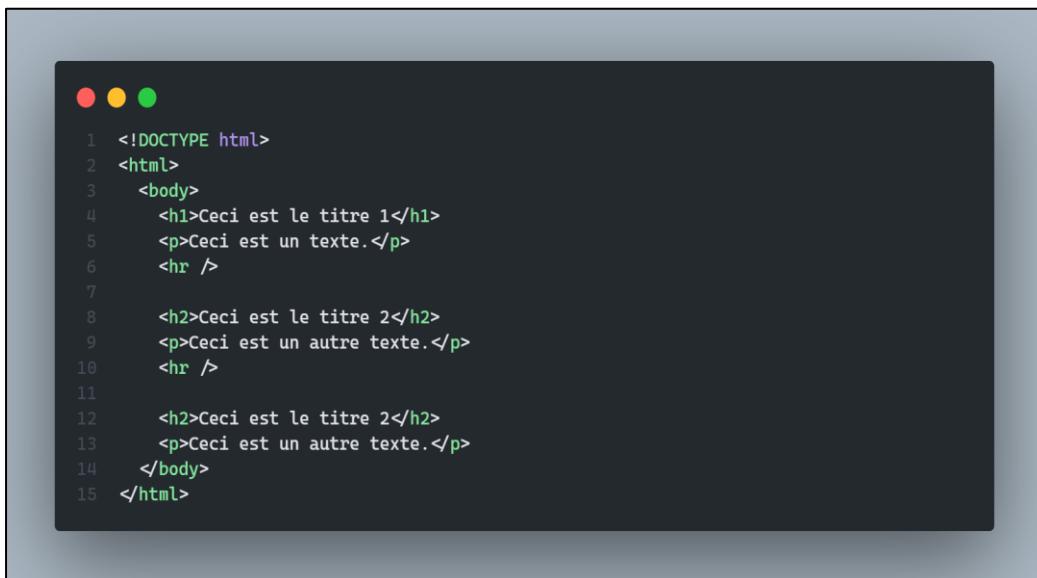
Paragraphes HTML

Règles horizontales HTML:

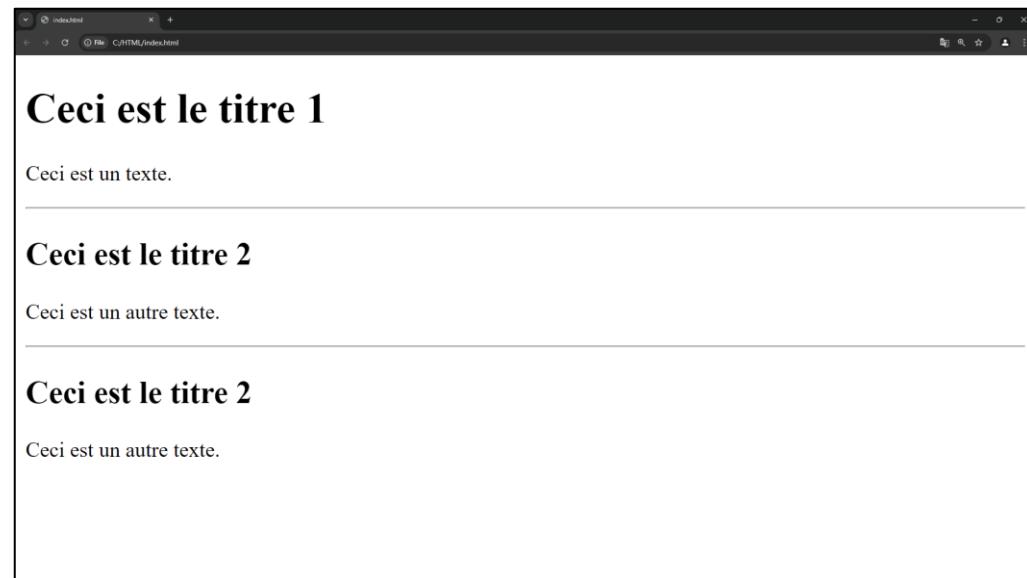
La balise `<hr>` définit une rupture thématique dans une page HTML et est le plus souvent affichée sous la forme d'une règle horizontale.

L'élément `<hr>` est utilisé pour séparer le contenu (ou définir un changement) dans une page HTML :

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Ceci est le titre 1</h1>
5     <p>Ceci est un texte.</p>
6     <hr />
7
8     <h2>Ceci est le titre 2</h2>
9     <p>Ceci est un autre texte.</p>
10    <hr />
11
12    <h2>Ceci est le titre 2</h2>
13    <p>Ceci est un autre texte.</p>
14  </body>
15 </html>
```



La balise `<hr>` est une balise vide, ce qui signifie qu'elle n'a pas de balise de fin.

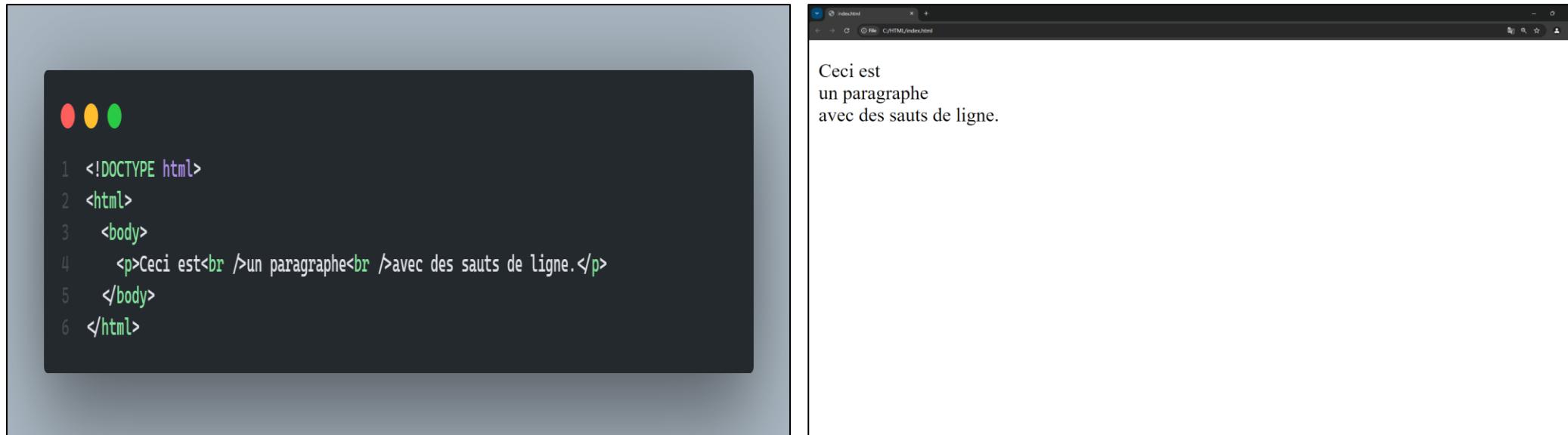
Paragraphes HTML

Sauts de ligne HTML:

L'élément HTML `
` définit un saut de ligne.

Utilisez `
` si vous souhaitez un saut de ligne (une nouvelle ligne) sans démarrer un nouveau paragraphe :

Exemple:



The image shows a comparison between a code editor and a web browser. On the left, a code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ceci est<br />un paragraphe<br />avec des sauts de ligne.</p>
5   </body>
6 </html>
```

On the right, a web browser window titled "index.html" shows the rendered output:

Ceci est
un paragraphe
avec des sauts de ligne.

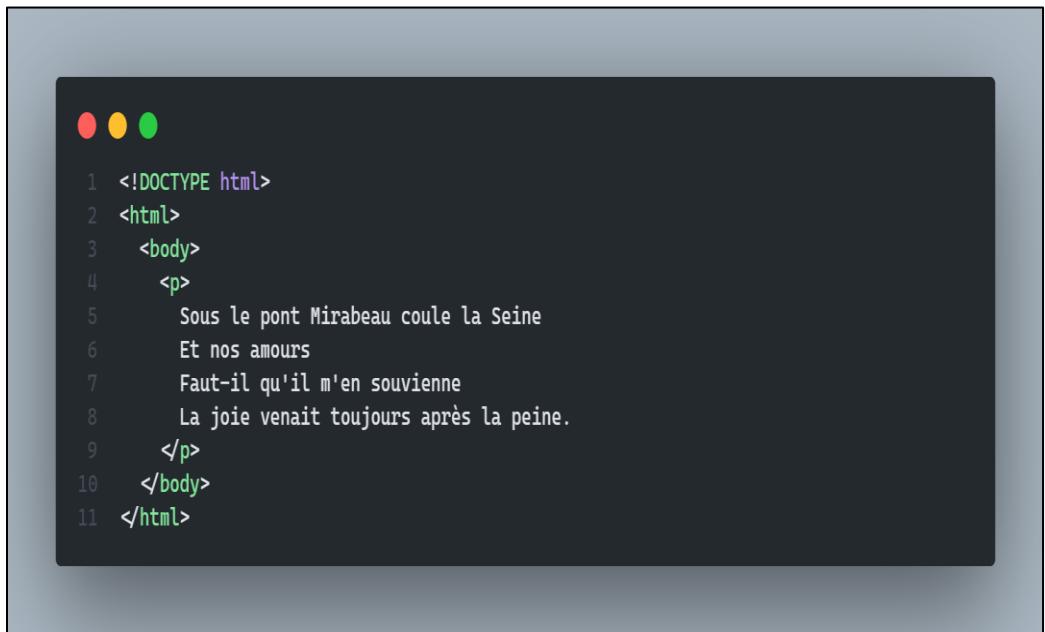
La balise `
` est une balise vide, ce qui signifie qu'elle n'a pas de balise de fin.

Paragraphes HTML

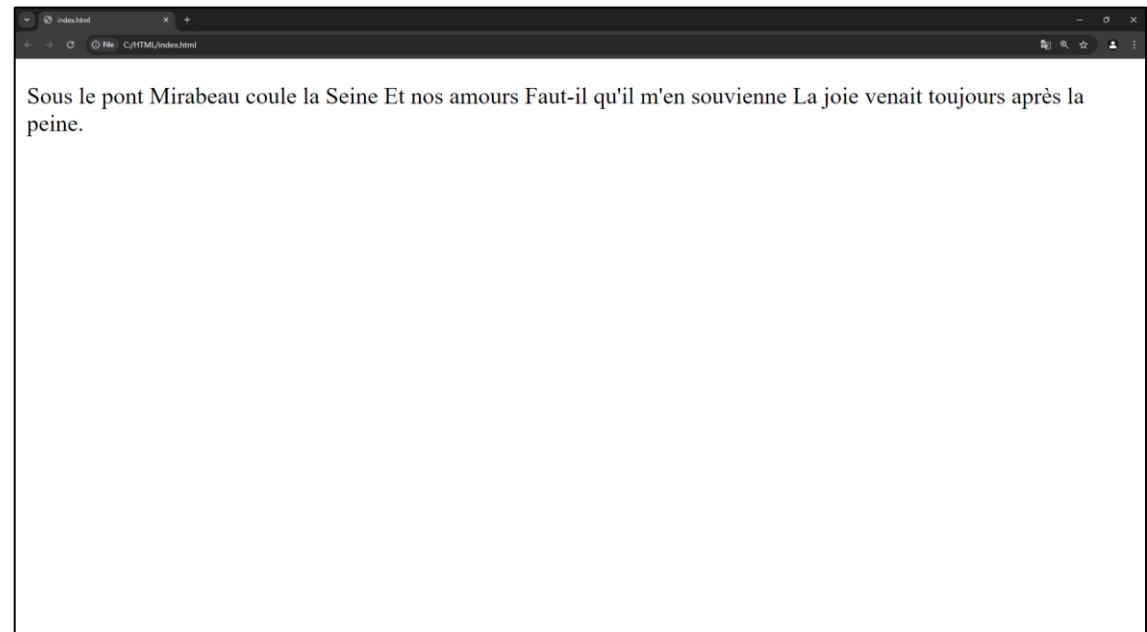
The Poem Problem:

Ce poème s'affichera sur une seule ligne:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>
5       Sous le pont Mirabeau coule la Seine
6       Et nos amours
7       Faut-il qu'il m'en souvienne
8       La joie venait toujours après la peine.
9     </p>
10    </body>
11  </html>
```



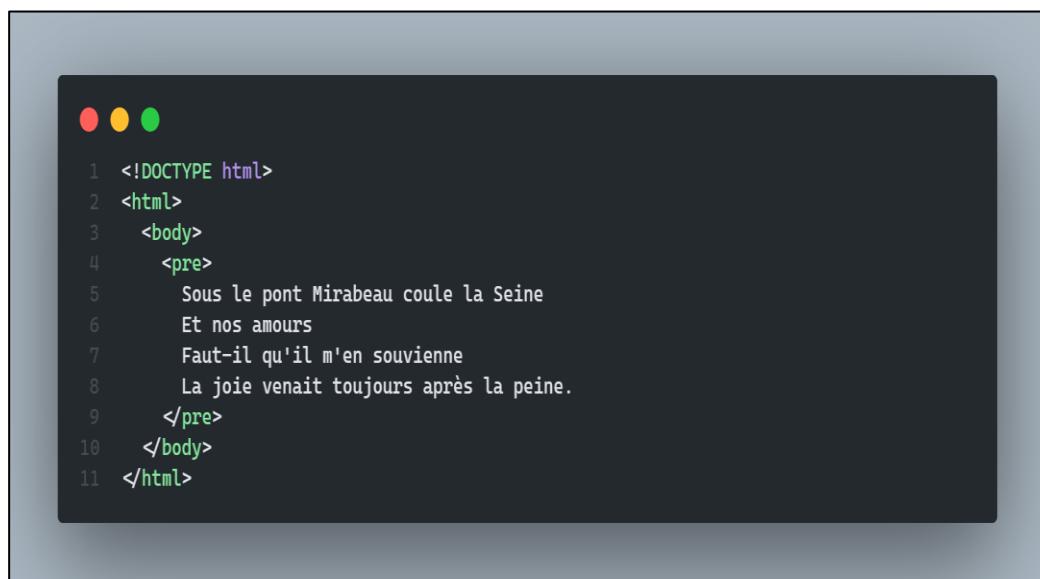
Paragraphes HTML

Solution - L'élément HTML <pre>:

L'élément HTML `<pre>` définit un texte préformaté.

Le texte à l'intérieur d'un élément `<pre>` est affiché dans une police à largeur fixe (généralement Courier) et conserve les espaces et les sauts de ligne:

Exemple:



The image consists of two side-by-side screenshots. On the left, a terminal window displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <pre>
5       Sous le pont Mirabeau coule la Seine
6       Et nos amours
7       Faut-il qu'il m'en souvienne
8       La joie venait toujours après la peine.
9     </pre>
10   </body>
11 </html>
```

On the right, a web browser window shows the rendered output of this code. The text is displayed in a monospaced font and preserves the original line breaks and spaces from the source code:

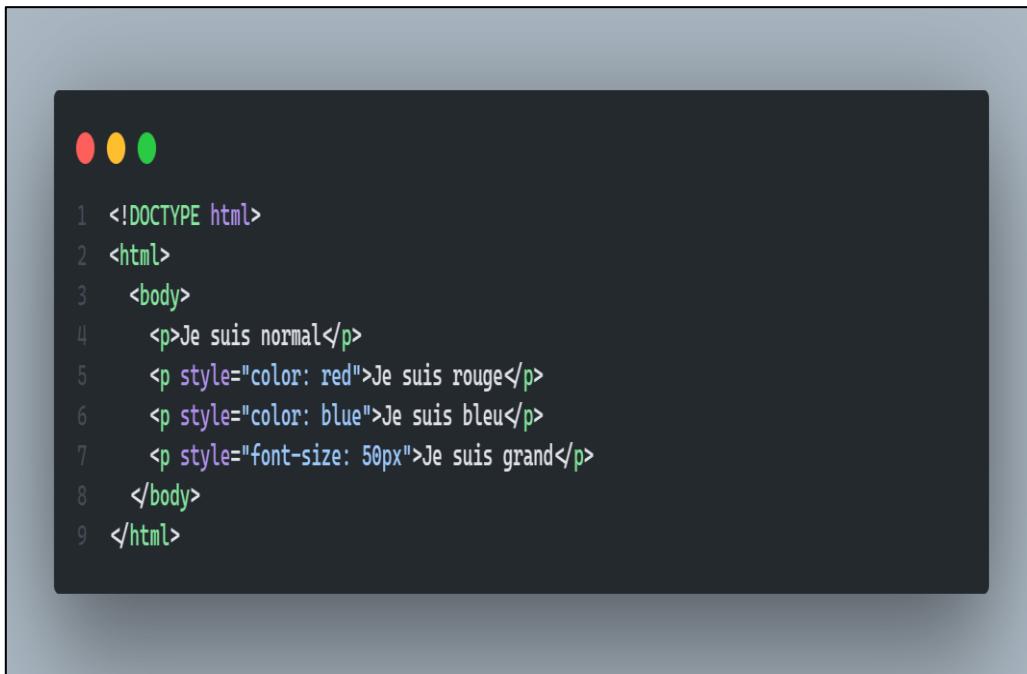
```
Sous le pont Mirabeau coule la Seine
Et nos amours
Faut-il qu'il m'en souvienne
La joie venait toujours après la peine.
```

La balise `pre` préserve à la fois les espaces et les sauts de ligne.

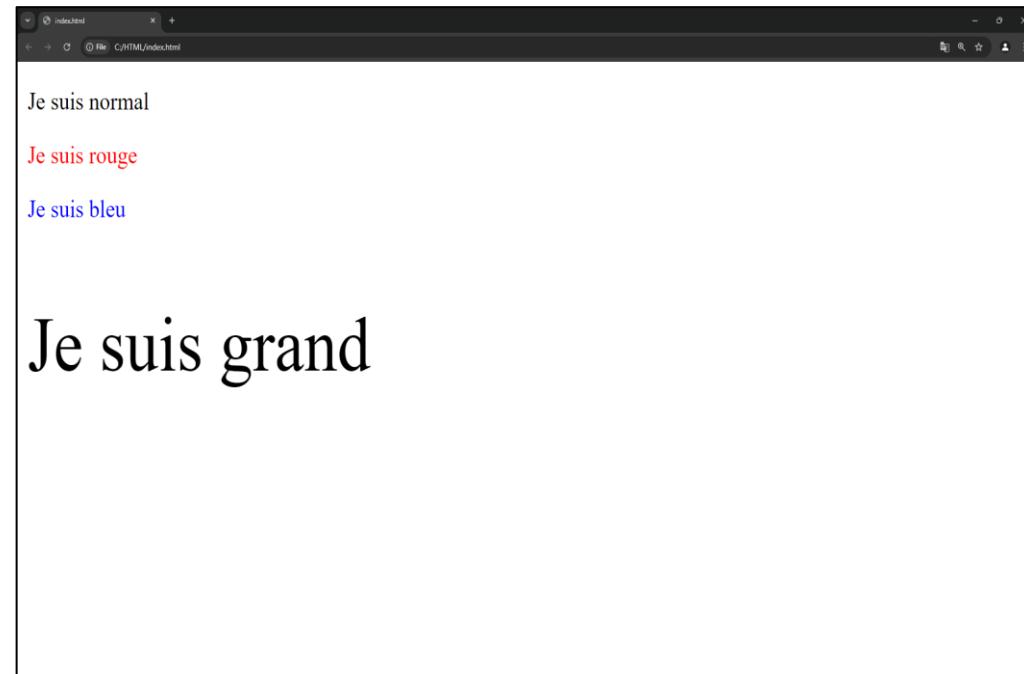
Styles HTML

L'attribut de style HTML est utilisé pour ajouter des styles à un élément, tels que la couleur, la police, la taille, etc.

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Je suis normal</p>
5     <p style="color: red">Je suis rouge</p>
6     <p style="color: blue">Je suis bleu</p>
7     <p style="font-size: 50px">Je suis grand</p>
8   </body>
9 </html>
```



Styles HTML

L'attribut de style HTML:

La définition du style d'un élément HTML peut être effectuée à l'aide de l'attribut **style**.

L'attribut **style** HTML a la syntaxe suivante :

```
<nomdelabalise style="propriété: valeur;">
```

La propriété est une propriété CSS. La valeur est une valeur CSS.

Styles HTML

Background Color:

La propriété CSS `background-color` définit la couleur d'arrière-plan d'un élément HTML.

Exemple: Définir la couleur d'arrière-plan d'une page sur rouge:



The image shows a comparison between a code editor and a web browser. On the left, a code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body style="background-color: red">
4     <h1>Ceci est un titre</h1>
5     <p>Ceci est un paragraphe</p>
6   </body>
7 </html>
```

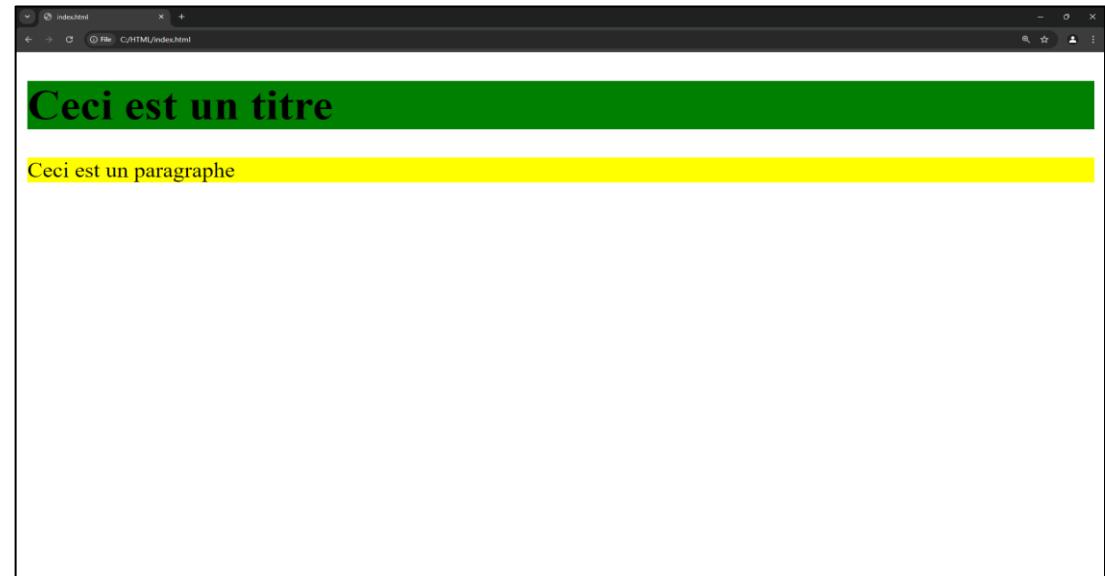
On the right, a web browser window titled "index.html" shows the rendered page with a solid red background. The text "Ceci est un titre" is displayed in a large black font, and "Ceci est un paragraphe" is displayed in a smaller black font below it.

Styles HTML

Exemple: définir la couleur d'arrière-plan pour deux éléments différents:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1 style="background-color: green">Ceci est un titre</h1>
5     <p style="background-color: yellow">Ceci est un paragraphe</p>
6   </body>
7 </html>
```

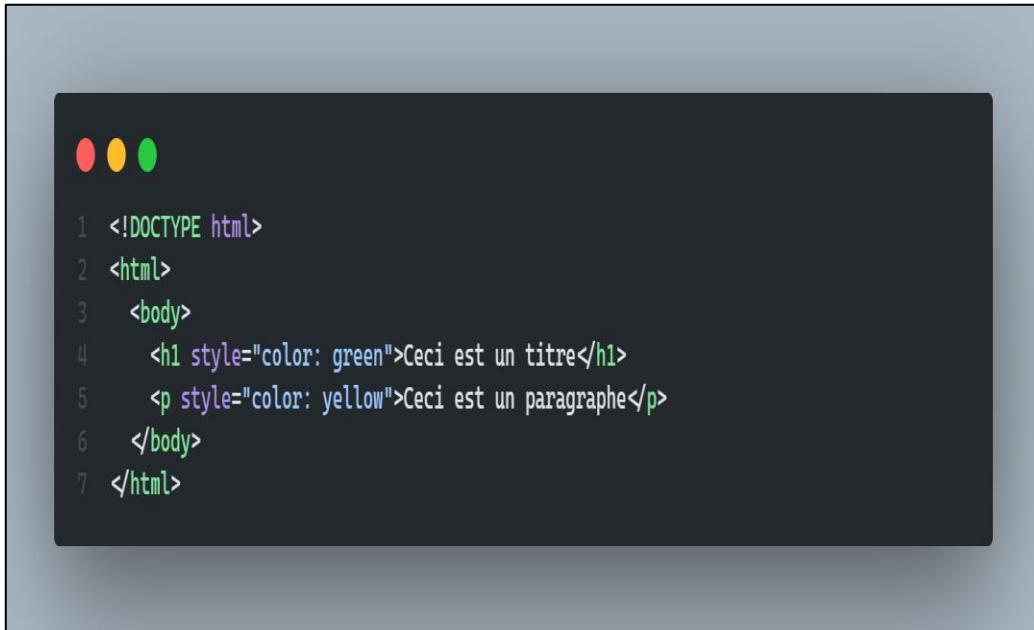


Styles HTML

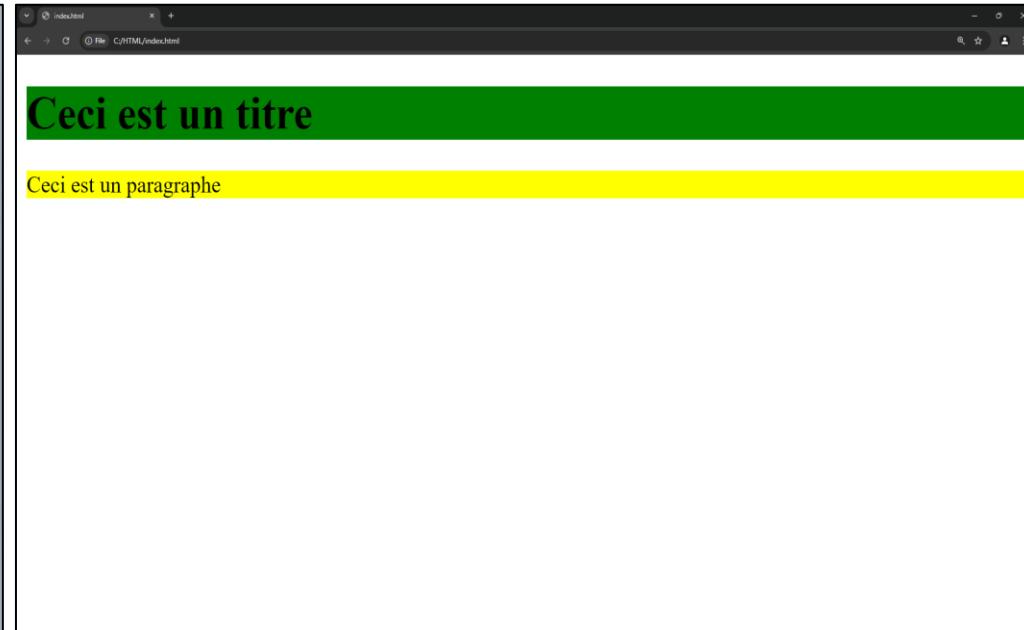
Couleur du texte:

La propriété CSS `color` définit la couleur du texte d'un élément HTML:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1 style="color: green">Ceci est un titre</h1>
5     <p style="color: yellow">Ceci est un paragraphe</p>
6   </body>
7 </html>
```



Styles HTML

Police:

La propriété CSS **font-family** définit la police à utiliser pour un élément HTML :



The image consists of two side-by-side screenshots. The left screenshot shows a terminal window with a dark background and three colored icons (red, yellow, green) at the top. It displays the following code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1 style="font-family: cursive">Ceci est un titre</h1>
5     <p style="font-family: Tahoma">Ceci est un paragraphe</p>
6   </body>
7 </html>
```

The right screenshot shows a web browser window titled "index.html". The address bar indicates the file is located at "C:/HTML/index.html". The page content is:

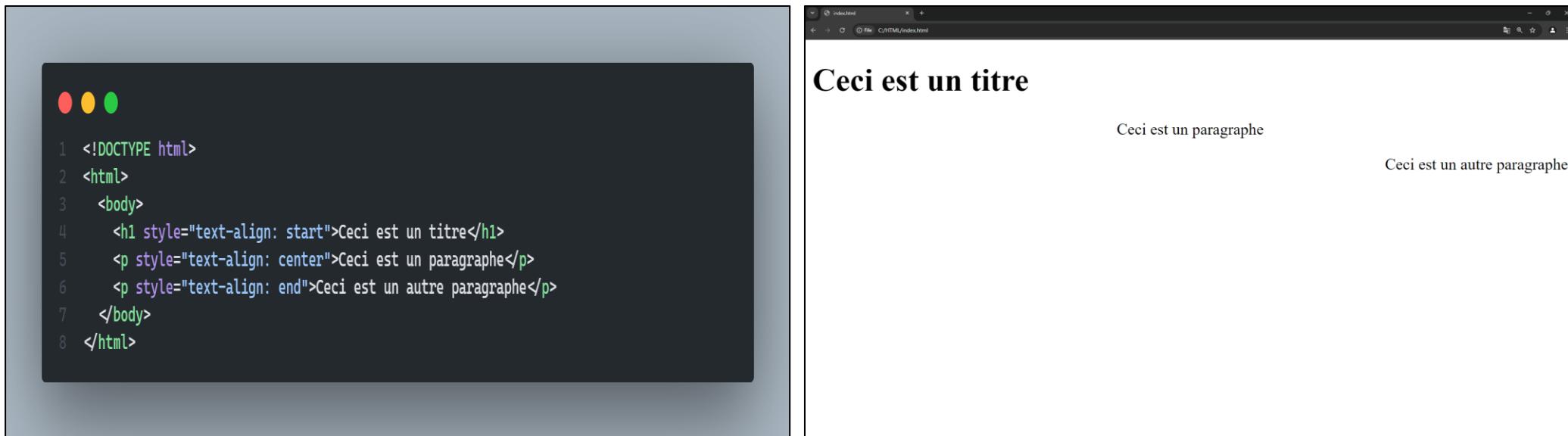
Ceci est un titre

Ceci est un paragraphe

Styles HTML

Alignement du texte :

La propriété CSS `text-align` définit l'alignement horizontal du texte pour un élément HTML:



The image shows a code editor on the left and a web browser on the right. The code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1 style="text-align: start">Ceci est un titre</h1>
5     <p style="text-align: center">Ceci est un paragraphe</p>
6     <p style="text-align: end">Ceci est un autre paragraphe</p>
7   </body>
8 </html>
```

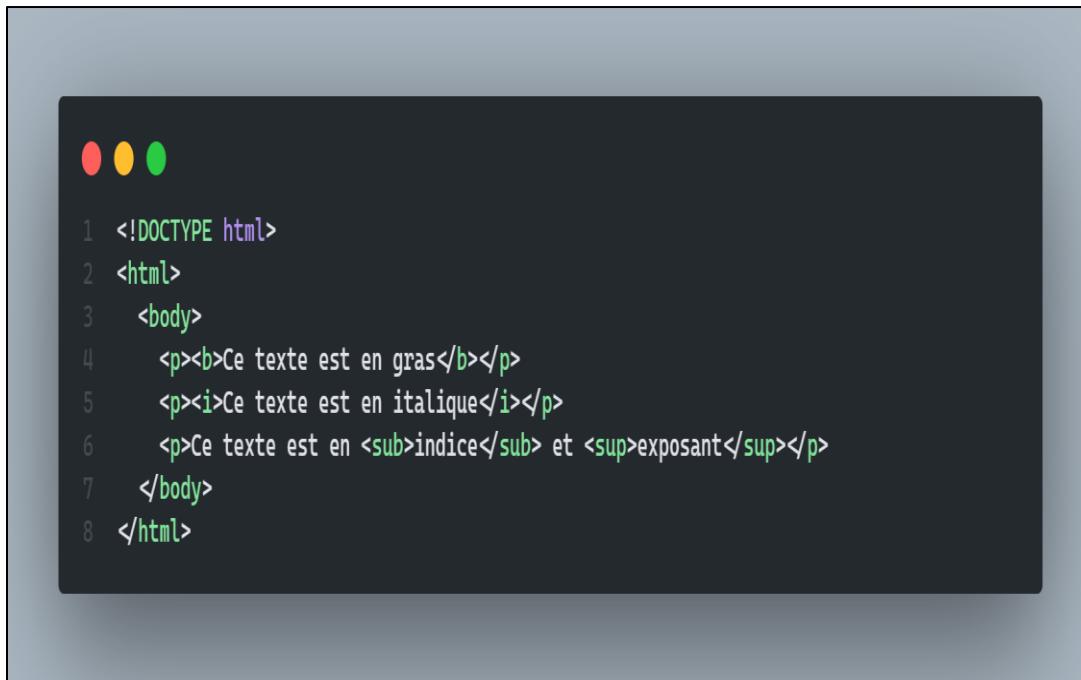
The browser window shows the rendered HTML. The title "Ceci est un titre" is aligned to the start of the line. The first paragraph "Ceci est un paragraphe" is centered. The second paragraph "Ceci est un autre paragraphe" is aligned to the end of the line.

Formatage HTML

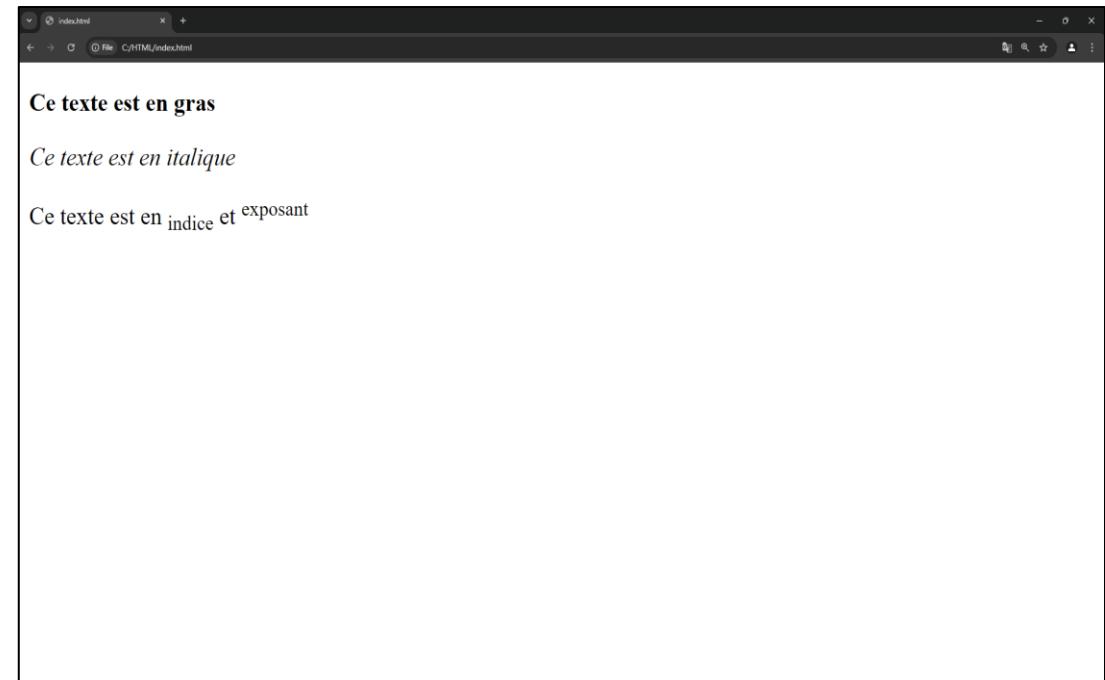
Formatage de texte HTML:

HTML contient plusieurs éléments permettant de définir du texte avec une signification particulière.

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p><b>Ce texte est en gras</b></p>
5     <p><i>Ce texte est en italique</i></p>
6     <p>Ce texte est en <sub>indice</sub> et <sup>exposant</sup></p>
7   </body>
8 </html>
```



Formatage HTML

Éléments HTML et :

Les éléments de formatage ont été conçus pour afficher des types de texte spéciaux:

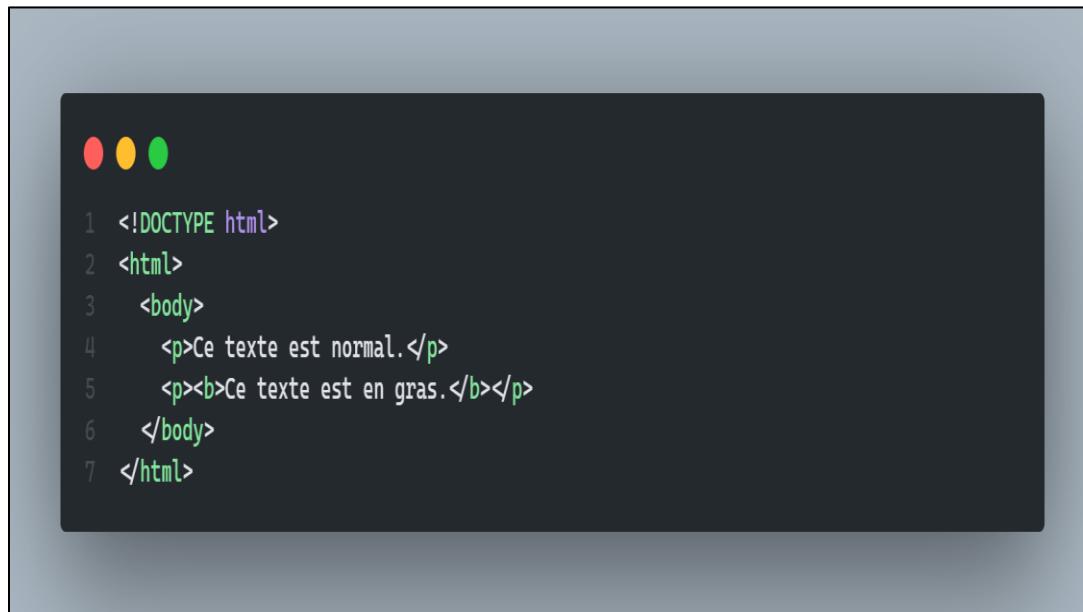
- <**>** - Texte important
- <*i*> - Texte en italique
- <*>* - Texte mis en valeur en italique
- <> - Texte marqué
- <> - Texte plus petit
- <~~>~~ - Texte supprimé (barré)
- <> - Texte inséré
- <_> - Texte en indice
- <[>] - Texte en exposant

Formatage HTML

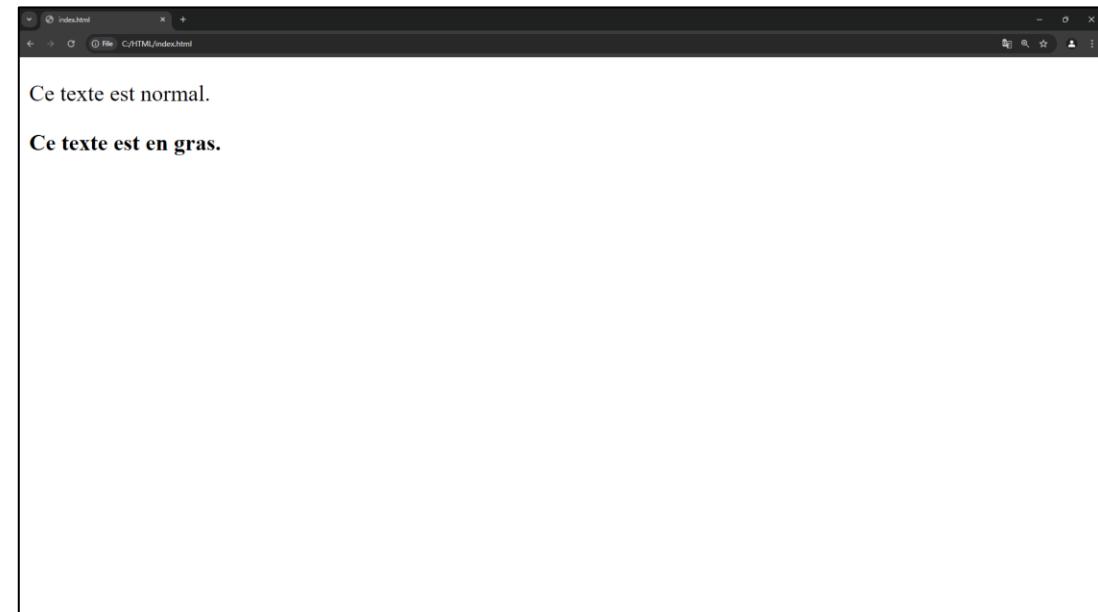
Éléments HTML **** et ****:

L'élément HTML **** définit un texte en gras, sans aucune importance supplémentaire.

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ce texte est normal.</p>
5     <p><b>Ce texte est en gras.</b></p>
6   </body>
7 </html>
```



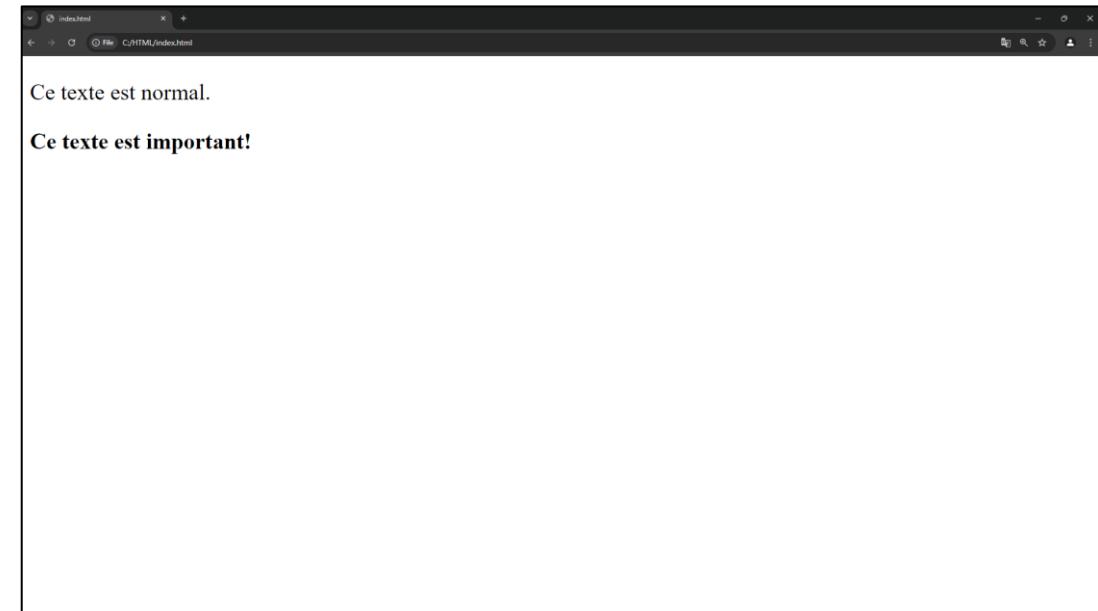
Formatage HTML

L'élément HTML **** définit un texte ayant une importance particulière. Le contenu à l'intérieur est généralement affiché en gras.

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ce texte est normal.</p>
5     <p><strong>Ce texte est important!</strong></p>
6   </body>
7 </html>
```



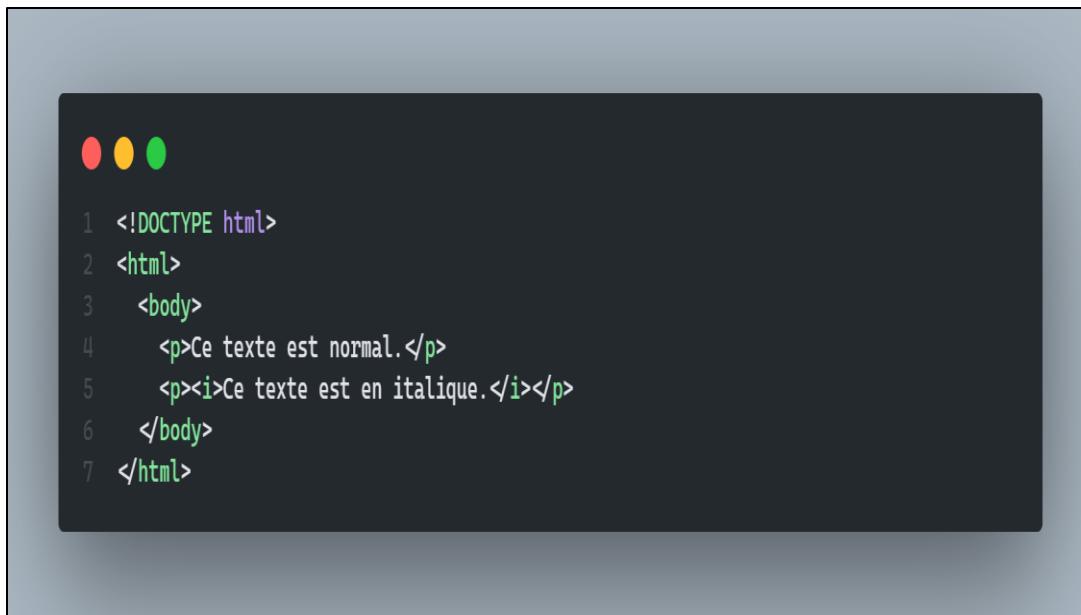
Formatage HTML

Éléments HTML <i> et :

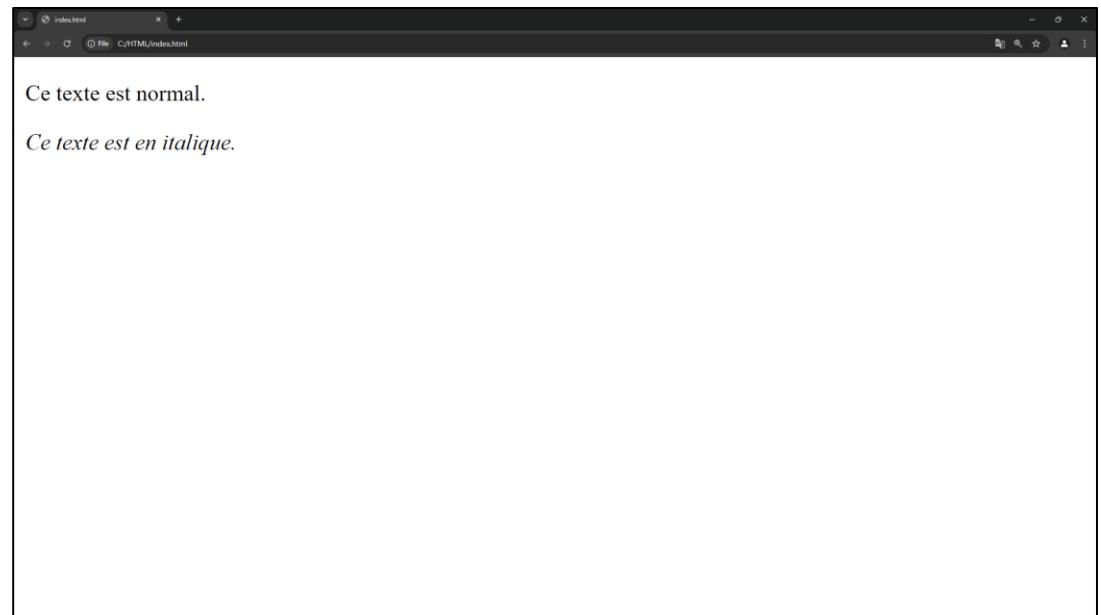
L'élément HTML <i> définit une partie de texte dans une voix ou une humeur alternative. Le contenu à l'intérieur est généralement affiché en italique.

Astuce: la balise <i> est souvent utilisée pour indiquer un terme technique, une phrase d'une autre langue, une pensée, un nom de navire, etc.

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ce texte est normal.</p>
5     <p><i>Ce texte est en italique.</i></p>
6   </body>
7 </html>
```

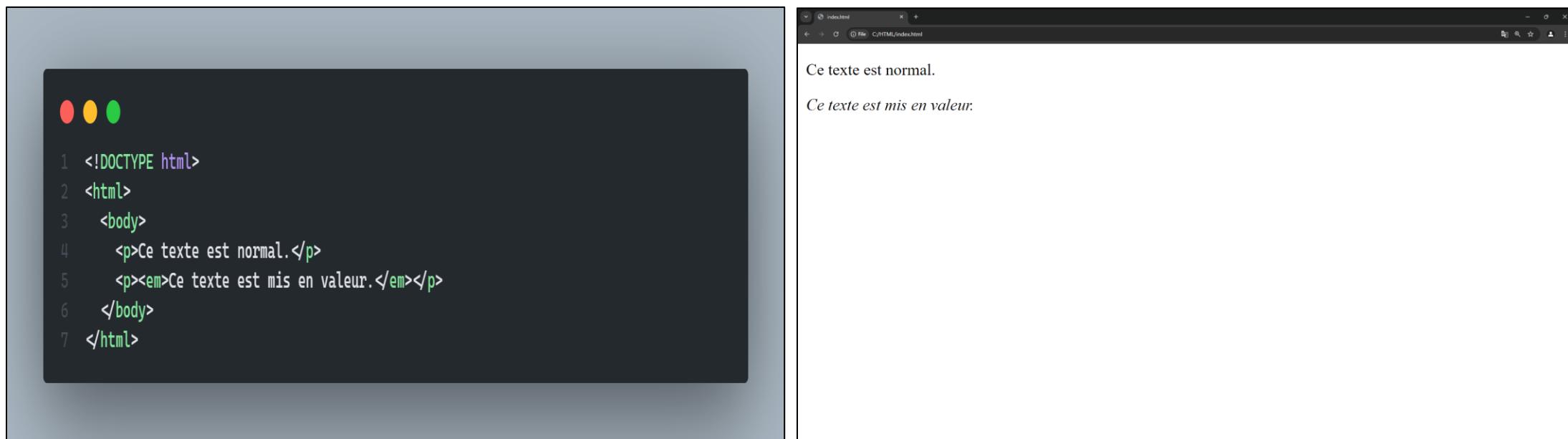


Formatage HTML

L'élément HTML `` définit le texte mis en valeur. Le contenu à l'intérieur est généralement affiché en italique.

Astuce: un lecteur d'écran prononcera les mots entre `` avec emphase, en utilisant l'accentuation verbale.

Exemple:



The image shows a comparison between a code editor and a web browser. On the left, a code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ce texte est normal.</p>
5     <p><em>Ce texte est mis en valeur.</em></p>
6   </body>
7 </html>
```

On the right, a web browser window titled "index.html" shows the rendered HTML. The first paragraph is in a standard black font, while the second paragraph contains the word "mis en valeur." in an italicized font, indicating it is being emphasized.

Formatage HTML

Élément HTML <small>:

L'élément HTML <small> définit un texte plus petit:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ceci est un texte normal.</p>
5     <p><small>Ceci est un texte plus petit.</small></p>
6   </body>
7 </html>
```



Formatage HTML

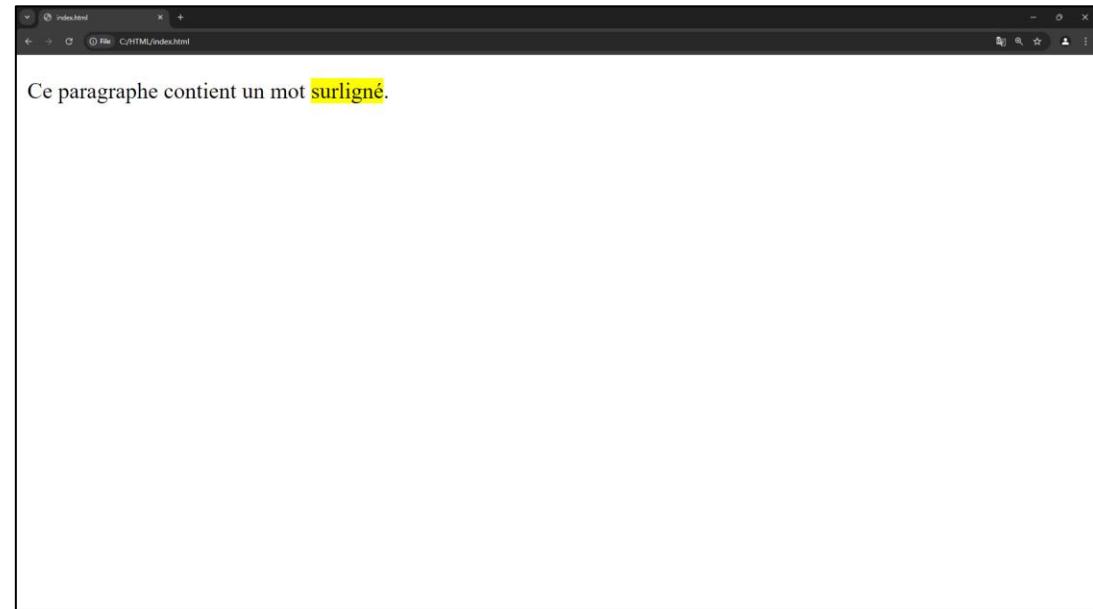
Élément HTML <mark>:

L'élément HTML <mark> définit le texte qui doit être marqué ou mis en surbrillance:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ce paragraphe contient un mot <mark>surligné</mark>.</p>
5   </body>
6 </html>
```

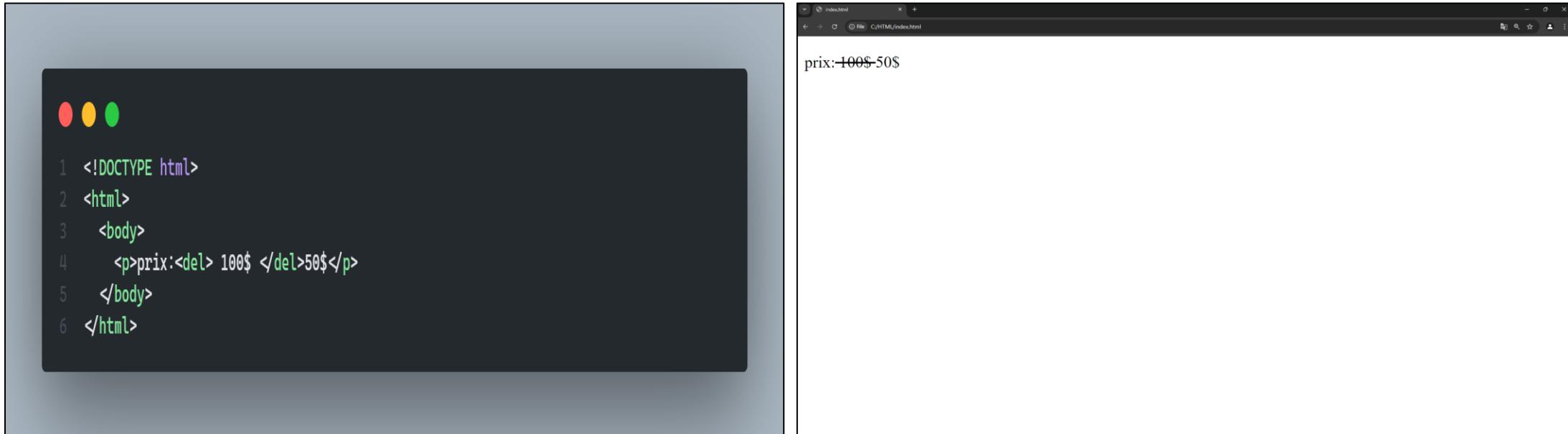


Formatage HTML

Élément HTML :

L'élément HTML définit le texte qui a été supprimé d'un document. Les navigateurs barrent généralement le texte supprimé:

Exemple:



The image shows a terminal window on the left and a web browser window on the right. The terminal window displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>prix:<del> 100$ </del>50$</p>
5   </body>
6 </html>
```

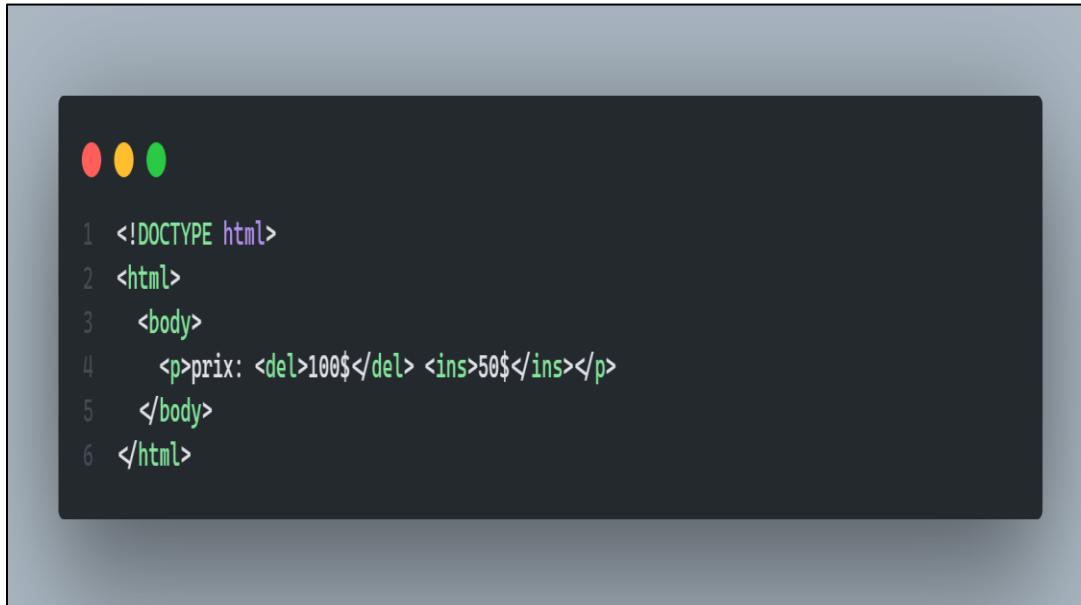
The web browser window shows the rendered HTML with the struck-through text "100\$".

Formatage HTML

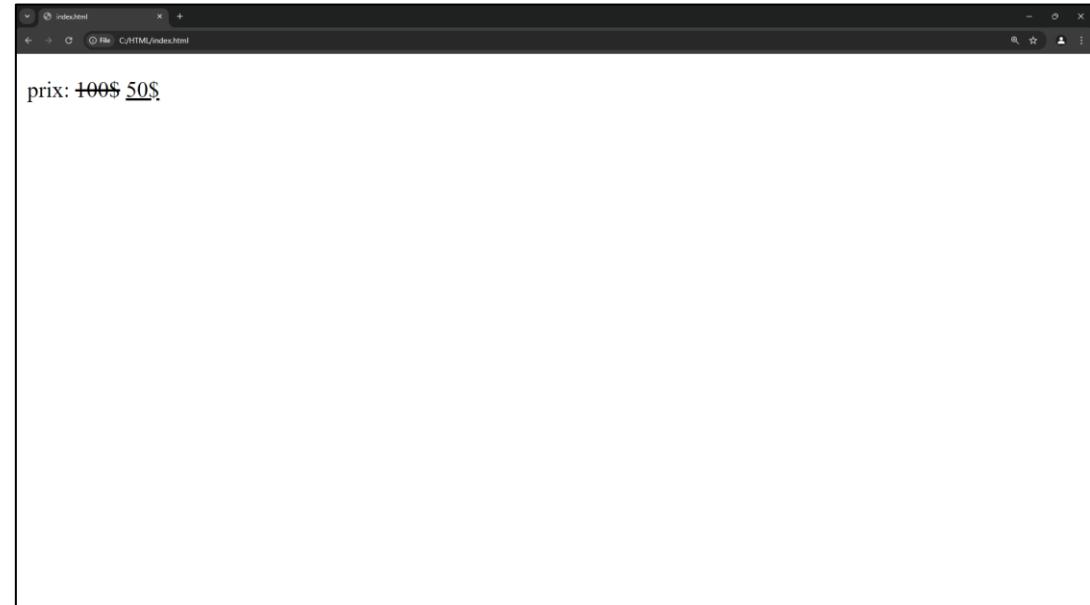
Élément HTML <ins>:

L'élément HTML <ins> définit un texte qui a été inséré dans un document. Les navigateurs soulignent généralement le texte inséré:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>prix: <del>100$</del> <ins>50$</ins></p>
5   </body>
6 </html>
```

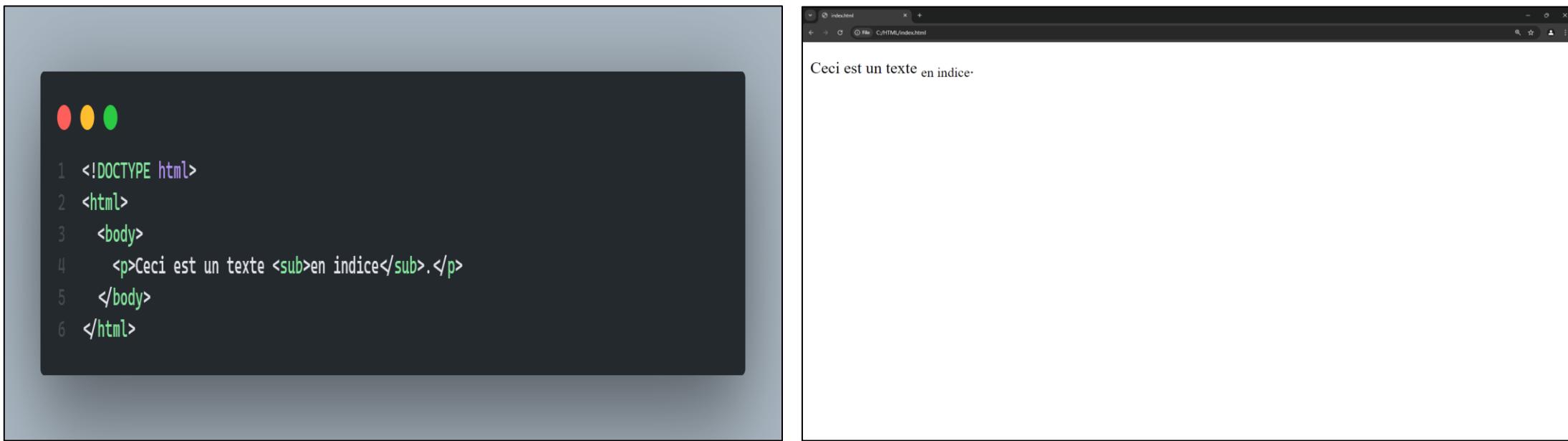


Formatage HTML

Élément HTML <sub>:

L'élément HTML <sub> définit le texte en indice. Le texte en indice apparaît un demi-caractère en dessous de la ligne normale et est parfois rendu dans une police plus petite. Le texte en indice peut être utilisé pour les formules chimiques, comme H₂O:

Exemple:



The image shows a terminal window on the left and a web browser window on the right. The terminal window displays the following code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ceci est un texte <sub>en indice</sub>.</p>
5   </body>
6 </html>
```

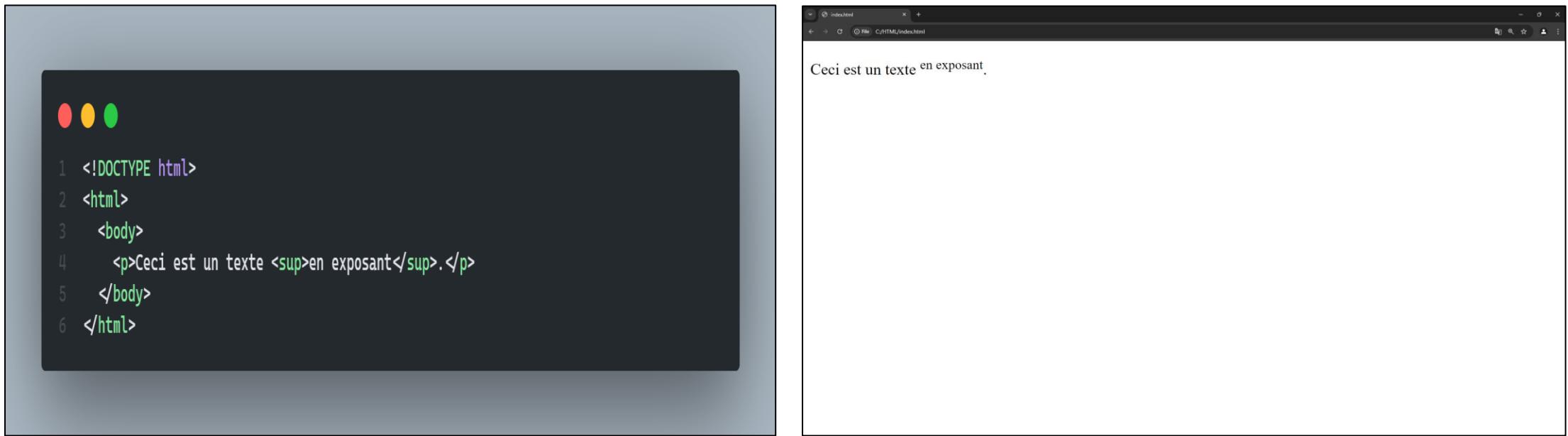
The web browser window shows the rendered HTML. The word "en" is displayed in a smaller, gray font, positioned below the baseline of the surrounding text.

Formatage HTML

Élément HTML <sup>:

L'élément HTML <sup> définit le texte en exposant. Le texte en exposant apparaît un demi-caractère au-dessus de la ligne normale et est parfois rendu dans une police plus petite. Le texte en exposant peut être utilisé pour les notes de bas de page, comme WWW[1]:

Exemple:



The image shows a comparison between a code editor and a web browser. On the left, a code editor displays the following HTML code:

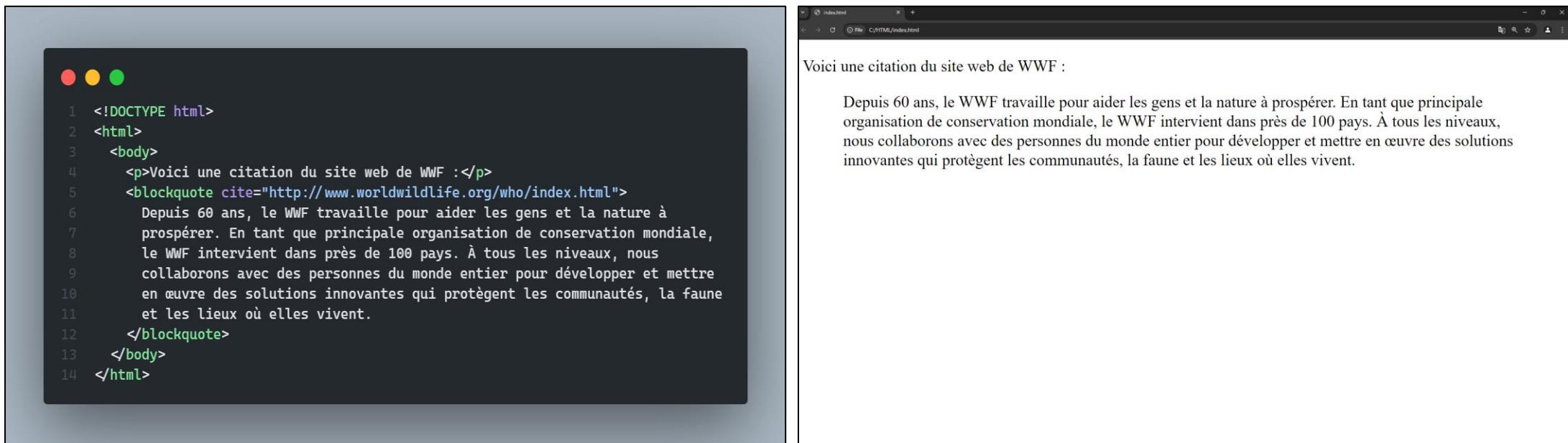
```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ceci est un texte <sup>en exposant</sup>.</p>
5   </body>
6 </html>
```

On the right, a web browser window titled "index.html" shows the rendered output: "Ceci est un texte en exposant". The word "en exposant" is displayed in a smaller, gray font, positioned above the baseline of the surrounding text.

Éléments de citation HTML

Dans ce chapitre, nous allons parcourir les éléments HTML `<blockquote>`, `<q>`, `<abbr>`, `<address>`, `<cite>` et `<bdo>`.

Exemple:



The image shows a split-screen view. On the left, a code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Voici une citation du site web de WWF :</p>
5     <blockquote cite="http://www.worldwildlife.org/who/index.html">
6       Depuis 60 ans, le WWF travaille pour aider les gens et la nature à
7       prospérer. En tant que principale organisation de conservation mondiale,
8       le WWF intervient dans près de 100 pays. À tous les niveaux,
9       nous collaborons avec des personnes du monde entier pour développer et mettre en œuvre des solutions
10      innovantes qui protègent les communautés, la faune et les lieux où elles vivent.
11    </blockquote>
12  </body>
13 </html>
```

On the right, a web browser window shows the rendered HTML. The page title is "index.html". The content of the page is:

Voici une citation du site web de WWF :

Depuis 60 ans, le WWF travaille pour aider les gens et la nature à prospérer. En tant que principale organisation de conservation mondiale, le WWF intervient dans près de 100 pays. À tous les niveaux, nous collaborons avec des personnes du monde entier pour développer et mettre en œuvre des solutions innovantes qui protègent les communautés, la faune et les lieux où elles vivent.

Éléments de citation HTML

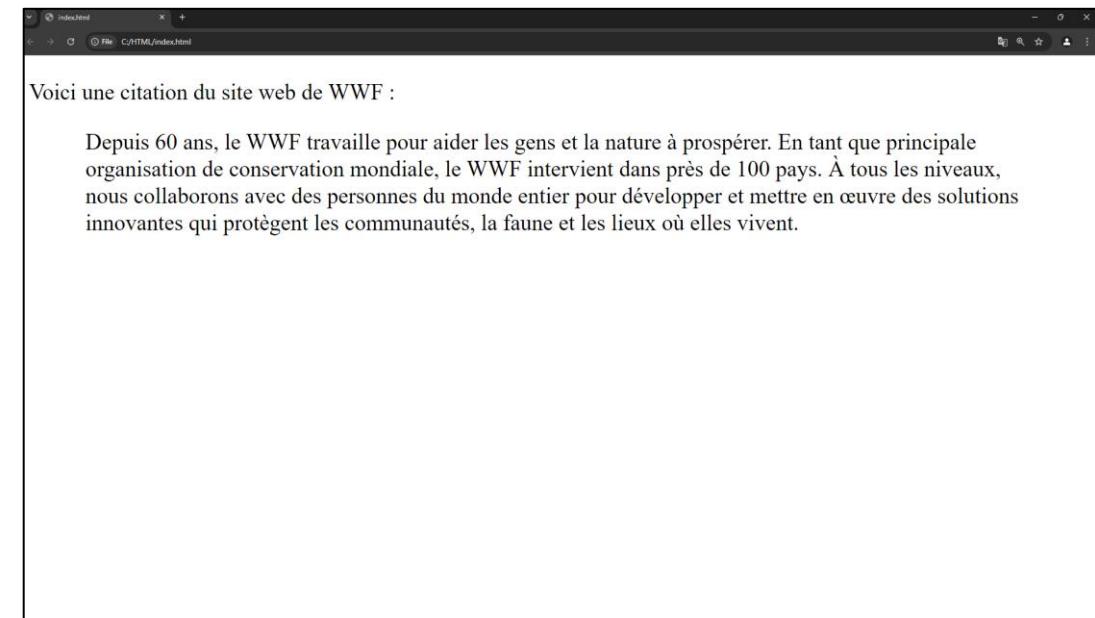
HTML <blockquote> pour les citations:

L'élément HTML <blockquote> définit une section citée à partir d'une autre source.

Les navigateurs indentent généralement les éléments <blockquote>.

Exemple:

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4      <p>Voici une citation du site web de WWF :</p>
5      <blockquote cite="http://www.worldwildlife.org/who/index.html">
6          Depuis 60 ans, le WWF travaille pour aider les gens et la nature à
7              prospérer. En tant que principale organisation de conservation mondiale,
8                  le WWF intervient dans près de 100 pays. À tous les niveaux, nous
9                      collaborons avec des personnes du monde entier pour développer et mettre en œuvre des solutions innovantes qui protègent les communautés, la faune
10                         et les lieux où elles vivent.
11
12     </blockquote>
13
14 </body>
15 </html>
```



Éléments de citation HTML

HTML <q> pour les citations courtes:

La balise HTML <q> définit une citation courte.

Les navigateurs insèrent normalement des guillemets autour de la citation.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>
5       L'objectif du WWF est de :
6       <q>construire un avenir où les gens vivent en harmonie avec la nature.</q>
7     </p>
8   </body>
9 </html>
```



Éléments de citation HTML

HTML <abbr> pour les abréviations:

La balise HTML <abbr> définit une abréviation ou un acronyme, comme "HTML", "CSS", "CSS".

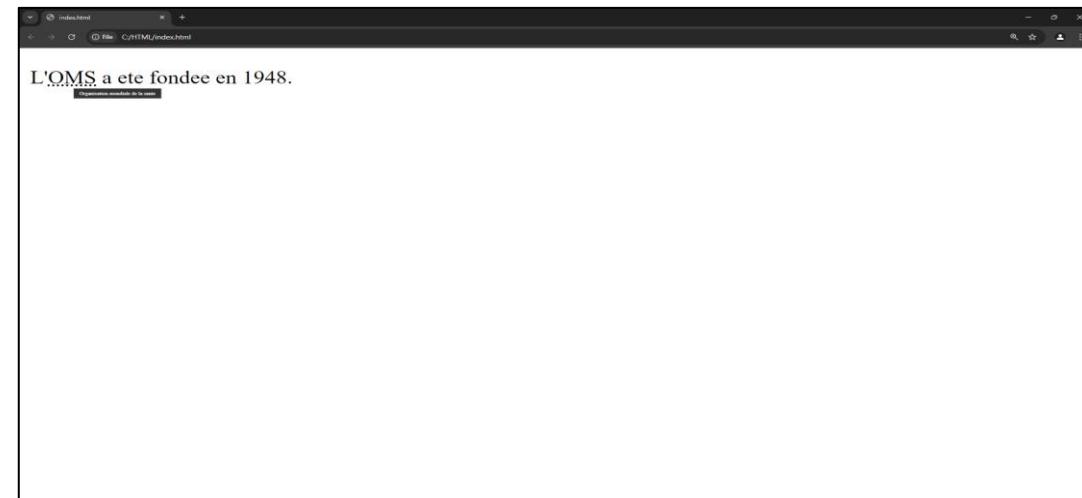
Les navigateurs insèrent normalement des guillemets autour de la citation.

Le marquage des abréviations peut fournir des informations utiles aux navigateurs, aux systèmes de traduction et aux moteurs de recherche.

Astuce : utilisez l'attribut global title pour afficher la description de l'abréviation/acronyme lorsque vous passez la souris sur l'élément.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>
5       L'<abbr title="Organisation mondiale de la santé">OMS</abbr> a été fondée
6       en 1948.
7     </p>
8   </body>
9 </html>
```



Éléments de citation HTML

HTML <address> pour les informations de contact:

La balise HTML <address> définit les informations de contact de l'auteur/propriétaire d'un document ou d'un article.

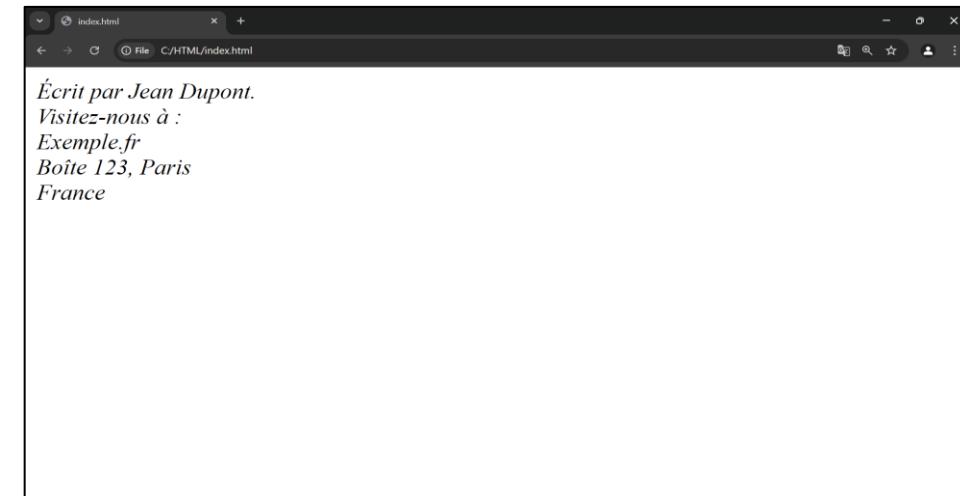
Les informations de contact peuvent être une adresse e-mail, une URL, une adresse physique, un numéro de téléphone, un identifiant de réseau social, etc.

Le texte de l'élément <address> s'affiche généralement en italique et les navigateurs ajouteront toujours un saut de ligne avant et après l'élément <address>.

Exemple:



```
1  <!DOCTYPE html>
2  <html>
3  <body>
4      <address>
5          Écrit par Jean Dupont.<br />
6          Visitez-nous à :<br />
7          Exemple.fr<br />
8          Boîte 123, Paris<br />
9          France
10     </address>
11  </body>
12 </html>
```



Éléments de citation HTML

HTML <cite> pour le titre de l'œuvre:

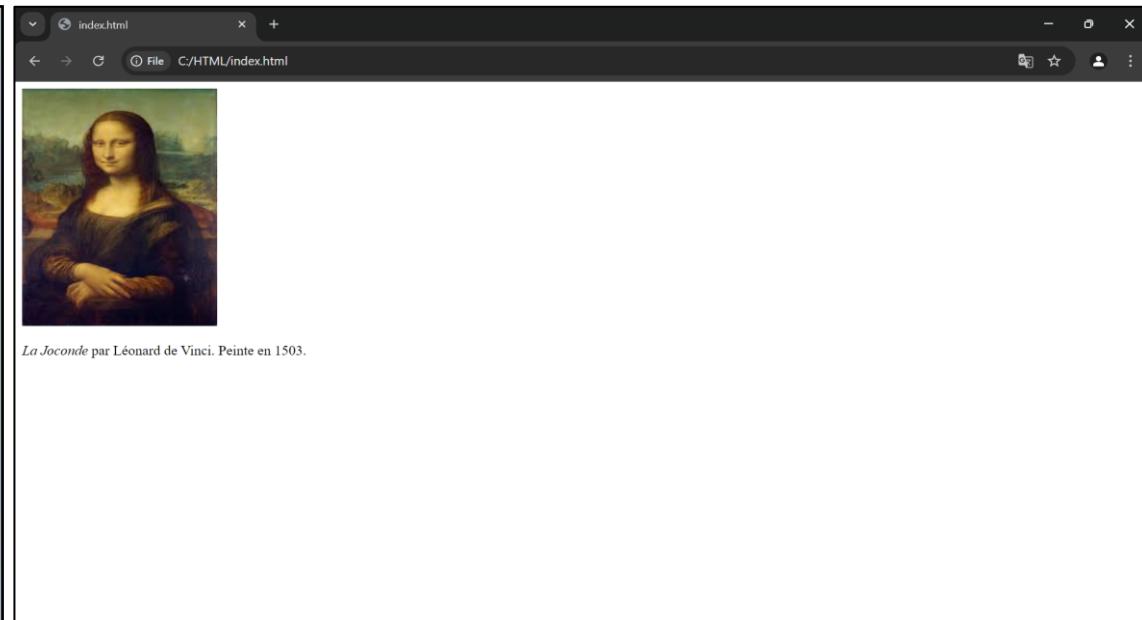
La balise HTML <cite> définit le titre d'une œuvre créative (par exemple un livre, un poème, une chanson, un film, une peinture, une sculpture, etc.).

Remarque: le nom d'une personne n'est pas le titre d'une œuvre.

Le texte de l'élément <cite> s'affiche généralement en italique.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5     <p><cite>La Joconde</cite> par Léonard de Vinci. Peinte en 1503.</p>
6   </body>
7 </html>
```

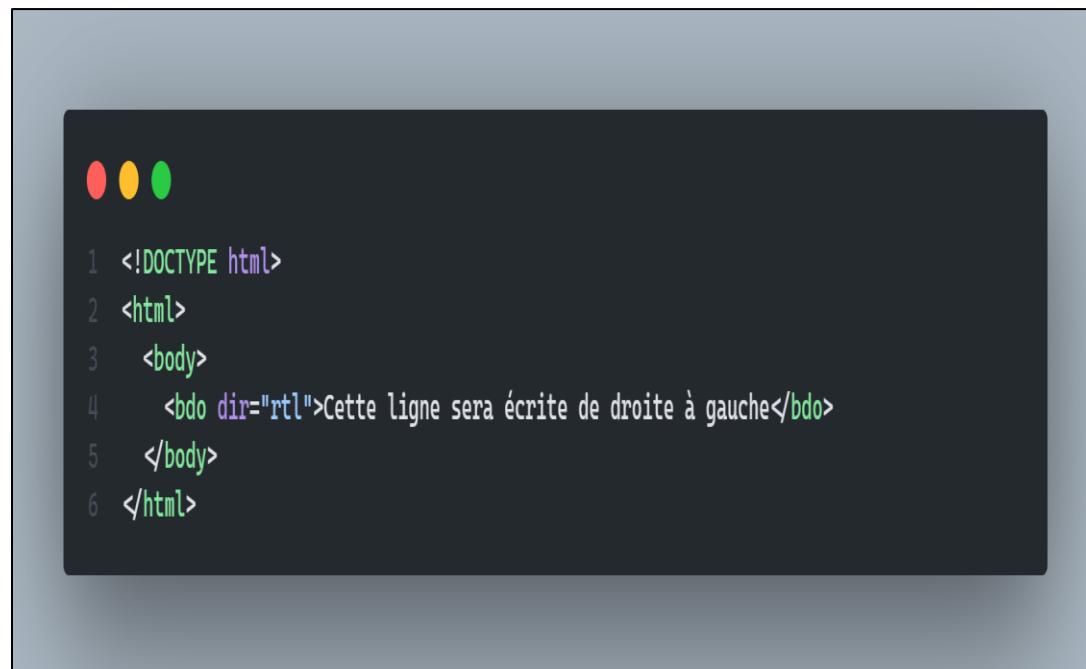


Éléments de citation HTML

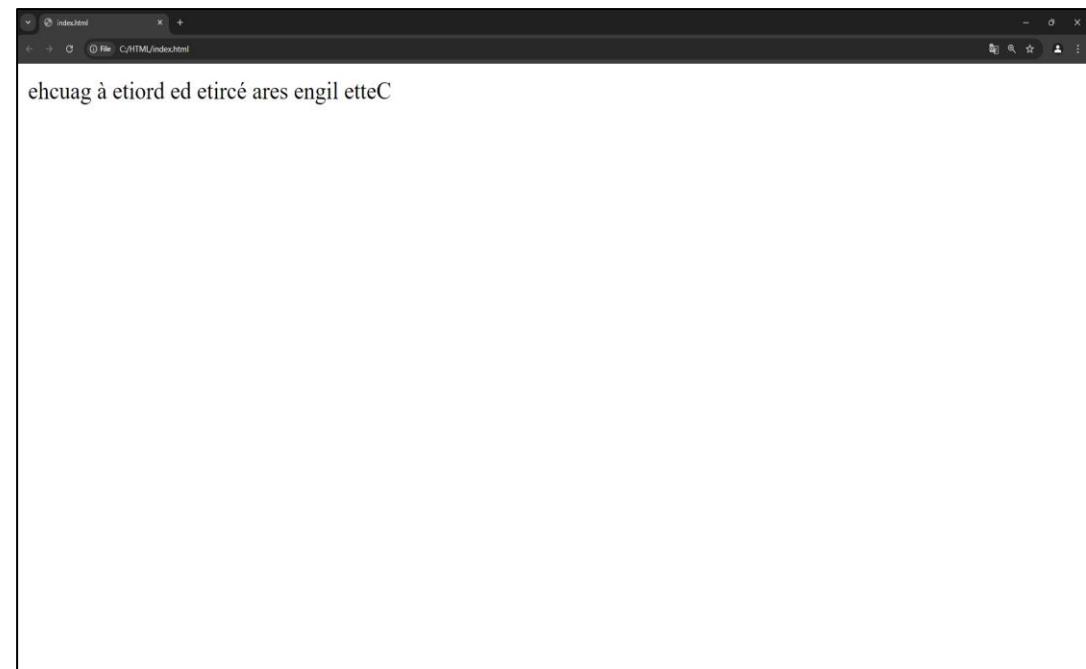
HTML <bdo> pour le remplacement bidirectionnel:

BDO signifie Bi-Directional Override.

La balise HTML <bdo> est utilisée pour remplacer la direction actuelle du texte:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <bdo dir="rtl">Cette ligne sera écrite de droite à gauche</bdo>
5   </body>
6 </html>
```

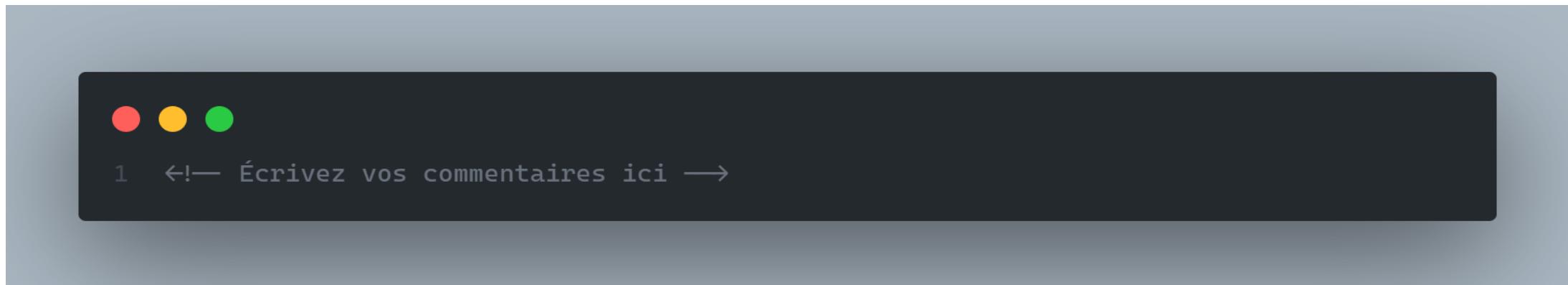


Commentaires HTML

Les commentaires HTML ne sont pas affichés dans le navigateur, mais ils peuvent vous aider à documenter votre code source HTML.

Balise de commentaire HTML:

Vous pouvez ajouter des commentaires à votre source HTML en utilisant la syntaxe suivante:



Notez qu'il y a un point d'exclamation (!) dans la balise de début, mais pas dans la balise de fin.

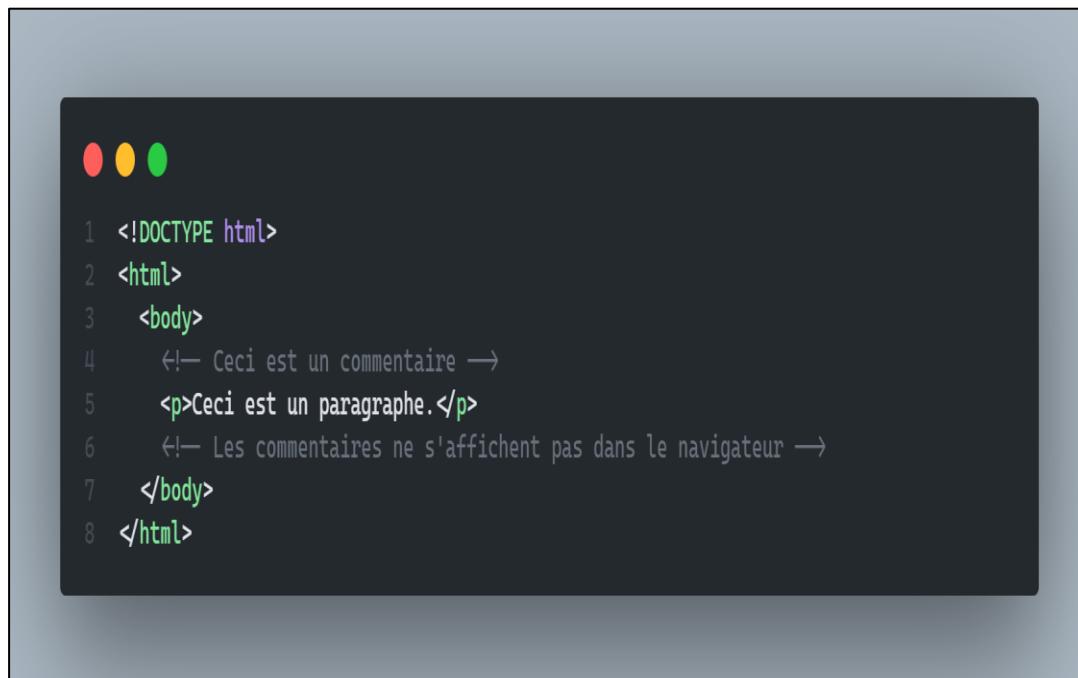
Remarque: les commentaires ne sont pas affichés par le navigateur, mais ils peuvent vous aider à documenter votre code source HTML.

Commentaires HTML

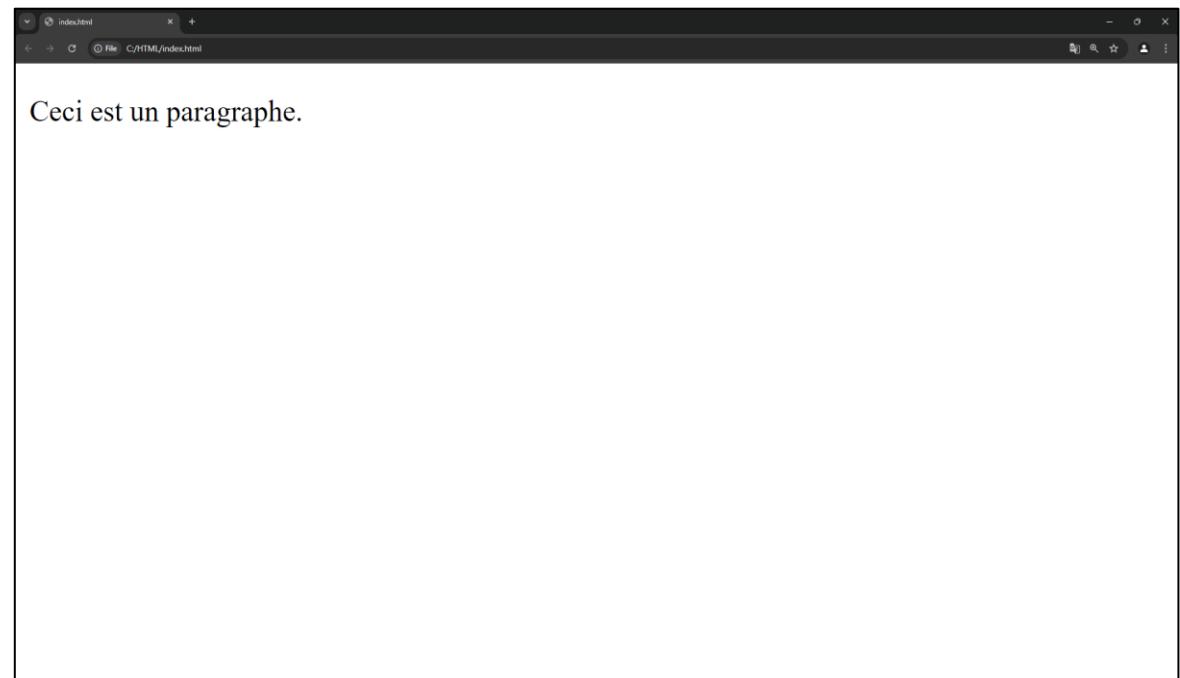
Ajouter des commentaires:

Avec les commentaires, vous pouvez placer des notifications et des rappels dans votre code HTML:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5     <p>Ceci est un paragraphe.</p>
6     
7   </body>
8 </html>
```

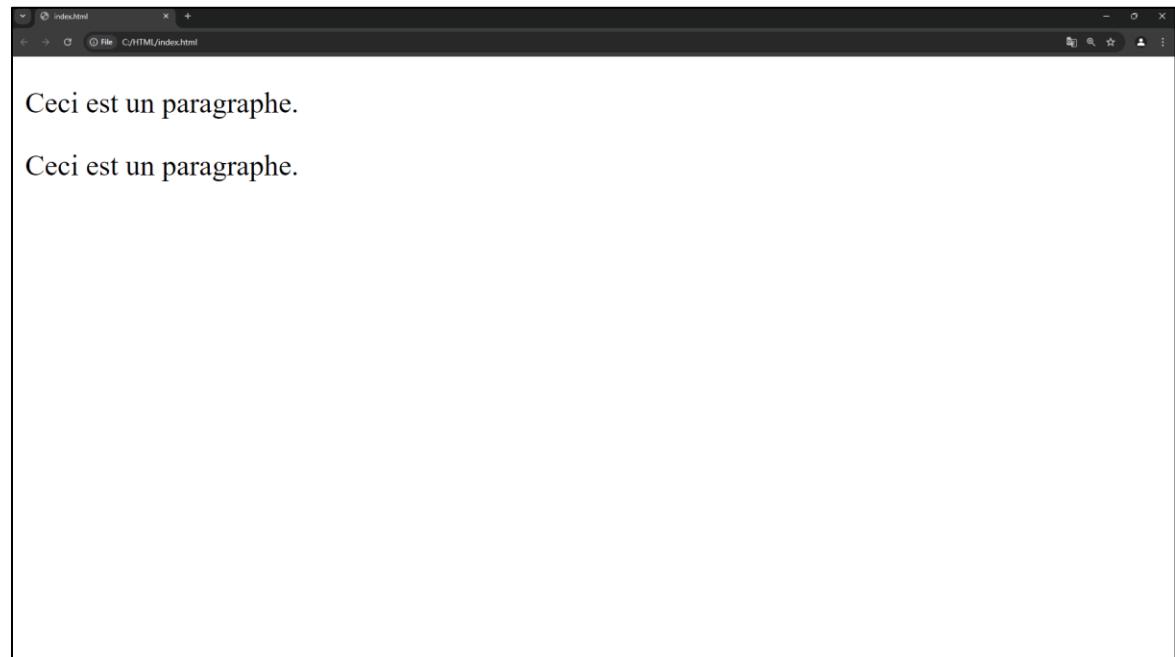


Commentaires HTML

Exemple: Masquer une section de code HTML:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ceci est un paragraphe.</p>
5     <!--
6       Ceci est un autre paragraphe </p>
7       Ceci est également un paragraphe.</p>
8     -->
9     <p>Ceci est un paragraphe.</p>
10   </body>
11 </html>
```



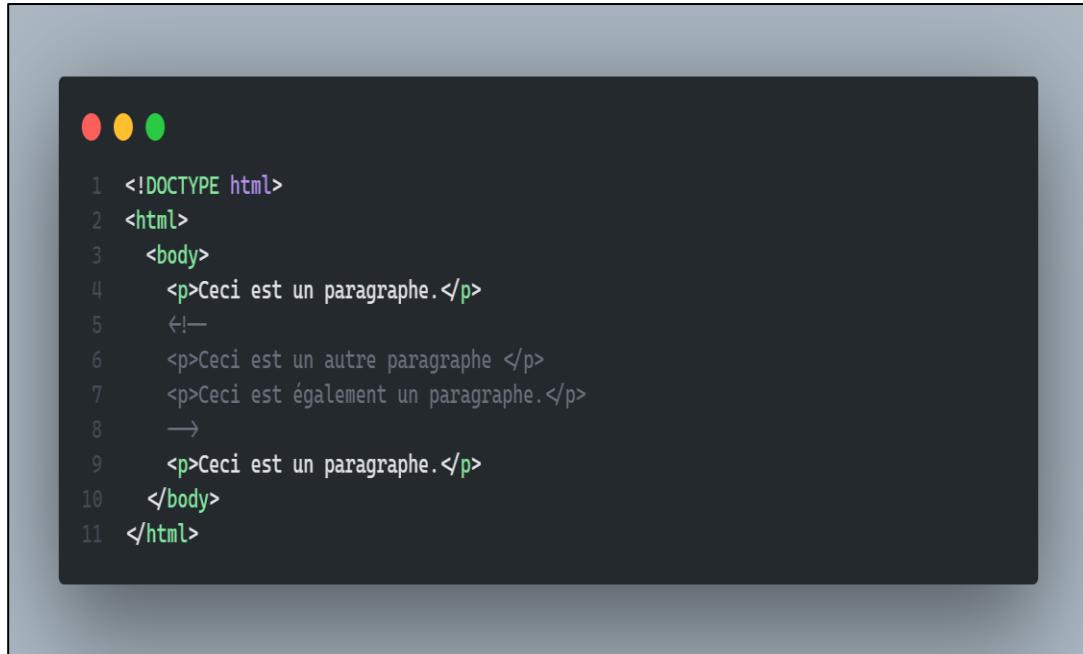
Vous pouvez également masquer plusieurs lignes. Tout ce qui se trouve entre le <!-- et le --> sera masqué à l'écran.

Commentaires HTML

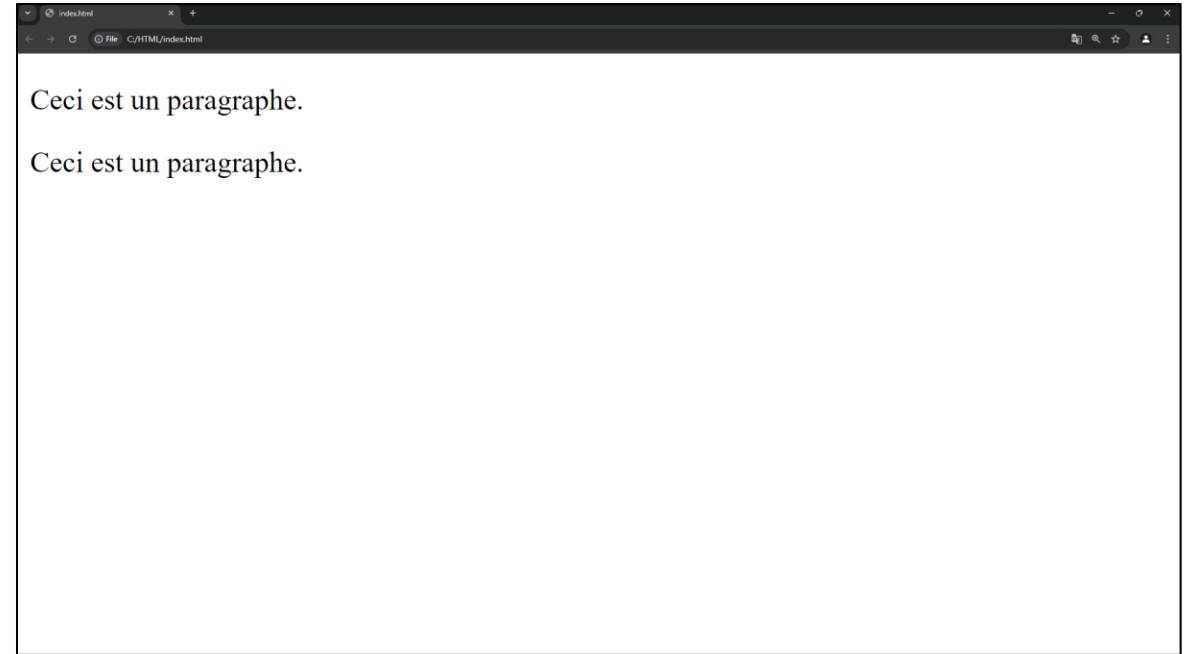
Masquer le contenu en ligne:

Les commentaires peuvent être utilisés pour masquer des parties au milieu du code HTML.

Exemple: Masquer une partie d'un paragraphe:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ceci est un paragraphe.</p>
5     <!--
6     <p>Ceci est un autre paragraphe </p>
7     <p>Ceci est également un paragraphe.</p>
8     -->
9     <p>Ceci est un paragraphe.</p>
10   </body>
11 </html>
```

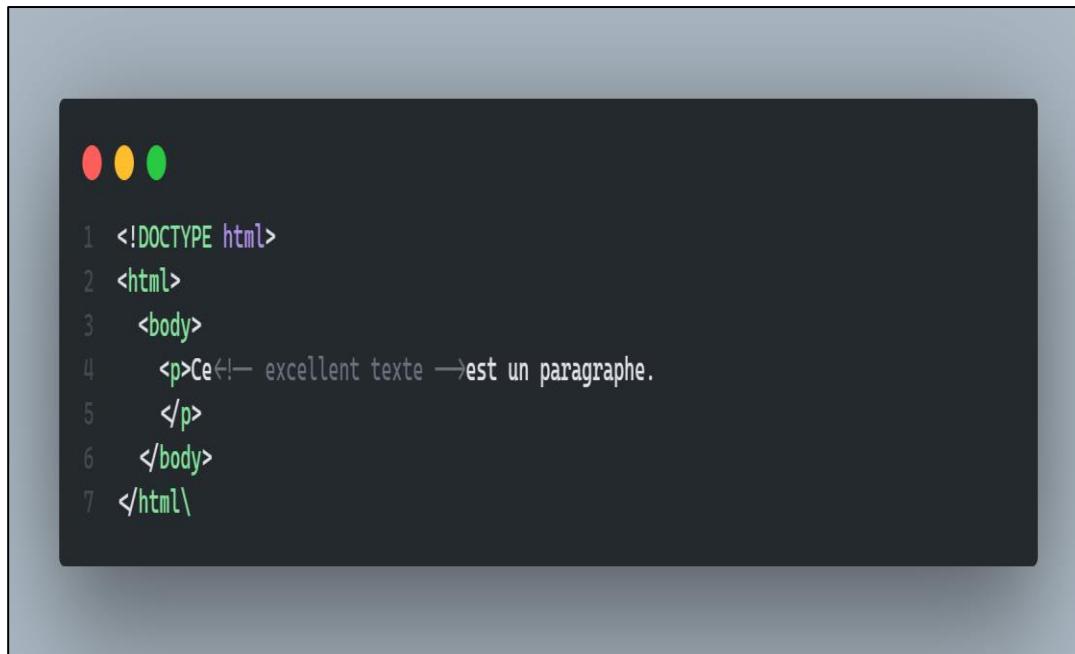


Commentaires HTML

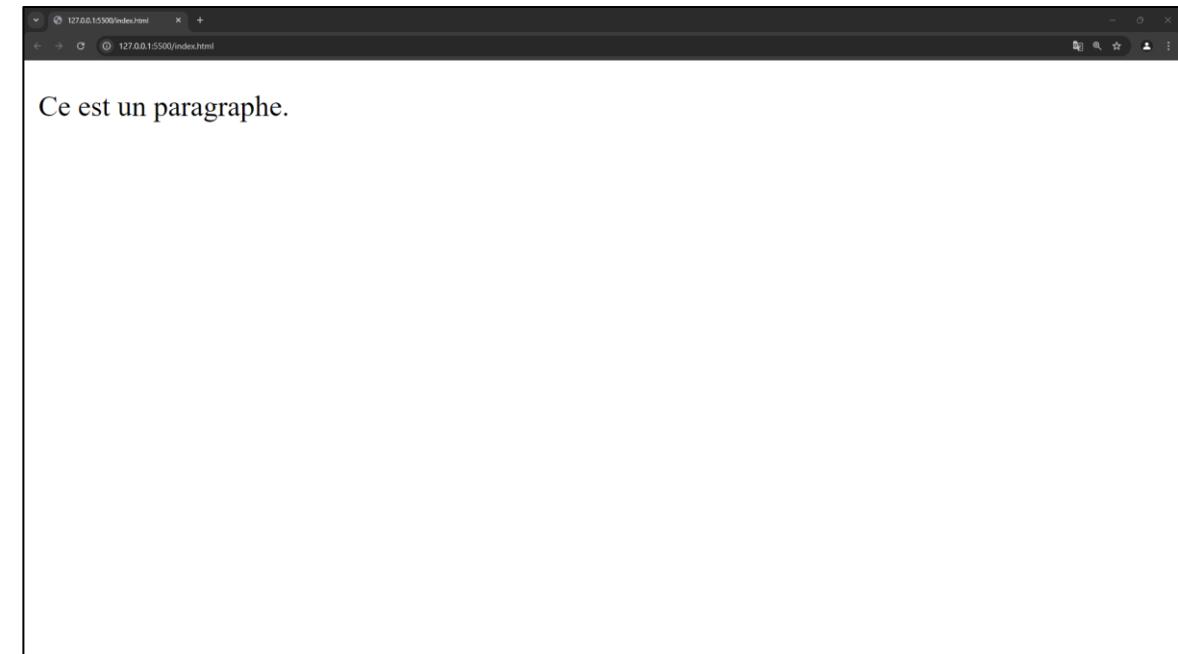
Masquer le contenu en ligne:

Les commentaires peuvent être utilisés pour masquer des parties au milieu du code HTML.

Exemple: Masquer une partie d'un paragraphe:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Ce<!-- excellent texte -->est un paragraphe.
5   </p>
6   </body>
7 </html>
```

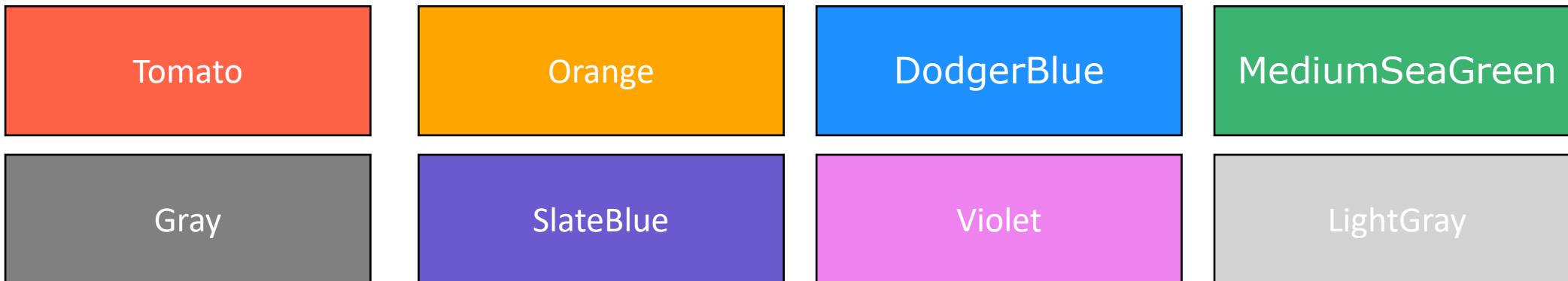


Couleurs HTML

Les couleurs HTML sont spécifiées avec des noms de couleurs prédéfinis ou avec des valeurs RGB, HEX, HSL, RGBA ou HSLA.

Noms des couleurs:

En HTML, une couleur peut être spécifiée en utilisant un nom de couleur :



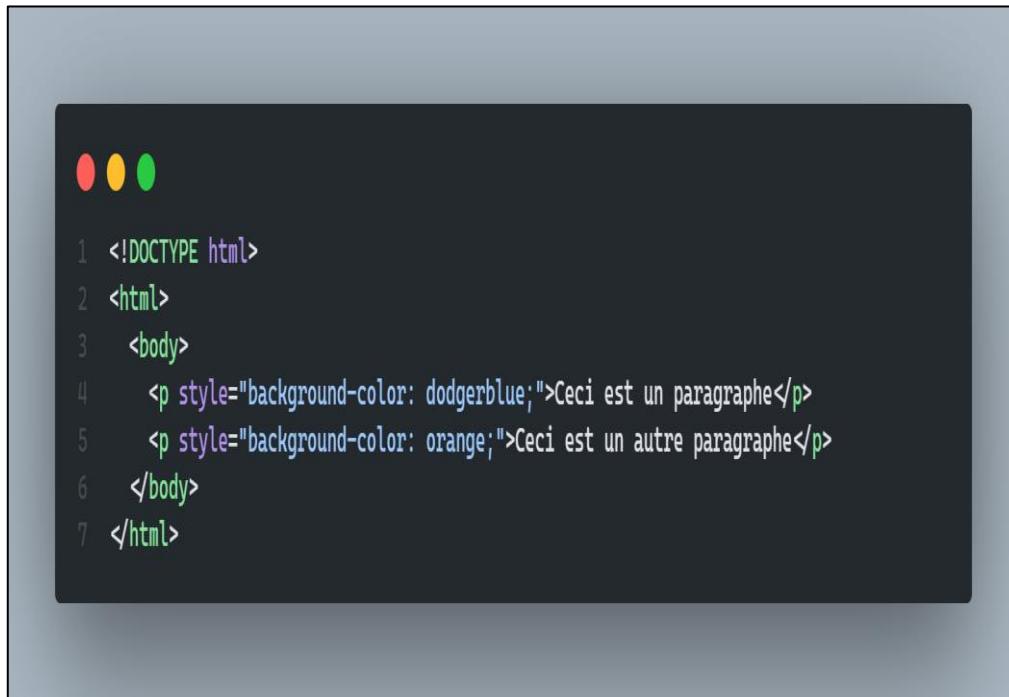
HTML prend en charge 140 noms de couleurs standard.

Couleurs HTML

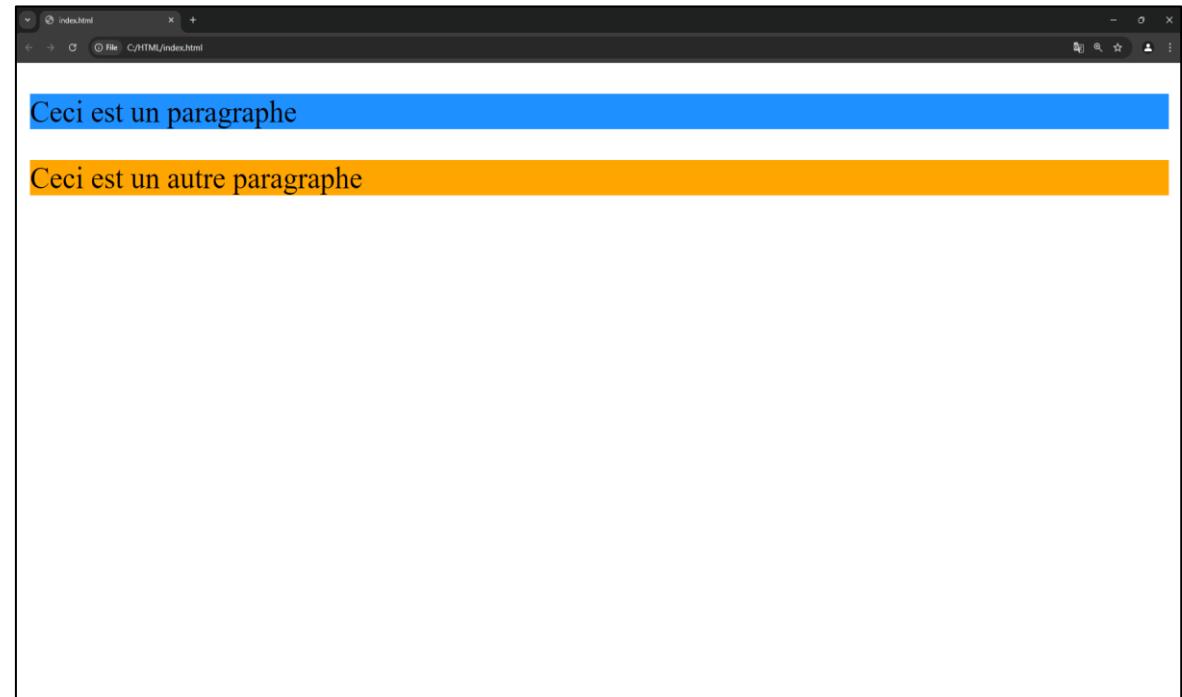
Couleur d'arrière-plan:

Vous pouvez définir la couleur d'arrière-plan des éléments HTML:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p style="background-color: dodgerblue;">Ceci est un paragraphe</p>
5     <p style="background-color: orange;">Ceci est un autre paragraphe</p>
6   </body>
7 </html>
```



Couleurs HTML

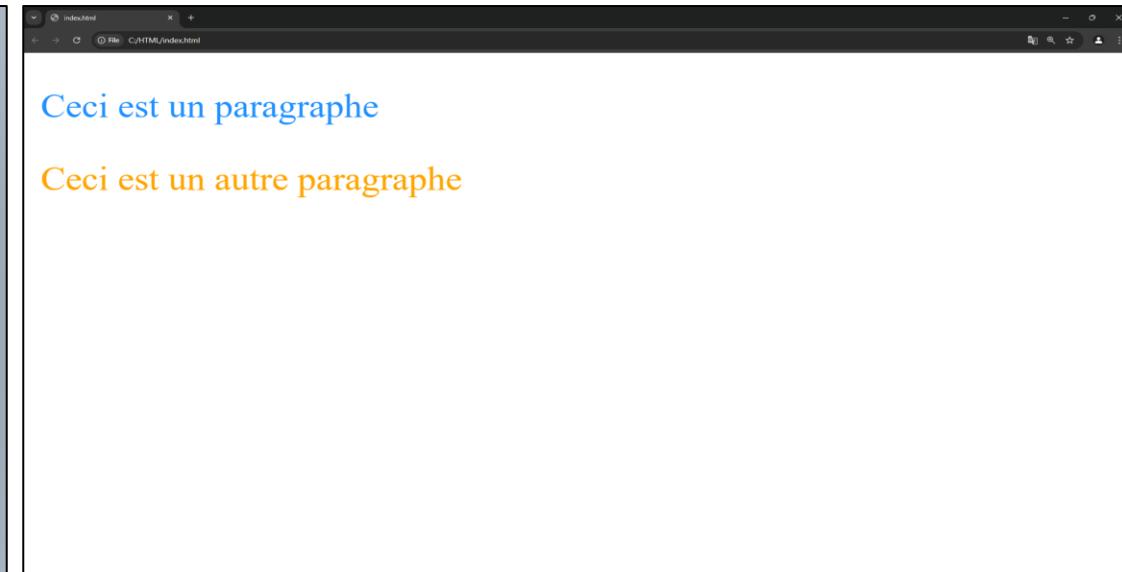
Couleur du texte:

Vous pouvez définir la couleur du texte:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p style="color: dodgerblue;">Ceci est un paragraphe</p>
5     <p style="color: orange;">Ceci est un autre paragraphe</p>
6   </body>
7 </html>
```

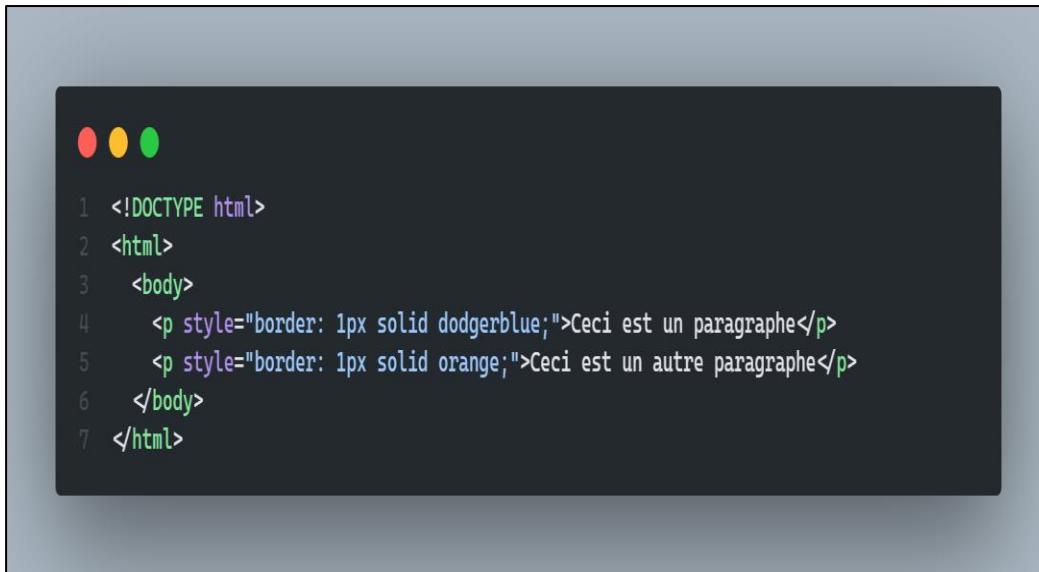


Couleurs HTML

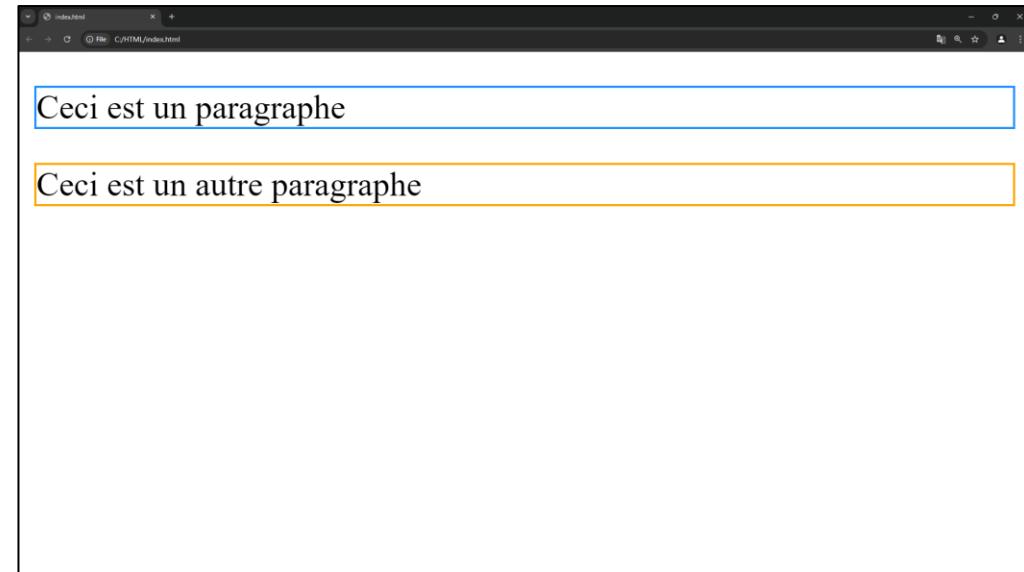
Couleur de bordure:

Vous pouvez définir la couleur des bordures:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p style="border: 1px solid dodgerblue;">Ceci est un paragraphe</p>
5     <p style="border: 1px solid orange;">Ceci est un autre paragraphe</p>
6   </body>
7 </html>
```



Couleurs HTML

Valeurs de couleur:

En HTML, les couleurs peuvent également être spécifiées à l'aide de valeurs RGB, HEX, HSL, RGBA et HSLA.

rgb(255, 99, 71)

#ff6347

hsl(9, 100%, 64%)

rgba(255, 99, 71, 0.5)

hsla(9, 100%, 64%, 0.5)

Couleurs HTML

Couleurs HTML RGB et RGBA:

Une valeur de couleur RGB représente les sources de lumière RED(ROUGE), GREEN(VERT) et BLUE(BLEUE).

Une valeur de couleur RGBA est une extension de RGB avec un canal Alpha (opacité).

Valeurs de couleur RGB:

En HTML, une couleur peut être spécifiée sous forme de valeur RGB, à l'aide de cette formule:

`rgb(red, green, blue)`

Chaque paramètre (red, green, blue) définit l'intensité de la couleur avec une valeur comprise entre 0 et 255.

Cela signifie qu'il y a $256 \times 256 \times 256 = 16777216$ couleurs possibles!

Par exemple, `rgb(255, 0, 0)` s'affiche en rouge, car le rouge est défini sur sa valeur la plus élevée (255) et les deux autres (vert et bleu) sont définis sur 0.

Couleurs HTML

Un autre exemple, `rgb(0, 255, 0)` s'affiche en vert, car le vert est défini sur sa valeur la plus élevée (255) et les deux autres (rouge et bleu) sont

Pour afficher le noir, définissez tous les paramètres de couleur sur 0, comme ceci : `rgb(0, 0, 0)`. définis sur 0.

Pour afficher du blanc, définissez tous les paramètres de couleur sur 255, comme ceci : `rgb(255, 255, 255)`.

Exemples:

`rgb(255, 0, 0)`

`rgb(0, 0, 255)`

`rgb(0, 255, 0)`

`rgb(150, 50, 30)`

`rgb(5, 10, 45)`

`rgb(20, 40, 150)`

`rgb(255, 255, 255)`

`rgb(255, 0, 0)`

Couleurs HTML

Nuances de gris:

Les nuances de gris sont souvent définies à l'aide de valeurs égales pour les trois paramètres :

Exemples:

rgb(60, 60, 60)

rgb(100, 100, 100)

rgb(160, 160, 160)

rgb(180, 180, 180)

rgb(200, 200, 200)

rgb(240, 240, 240)

rgb(140, 140, 140)

rgb(120, 120, 120)

Couleurs HTML

Valeurs de couleur RGBA:

Les valeurs de couleur RGBA sont une extension des valeurs de couleur RGB avec un canal Alpha - qui spécifie l'opacité d'une couleur.

Une valeur de couleur RGBA est spécifiée avec :

`rgba(red, green, blue, alpha)`

Le paramètre alpha est un nombre compris entre 0,0 (entièrement transparent) et 1,0 (pas transparent du tout):

Exemples:

`rgba(255, 99, 71, 0.6)`

`rgba(255, 99, 71, 0.2)`

`rgba(255, 99, 71, 0.4)`

`rgba(255, 99, 71, 0.6)`

`rgba(255, 99, 71, 0.8)`

`rgba(255, 99, 71, 1)`

Couleurs HTML

Couleurs HTML HEX:

Une couleur hexadécimale est spécifiée avec : #RRGGBB, où les entiers hexadécimaux RR RED(ROUGE), GG GREEN(VERT) et BB BLUE(BLEU) spécifient les composants de la couleur.

HEX Color Values:

En HTML, une couleur peut être spécifiée à l'aide d'une valeur hexadécimale sous la forme :

#rrggbb

Où rr (rouge), gg (vert) et bb (bleu) sont des valeurs hexadécimales comprises entre 00 et ff (identiques au nombre décimal 0-255).

Par exemple, #ff0000 s'affiche en rouge, car le rouge est défini sur sa valeur la plus élevée (ff) et les deux autres (vert et bleu) sont définis sur 00.

Autre exemple, #00ff00 s'affiche en vert, car le vert est défini sur sa valeur la plus élevée (ff) et les deux autres (rouge et bleu) sont définis sur 00.

Pour afficher le noir, définissez tous les paramètres de couleur sur 00, comme ceci : #000000. Pour afficher le blanc, définissez tous les paramètres de couleur sur ff, comme ceci : #ffffff.

Couleurs HTML

Exemples:

#ff0000

#0000ff

#3cb371

#ee82ee

#ffa500

#6

#ffff00

#ff00ff

Nuances de gris:

Les nuances de gris sont souvent définies à l'aide de valeurs égales pour les trois paramètres:

Exemples:

#404040

#686868

#a0a0a0

#bebebe

#dcdcdc

#f8f8f8

#cccccc

#dddddd

Couleurs HTML

Couleurs HTML HSL et HSLA:

HSL signifie hue(teinte), saturation(saturation) et lightness(luminosité).

Les valeurs de couleur HSLA sont une extension de HSL avec un canal Alpha (opacité).

Valeurs de couleur HSL:

En HTML, une couleur peut être spécifiée à l'aide de la teinte, de la saturation et de la luminosité (HSL) sous la forme :

`hsl(teinte, saturation, luminosité)`

La teinte est un degré sur la roue chromatique de 0 à 360. 0 est rouge, 120 est vert et 240 est bleu.

La saturation est une valeur en pourcentage. 0 % signifie une nuance de gris et 100 % est la couleur complète.

La luminosité est également une valeur en pourcentage. 0 % est noir et 100 % est blanc.

Couleurs HTML

Exemples:

`hsl(0, 100%, 50%)`

`hsl(240, 100%, 50%)`

`hsl(147, 50%, 47%)`

`hsl(300, 76%, 72%)`

`hsl(39, 100%, 50%)`

`hsl(248, 53%, 58%)`

Saturation

La saturation peut être décrite comme l'intensité d'une couleur.

100 % correspond à une couleur pure, sans nuances de gris.

50 % correspond à 50 % de gris, mais vous pouvez toujours voir la couleur.

0 % correspond à un gris complet ; vous ne pouvez plus voir la couleur.

Exemples:

`hsl(0, 100%, 50%)`

`hsl(0, 80%, 50%)`

`hsl(0, 60%, 50%)`

`hsl(0, 40%, 50%)`

`hsl(0, 20%, 50%)`

`hsl(0, 0%, 50%)`

Couleurs HTML

Luminosité:

La luminosité d'une couleur peut être décrite comme la quantité de lumière que vous souhaitez donner à la couleur, où 0 % signifie aucune lumière (noir), 50 % signifie 50 % de lumière (ni foncé ni clair) et 100 % signifie une luminosité totale (blanc).

Exemples:

`hsl(0, 100%, 0%)`

`hsl(0, 100%, 75%)`

`hsl(0, 100%, 25%)`

`hsl(0, 100%, 90%)`

`hsl(0, 100%, 50%)`

`hsl(0, 100%, 100%)`

Nuances de gris:

Les nuances de gris sont souvent définies en définissant la teinte et la saturation sur 0, et en ajustant la luminosité de 0 % à 100 % pour obtenir des nuances plus foncées/plus claires:

Exemples:

`hsl(0, 0%, 20%)`

`hsl(0, 0%, 60%)`

`hsl(0, 0%, 30%)`

`hsl(0, 0%, 70%)`

`hsl(0, 0%, 40%)`

`hsl(0, 0%, 90%)`

Couleurs HTML

Valeurs de couleur HSLA:

Les valeurs de couleur HSLA sont une extension des valeurs de couleur HSL, avec un canal Alpha - qui spécifie l'opacité d'une couleur.

Une valeur de couleur HSLA est spécifiée avec :

`hsla(teinte, saturation, luminosité, alpha)`

Le paramètre alpha est un nombre compris entre 0,0 (entièrement transparent) et 1,0 (pas transparent du tout):

Exemples:

`hsla(9, 100%, 64%, 0)`

`hsla(9, 100%, 64%, 0.6)`

`hsla(9, 100%, 64%, 0.2)`

`hsla(9, 100%, 64%, 0.8)`

`hsla(9, 100%, 64%, 0.4)`

`hsla(9, 100%, 64%, 1)`

HTML CSS

CSS signifie Cascading Style Sheets(feuilles de style en cascade).

Le CSS permet d'économiser beaucoup de travail. Il permet de contrôler la mise en page de plusieurs pages Web à la fois.

Qu'est-ce que le CSS?

Les feuilles de style en cascade (CSS) sont utilisées pour formater la mise en page d'une page Web.

Avec CSS, vous pouvez contrôler la couleur, la police, la taille du texte, l'espacement entre les éléments, la manière dont les éléments sont positionnés et disposés, les images d'arrière-plan ou les couleurs d'arrière-plan à utiliser, les différents affichages pour différents appareils et tailles d'écran, et bien plus encore !

Astuce : le terme "cascade" signifie qu'un style appliqué à un élément parent s'appliquera également à tous les éléments enfants du parent. Ainsi, si vous définissez la couleur du corps du texte sur "bleu", tous les titres, paragraphes et autres éléments de texte du corps auront également la même couleur (sauf si vous spécifiez autre chose) !

HTML CSS

Utilisation de CSS:

Vous pouvez ajouter du CSS aux documents HTML de trois manières :

- **En ligne** : en utilisant l'attribut `style` dans les éléments HTML
- **En interne** : en utilisant un élément `<style>` dans la section `<head>`
- **En externe** : en utilisant un élément `<link>` pour créer un lien vers un fichier CSS externe

La manière la plus courante d'ajouter du CSS consiste à conserver les styles dans des fichiers CSS externes. Cependant, dans ce didacticiel, nous utiliserons des styles en ligne et internes, car cela est plus facile à démontrer et plus facile à essayer par vous-même.

HTML CSS

CSS en ligne

Un CSS en ligne permet d'appliquer un style unique à un seul élément HTML.

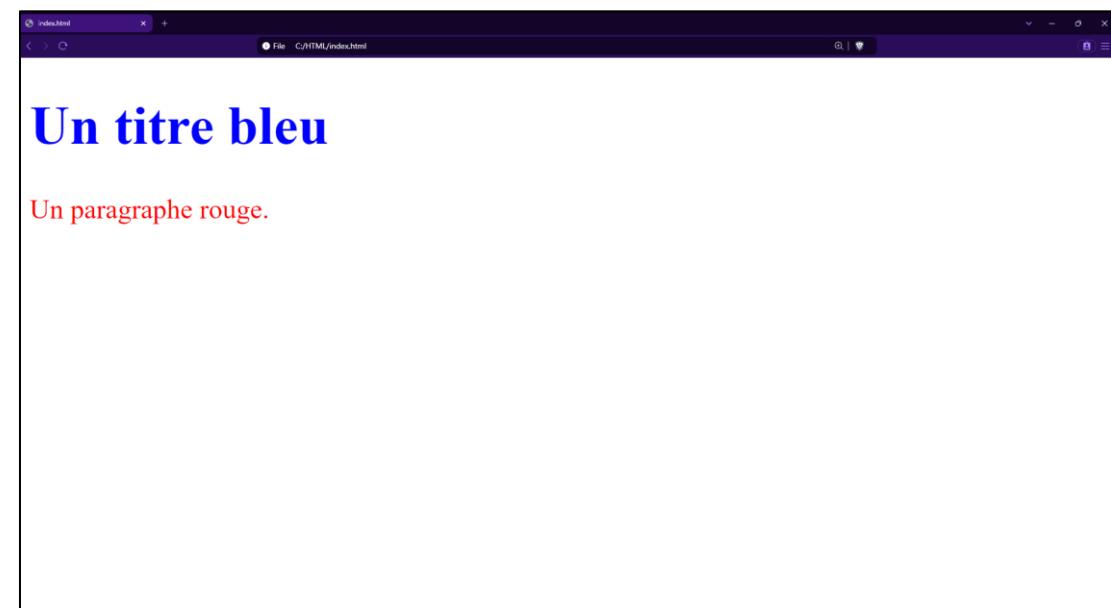
Un CSS en ligne utilise l'attribut **style** d'un élément HTML.

L'exemple suivant définit la couleur du texte de l'élément **<h1>** sur bleu et la couleur du texte de l'élément **<p>** sur rouge:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1 style="color: blue">Un titre bleu</h1>
5     <p style="color: red">Un paragraphe rouge.</p>
6   </body>
7 </html>
```



HTML CSS

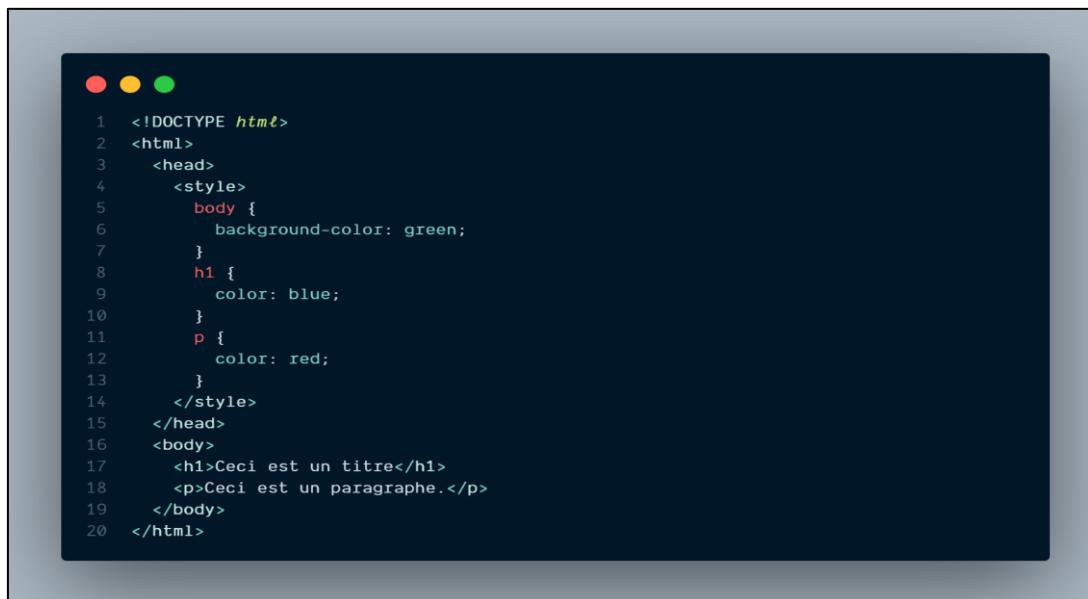
CSS interne:

Un CSS interne est utilisé pour définir un style pour une seule page HTML.

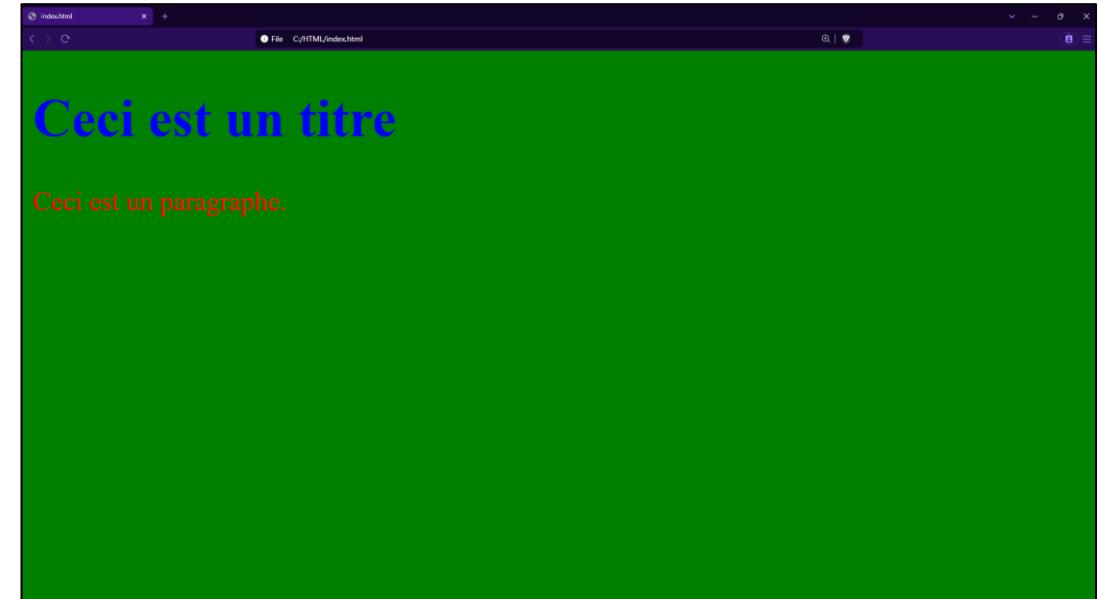
Un CSS interne est défini dans la section `<head>` d'une page HTML, dans un élément `<style>`.

L'exemple suivant définit la couleur du texte de TOUS les éléments `<h1>` (sur cette page) sur bleu et la couleur du texte de TOUS les éléments `<p>` sur rouge. De plus, la page s'affichera avec une couleur d'arrière-plan "vert":

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       body {
6         background-color: green;
7       }
8       h1 {
9         color: blue;
10      }
11      p {
12        color: red;
13      }
14    </style>
15  </head>
16  <body>
17    <h1>Ceci est un titre</h1>
18    <p>Ceci est un paragraphe.</p>
19  </body>
20 </html>
```



HTML CSS

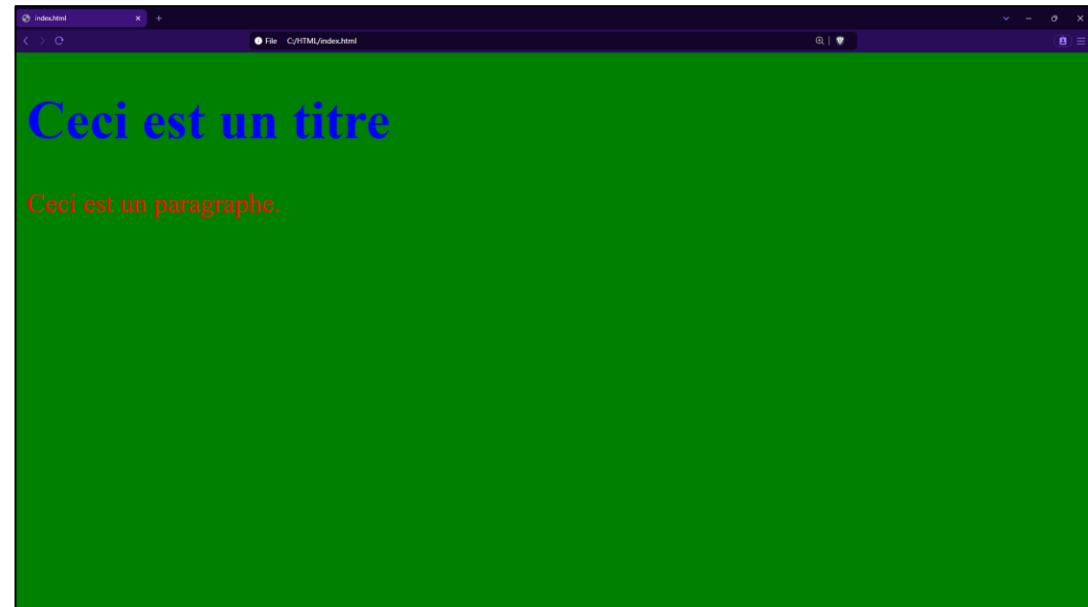
CSS externe:

Une feuille de style externe est utilisée pour définir le style de nombreuses pages HTML.

Pour utiliser une feuille de style externe, ajoutez un lien vers celle-ci dans la section `<head>` de chaque page HTML:

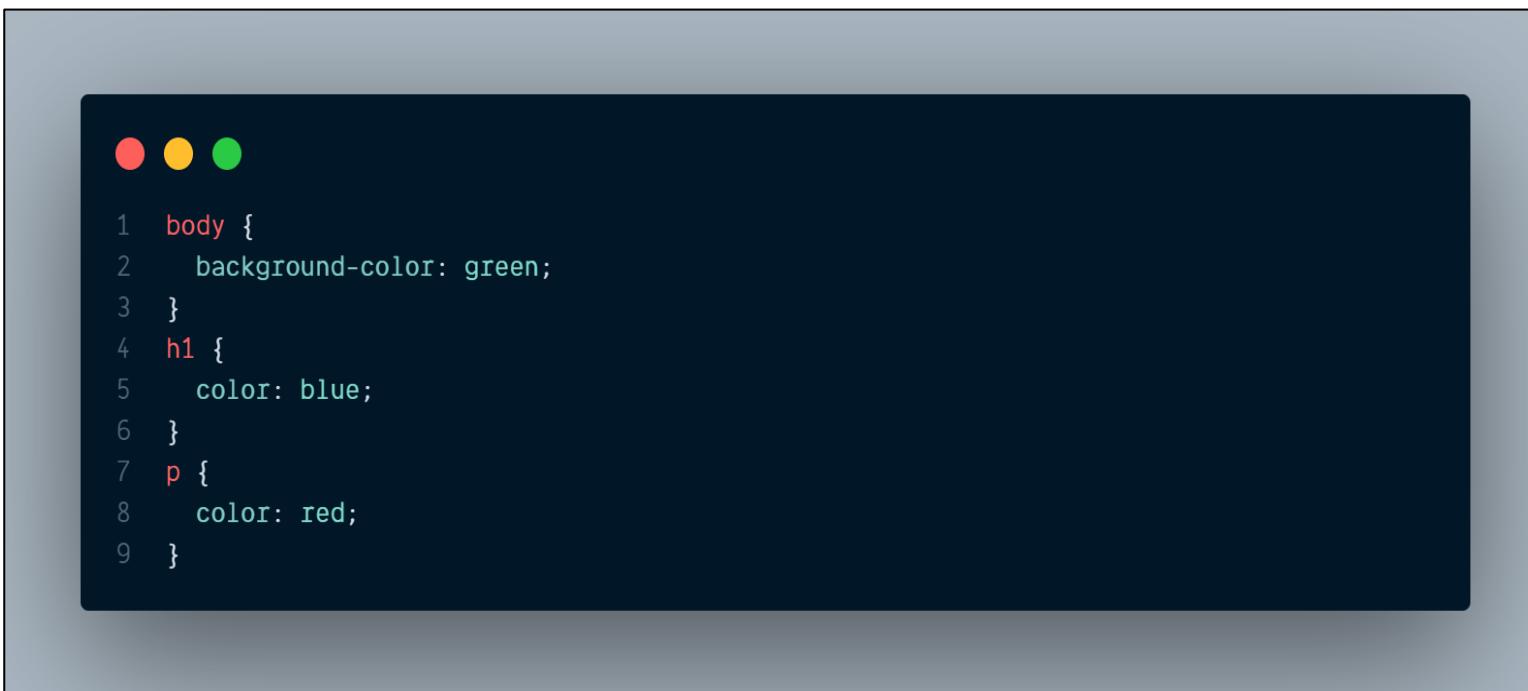
Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" href="styles.css" />
5   </head>
6   <body>
7     <h1>Ceci est un titre</h1>
8     <p>Ceci est un paragraphe.</p>
9   </body>
10 </html>
```



La feuille de style externe peut être écrite dans n'importe quel éditeur de texte. Le fichier ne doit contenir aucun code HTML et doit être enregistré avec une extension .css.

Voici à quoi ressemble le fichier "styles.css":



```
● ● ●  
1 body {  
2   background-color: green;  
3 }  
4 h1 {  
5   color: blue;  
6 }  
7 p {  
8   color: red;  
9 }
```

Astuce: avec une feuille de style externe, vous pouvez modifier l'apparence d'un site Web entier en modifiant un seul fichier !

HTML CSS

Couleurs, polices et tailles CSS:

Nous allons ici vous présenter quelques propriétés CSS couramment utilisées.

- La propriété CSS `color` définit la couleur du texte à utiliser.
- La propriété CSS `font-family` définit la police à utiliser.
- La propriété CSS `font-size` définit la taille du texte à utiliser.

Exemple: Utilisation des propriétés CSS color, font-family et font-size :



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       h1 {
6         color: blue;
7         font-family: cursive;
8         font-size: 100px;
9       }
10      p {
11        color: red;
12        font-family: Arial, Helvetica, sans-serif;
13        font-size: 50px;
14      }
15    </style>
16  </head>
17  <body>
18    <h1>Ceci est un titre</h1>
19    <p>Ceci est un paragraphe.</p>
20  </body>
21 </html>
```



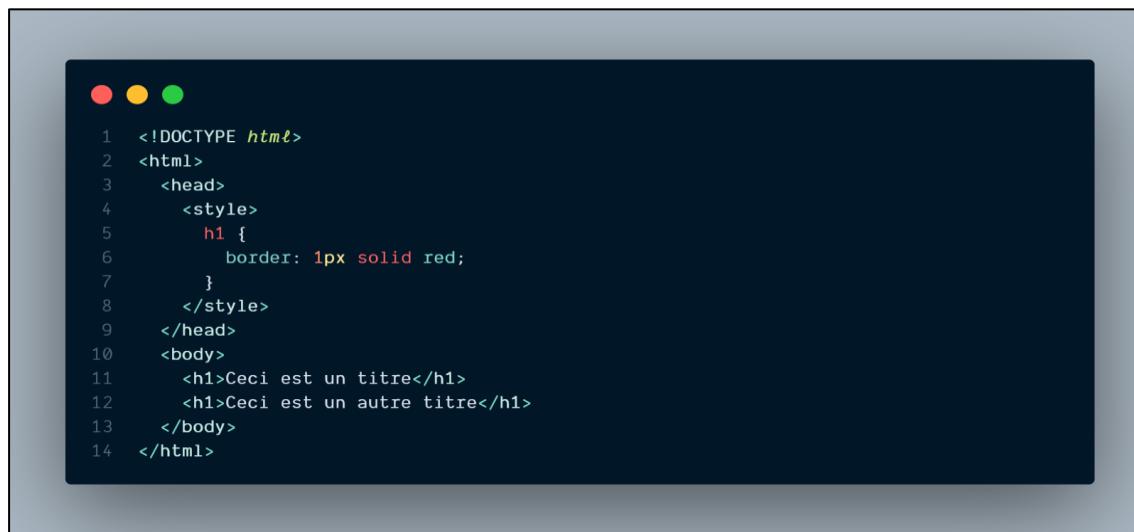
HTML CSS

CSS Border:

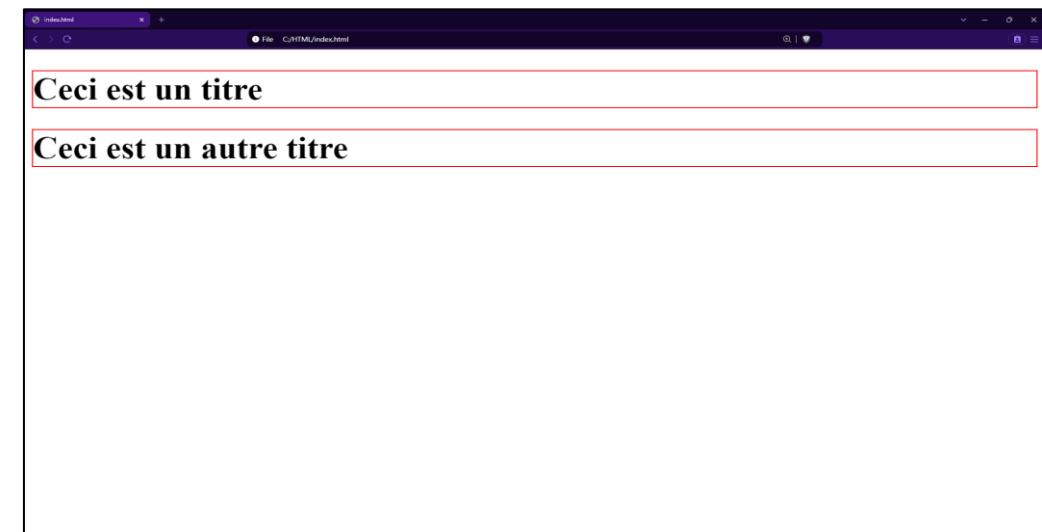
La propriété CSS border définit une bordure autour d'un élément HTML.

Astuce: vous pouvez définir une bordure pour presque tous les éléments HTML.

Exemple: Utilisation de la propriété CSS border:



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       h1 {
6         border: 1px solid red;
7       }
8     </style>
9   </head>
10  <body>
11    <h1>Ceci est un titre</h1>
12    <h1>Ceci est un autre titre</h1>
13  </body>
14 </html>
```



HTML CSS

CSS Padding:

La propriété CSS padding définit un remplissage (espace) entre le texte et la bordure.

Exemple: Utilisation des propriétés CSS border et padding:

The image displays two windows side-by-side. On the left is a dark-themed code editor showing the following HTML and CSS code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       h1 {
6         padding: 40px;
7         border: 1px solid red;
8       }
9     </style>
10   </head>
11   <body>
12     <h1>Ceci est un titre</h1>
13     <h1>Ceci est un autre titre</h1>
14   </body>
15 </html>
```

On the right is a browser window titled "index.html" showing the rendered HTML. It contains two

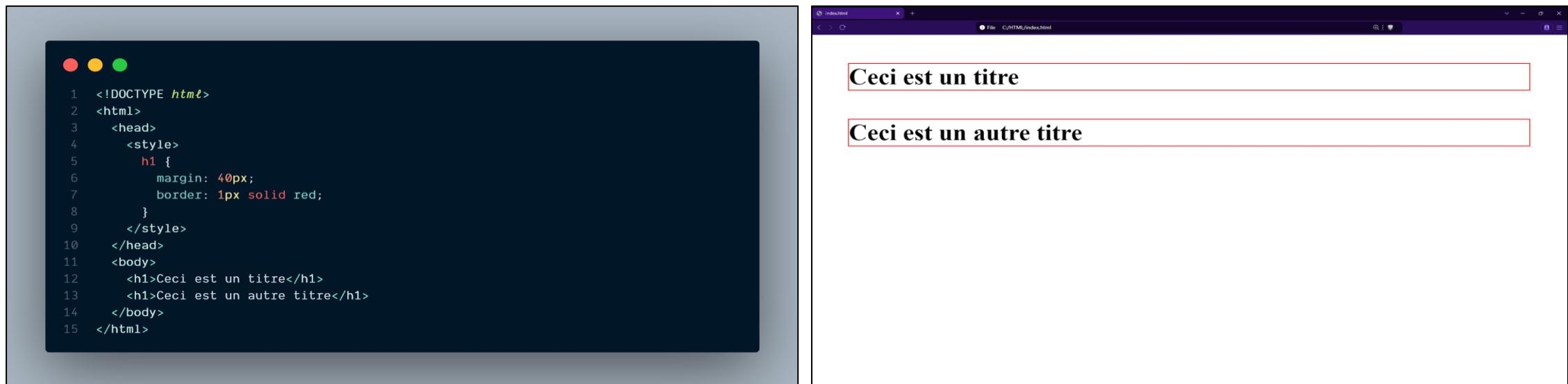
elements. The first has a red border and 40px of padding, resulting in a large gap from the top edge. The second is positioned directly below it without any additional margin or padding.

HTML CSS

CSS Margin:

La propriété CSS margin définit une marge (espace) à l'extérieur de la bordure.

Exemple: Utilisation des propriétés CSS border et margin:



The image shows a code editor window on the left and a browser window on the right. The code editor displays the following HTML and CSS code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       h1 {
6         margin: 40px;
7         border: 1px solid red;
8       }
9     </style>
10   </head>
11   <body>
12     <h1>Ceci est un titre</h1>
13     <h1>Ceci est un autre titre</h1>
14   </body>
15 </html>
```

The browser window shows two **h1** elements. The first one has a top margin of 40px and a red border. The second one is positioned directly below it without any additional margin or border.

HTML CSS

Lien vers un CSS externe:

Les feuilles de style externes peuvent être référencées avec une URL complète ou avec un chemin relatif à la page Web actuelle.

Exemple: Cet exemple utilise une URL complète pour créer un lien vers une feuille de style :

```
<link rel="stylesheet" href="https://www.example.com.project/css/styles.css" />
```

Exemple: Cet exemple renvoie vers une feuille de style située dans le dossier css:

```
<link rel="stylesheet" href="css/styles.css" />
```

Exemple: Cet exemple renvoie vers une feuille de style située dans le même dossier que la page actuelle :

```
<link rel="stylesheet" href="styles.css" />
```

Liens HTML

On trouve des liens dans presque toutes les pages Web. Les liens permettent aux utilisateurs de cliquer pour passer d'une page à l'autre.

Liens HTML - Hyperliens:

Les liens HTML sont des hyperliens.

Vous pouvez cliquer sur un lien et accéder à un autre document.

Lorsque vous déplacez la souris sur un lien, la flèche de la souris se transforme en petite main.

Remarque: un lien ne doit pas nécessairement être du texte. Un lien peut être une image ou tout autre élément HTML !

Liens HTML

Liens HTML - Syntaxe:

La balise HTML `<a>` définit un lien hypertexte. Sa syntaxe est la suivante:

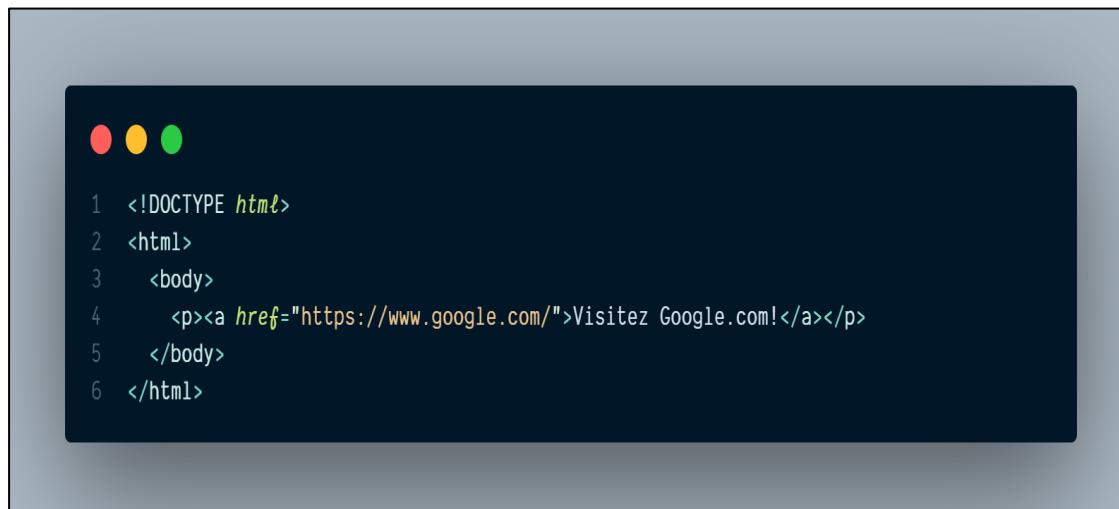
```
<a href="url">link text</a>
```

L'attribut le plus important de l'élément `<a>` est l'attribut `href`, qui indique la destination du lien.

Le texte du lien est la partie qui sera visible pour le lecteur.

En cliquant sur le texte du lien, le lecteur sera redirigé vers l'adresse URL spécifiée.

Exemple: Cet exemple montre comment créer un lien vers Google



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p><a href="https://www.google.com/">Visitez Google.com!</a></p>
5   </body>
6 </html>
```



Liens HTML

Par défaut, les liens s'affichent comme suit dans tous les navigateurs :

- Un lien non visité est souligné et bleu
- Un lien visité est souligné et violet
- Un lien actif est souligné et rouge

Astuce: Les liens peuvent bien sûr être stylisés avec CSS, pour un autre look !

Liens HTML

Liens HTML - L'attribut target:

Par défaut, la page liée s'affiche dans la fenêtre de navigateur actuelle. Pour modifier cela, vous devez spécifier une autre cible pour le lien.

L'attribut **target** spécifie où ouvrir le document lié.

L'attribut **target** peut avoir l'une des valeurs suivantes :

- **_self** - Par défaut. Ouvre le document dans la même fenêtre/onglet que celui sur lequel vous avez cliqué
- **_blank** - Ouvre le document dans une nouvelle fenêtre ou un nouvel onglet
- **_parent** - Ouvre le document dans le cadre parent
- **_top** - Ouvre le document dans le corps entier de la fenêtre

Liens HTML

URL absolues et URL relatives:

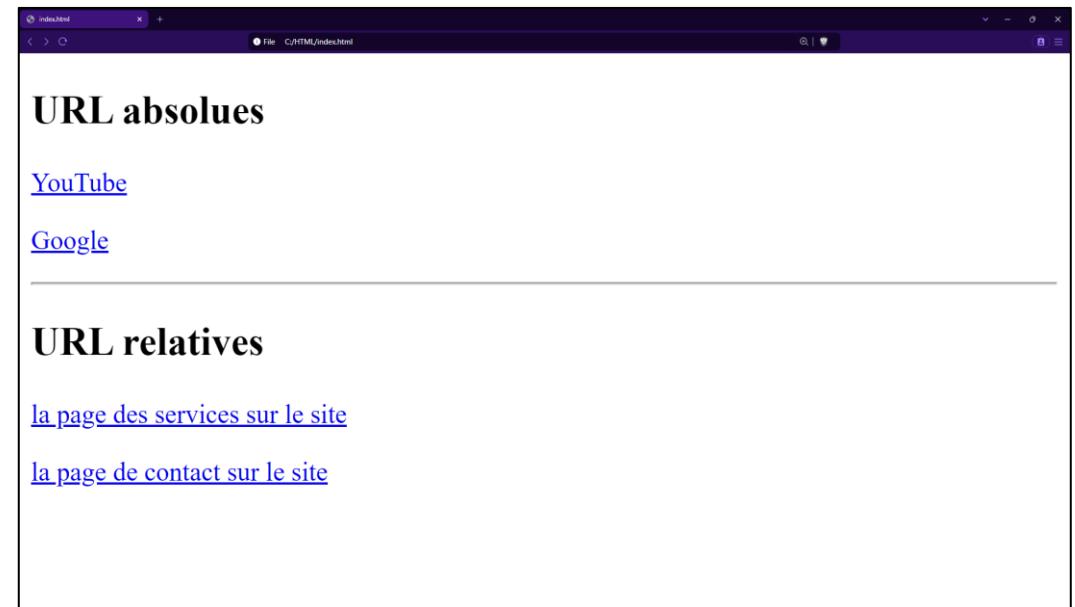
Dans l'exemple précédent, nous utilisons une URL absolue (une adresse Web complète) dans l'attribut href.

Un lien local (un lien vers une page du même site Web) est spécifié avec une URL relative (sans la partie "https://www"):

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>URL absolues</h2>
5     <p><a href="https://www.youtube.org/">YouTube</a></p>
6     <p><a href="https://www.google.com/">Google</a></p>
7     <hr />
8     <h2>URL relatives</h2>
9     <p><a href="services.html">la page des services sur le site</a></p>
10    <p><a href="contact.html">la page de contact sur le site</a></p>
11  </body>
12 </html>
```



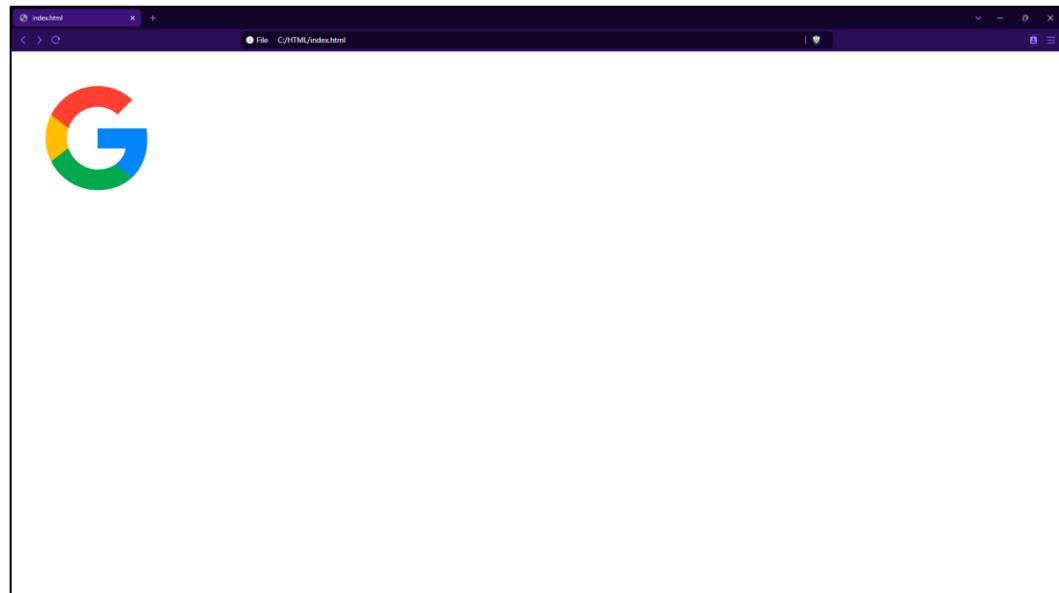
Liens HTML

Liens HTML - Utiliser une image comme lien:

Pour utiliser une image comme lien, placez simplement la balise `` à l'intérieur de la balise `<a>`:

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <a href="https://www.google.com">
5       
6     </a>
7   </body>
8 </html>
```



Liens HTML

Lien vers une adresse e-mail:

Utilisez **mailto:** dans l'attribut **href** pour créer un lien qui ouvre le programme de messagerie de l'utilisateur (pour lui permettre d'envoyer un nouvel e-mail):

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p><a href="mailto:utilisateur@example.com">Envoyer un e-mail</a></p>
5   </body>
6 </html>
```

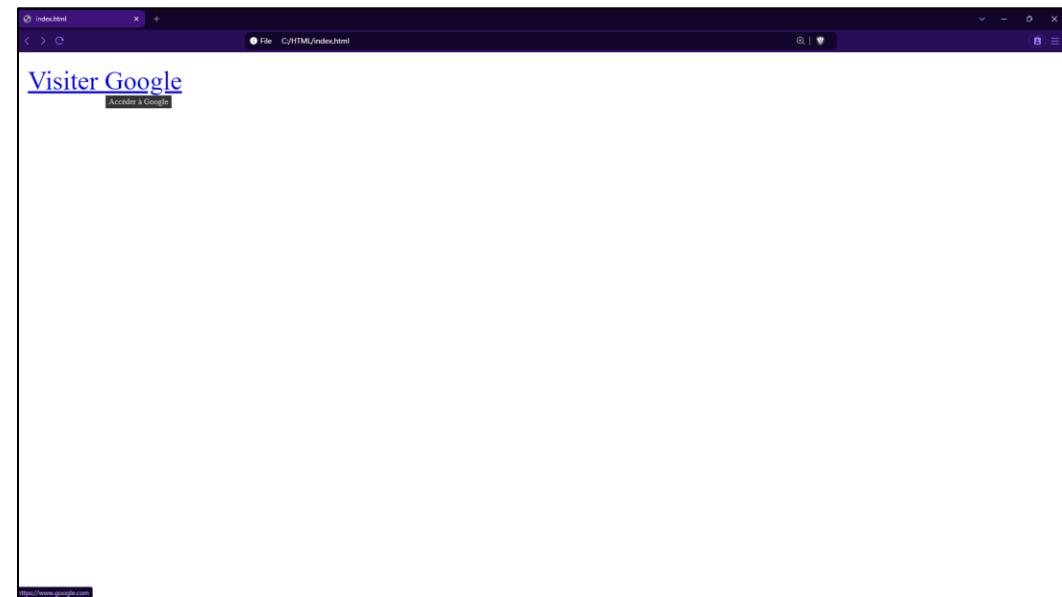
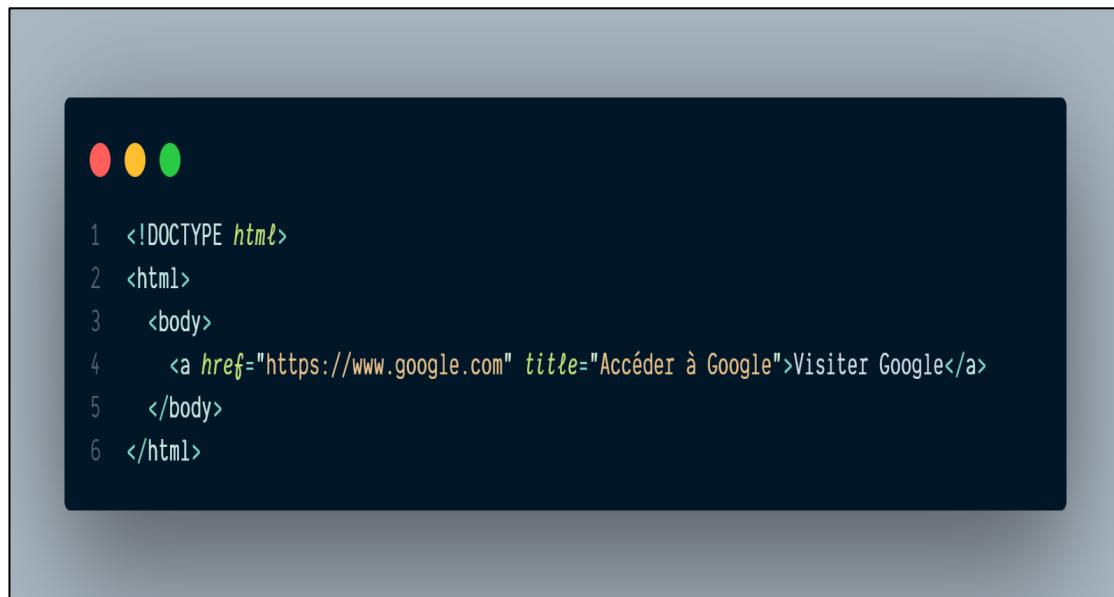


Liens HTML

Titres des liens:

L'attribut **title** spécifie des informations supplémentaires sur un élément. Les informations sont généralement affichées sous forme de texte d'info-bulle lorsque la souris passe sur l'élément.):

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <a href="https://www.google.com" title="Accéder à Google">Visiter Google</a>
5   </body>
6 </html>
```

The image shows two screenshots side-by-side. On the left is a code editor window displaying an HTML file named 'index.html'. The code contains a single anchor tag with a href attribute pointing to Google and a title attribute providing a tooltip message. On the right is a web browser window showing the rendered page. The link 'Visiter Google' is visible, and a mouse cursor hovering over it triggers a tooltip containing the text 'Accéder à Google'.

Images HTML

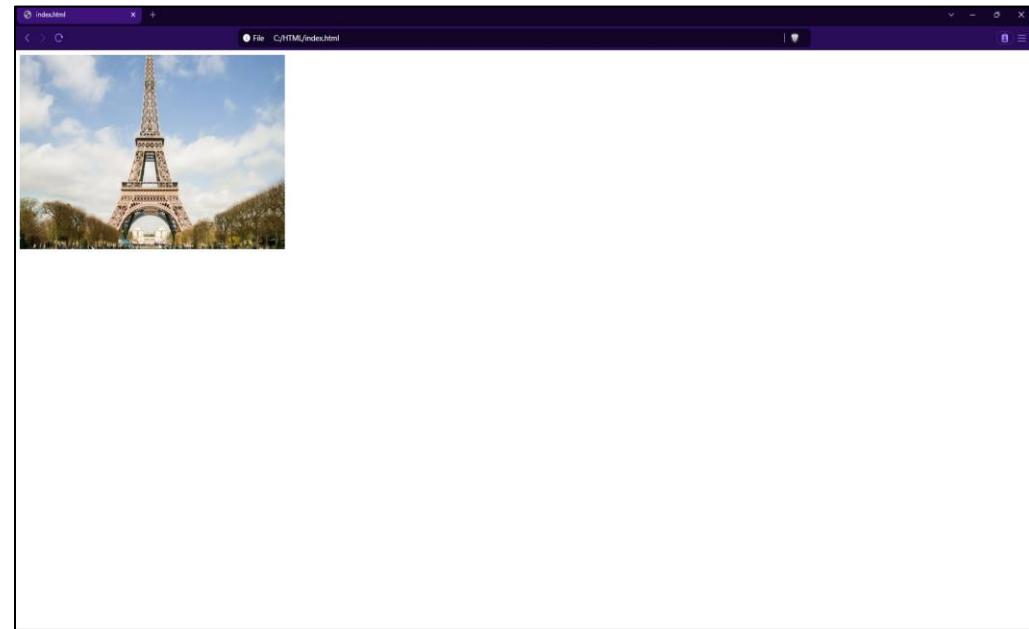
Les images peuvent améliorer la conception et l'apparence d'une page Web.



Images HTML

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```



Images HTML

Syntaxe des images HTML:

La balise HTML `` est utilisée pour intégrer une image dans une page Web.

Les images ne sont pas techniquement insérées dans une page Web ; les images sont liées à des pages Web. La balise `` crée un espace de stockage pour l'image référencée.

La balise `` est vide, elle ne contient que des attributs et n'a pas de balise de fermeture.

La balise `` a deux attributs obligatoires :

- `src` - Spécifie le chemin d'accès à l'image
- `alt` - Spécifie un texte alternatif pour l'image

Syntaxe:

```

```

Images HTML

L'attribut `src`:

L'attribut `src` requis spécifie le chemin (URL) vers l'image.

Remarque : lorsqu'une page Web se charge, c'est le navigateur qui récupère l'image à partir d'un serveur Web et l'insère dans la page. Par conséquent, assurez-vous que l'image reste au même endroit par rapport à la page Web, sinon vos visiteurs obtiendront une icône de lien brisé. L'icône de lien brisé et le texte alternatif `alt` s'affichent si le navigateur ne trouve pas l'image.

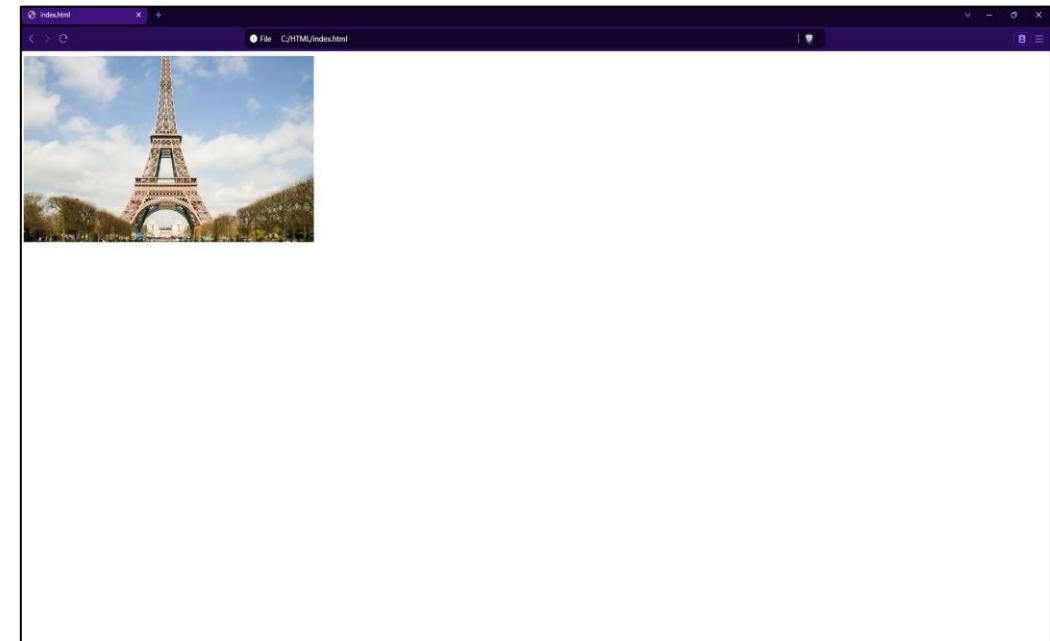
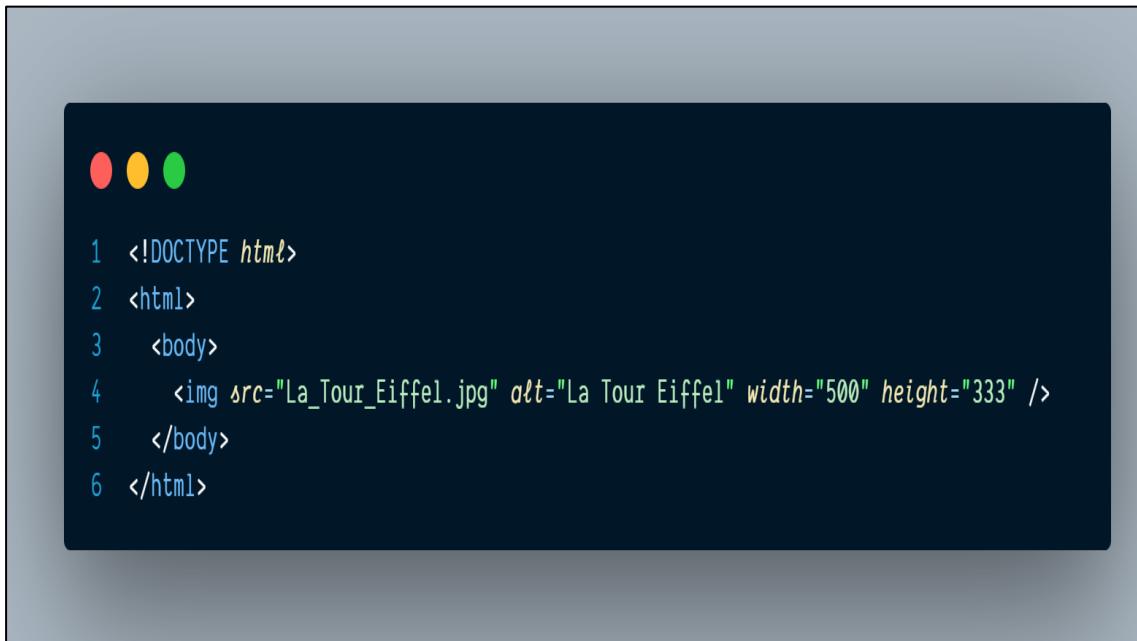
L'attribut `alt`:

L'attribut `alt` obligatoire fournit un texte alternatif pour une image, si l'utilisateur ne peut pas la visualiser pour une raison quelconque (en raison d'une connexion lente, d'une erreur dans l'attribut `src` ou si l'utilisateur utilise un lecteur d'écran).

La valeur de l'attribut `alt` doit décrire l'image

Images HTML

Exemple:

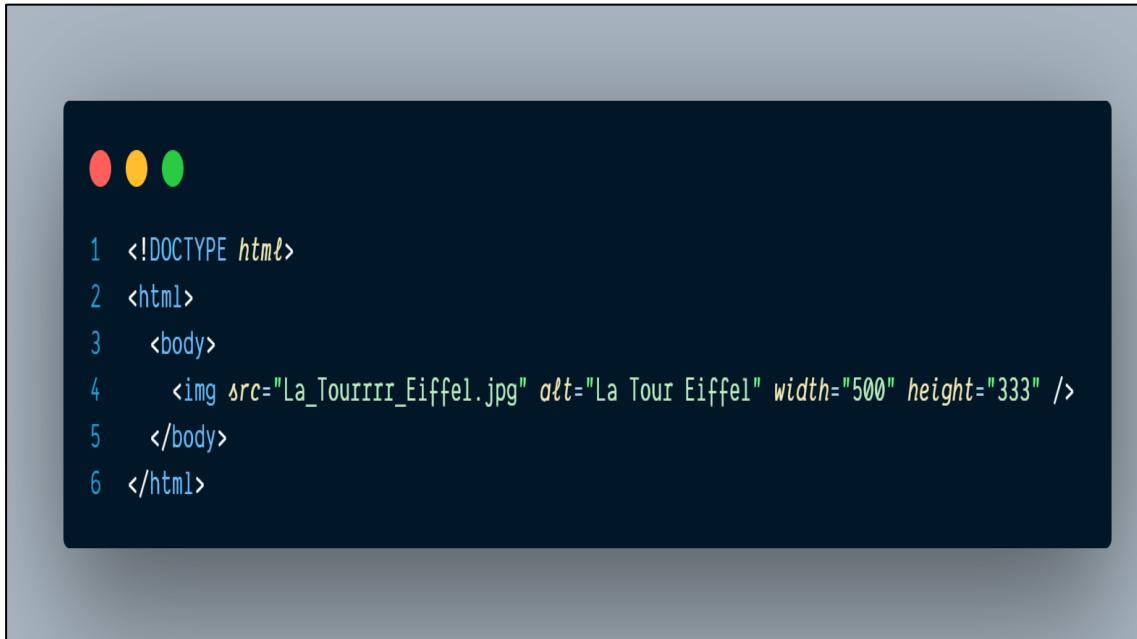


```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```

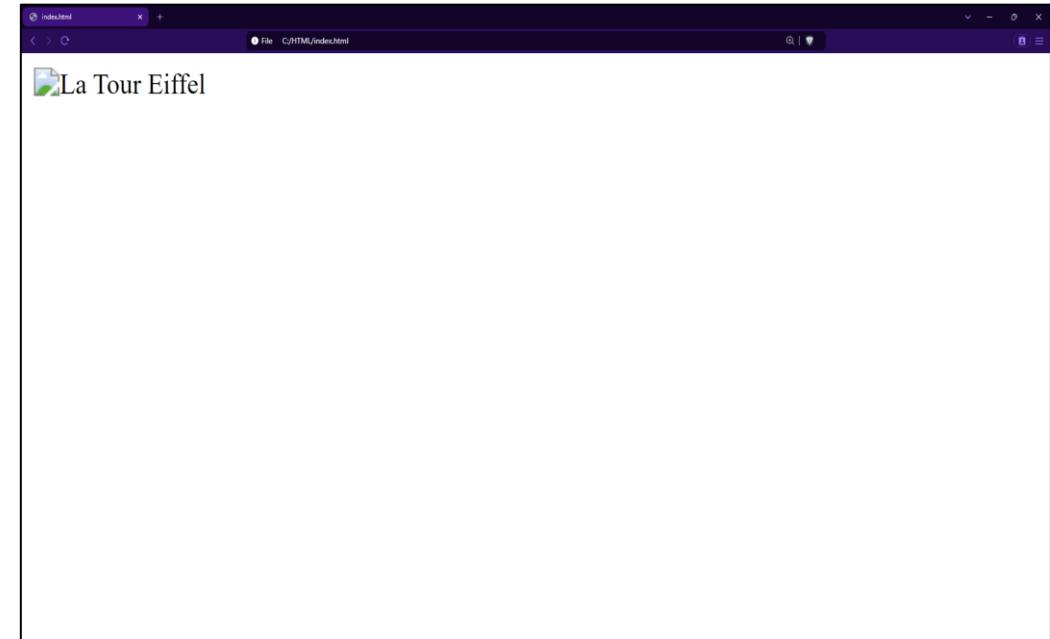
Si un navigateur ne trouve pas d'image, il affichera la valeur de l'attribut **alt**:

Images HTML

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```

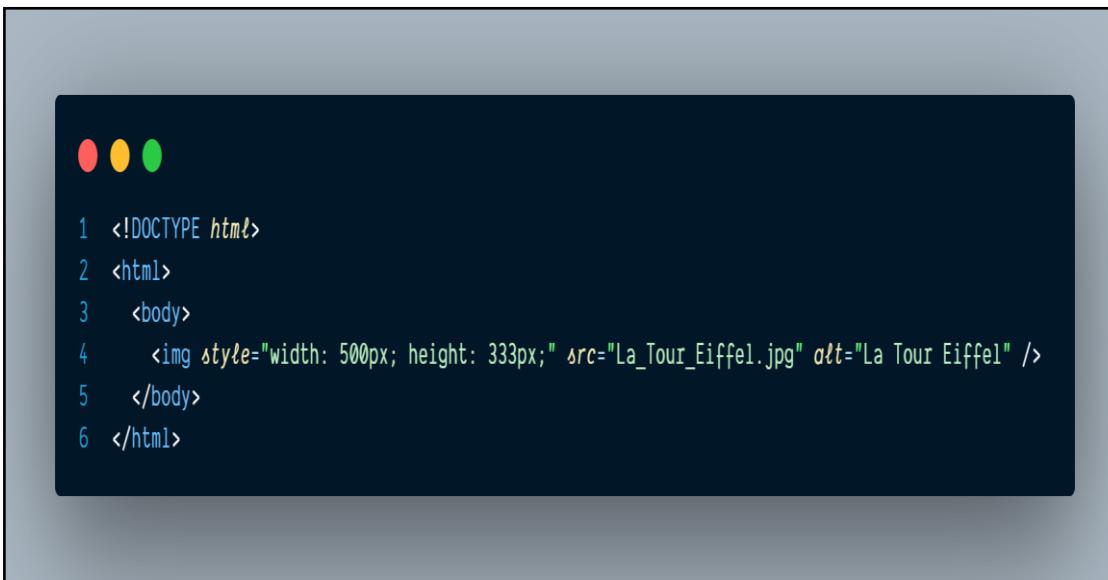


Images HTML

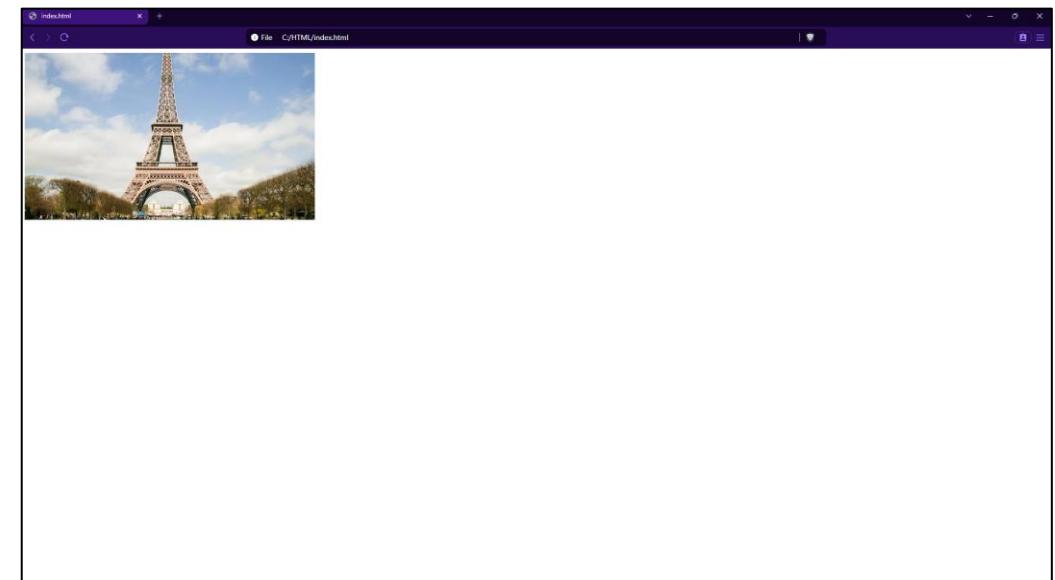
Taille de l'image - Largeur et hauteur:

Vous pouvez utiliser l'attribut **style** pour spécifier la largeur et la hauteur d'une image.

Exemple:



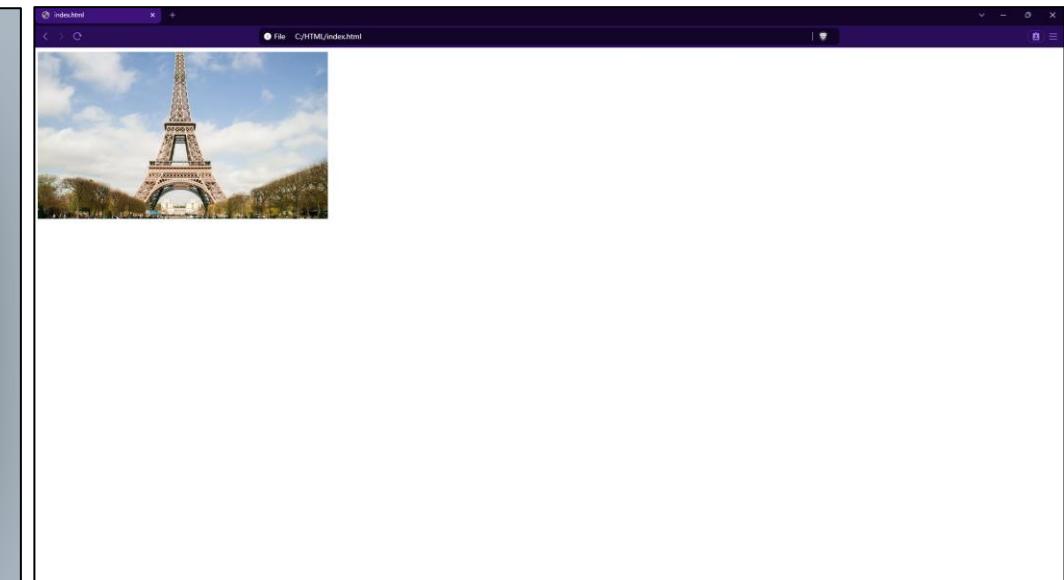
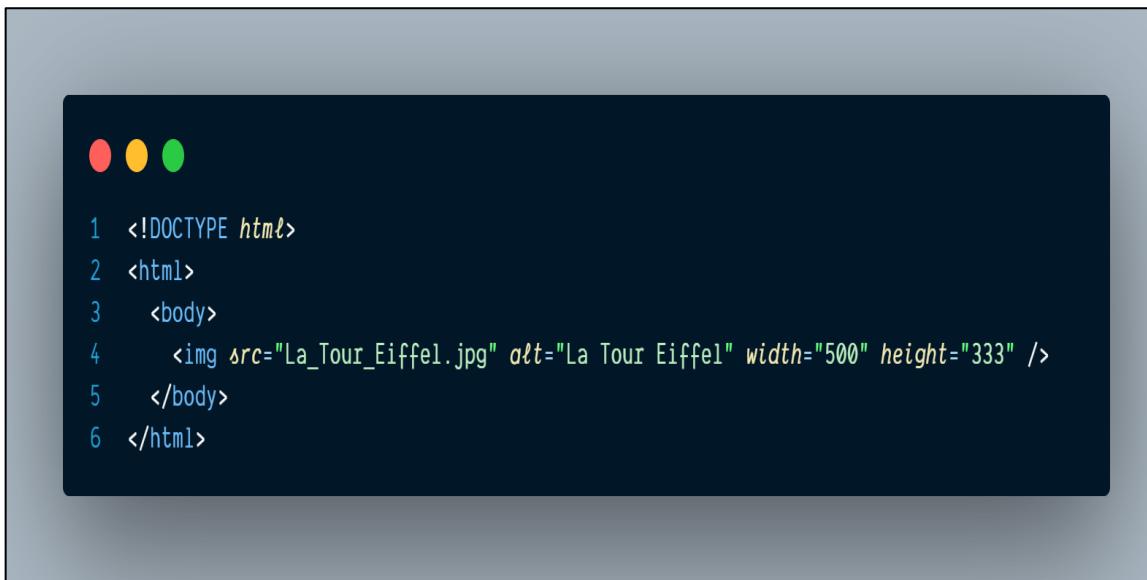
```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```



Alternativement, vous pouvez utiliser les attributs **width** et **height** :

Images HTML

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```

Les attributs **width** et **height** définissent toujours la largeur et la hauteur de l'image en **pixels**.

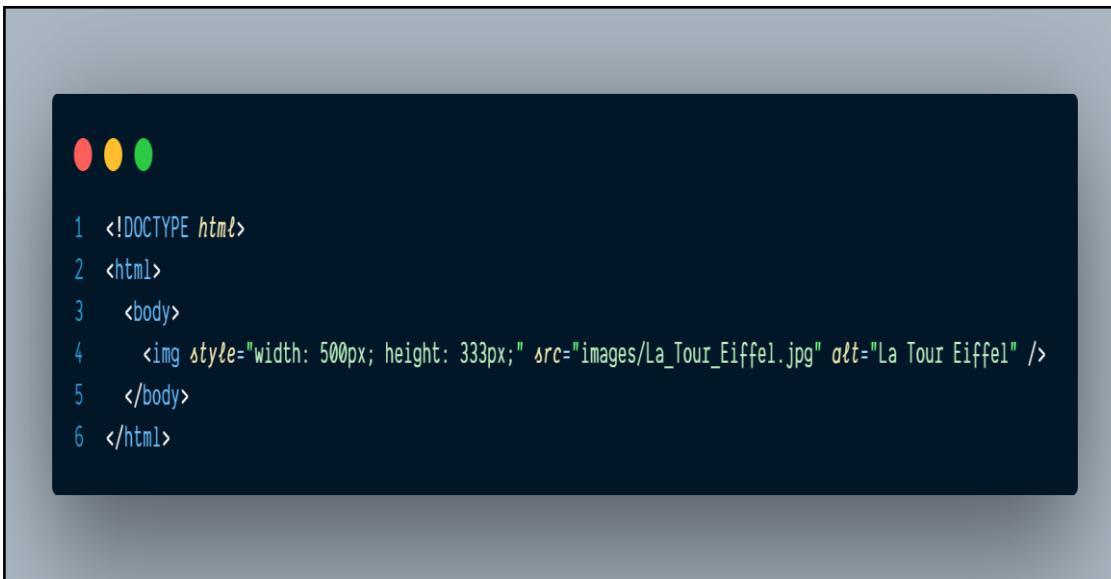
Remarque: indiquez toujours la largeur et la hauteur d'une image. Si la largeur et la hauteur ne sont pas spécifiées, la page Web peut scintiller pendant le chargement de l'image.

Images HTML

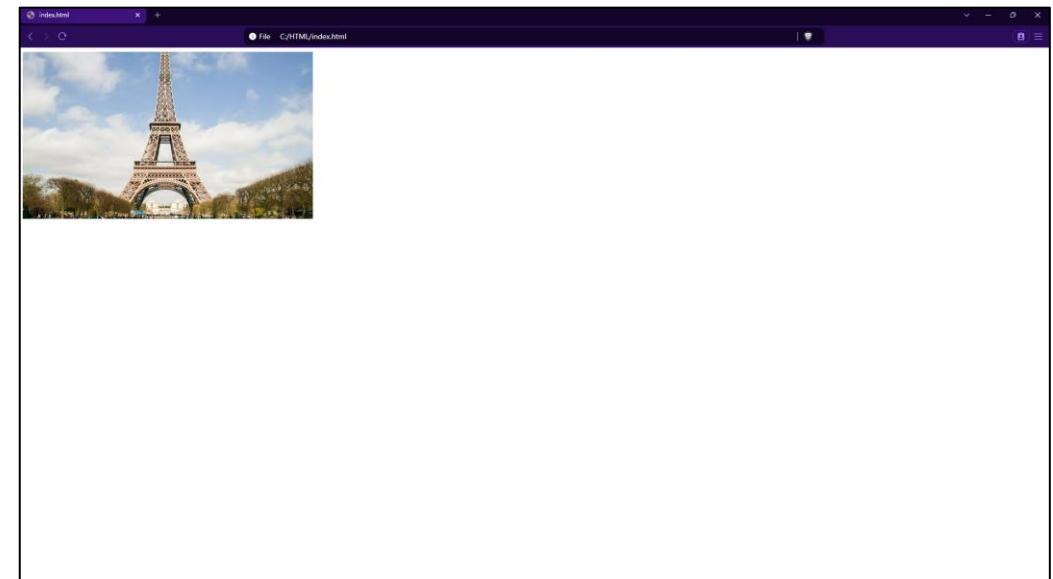
Images dans un autre dossier:

Si vous avez vos images dans un sous-dossier, vous devez inclure le nom du dossier dans l'attribut src :

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```

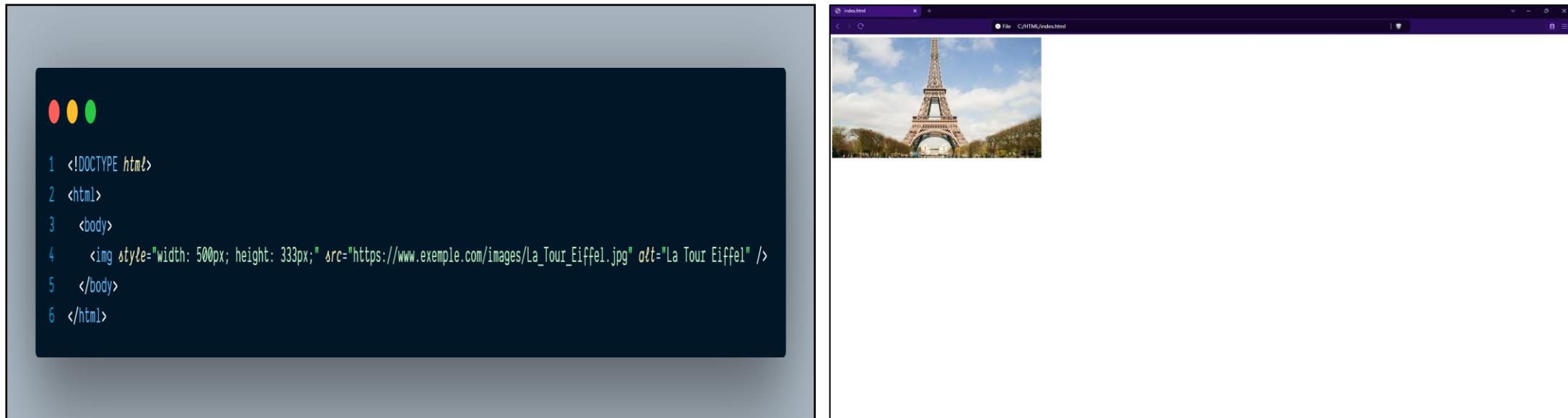


Images HTML

Images sur un autre serveur/site Web:

Pour pointer vers une image sur un autre serveur, vous devez spécifier une URL absolue (complète) dans l'attribut src:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```

Remarques sur les images externes: les images externes peuvent être protégées par des droits d'auteur. Si vous n'obtenez pas l'autorisation de les utiliser, vous risquez de violer les lois sur les droits d'auteur. De plus, vous ne pouvez pas contrôler les images externes ; elles peuvent être soudainement supprimées ou modifiées.

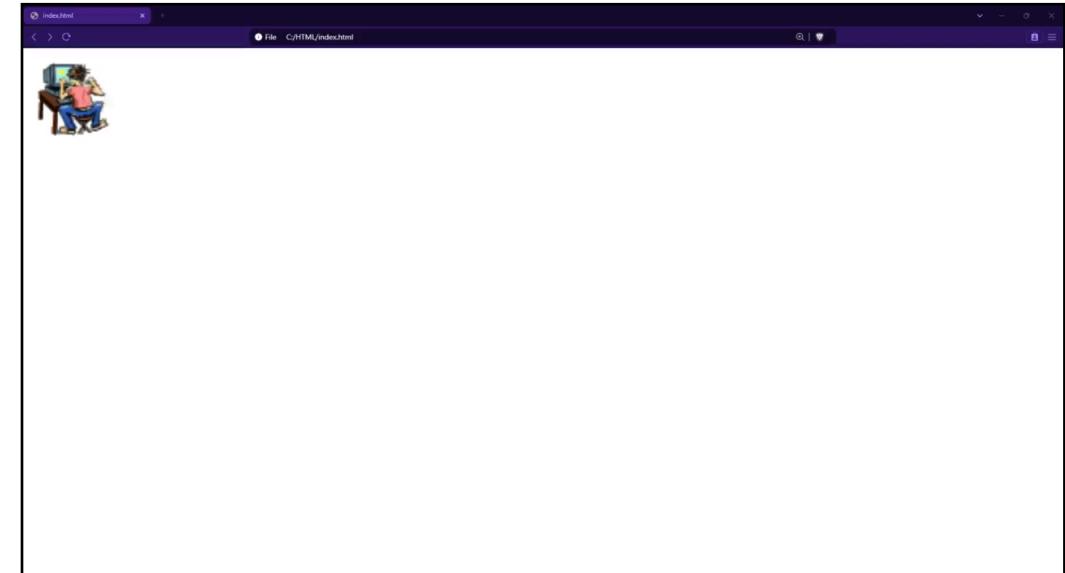
Images HTML

Images animées:

Le HTML permet de créer des GIF animés:

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5   </body>
6 </html>
```



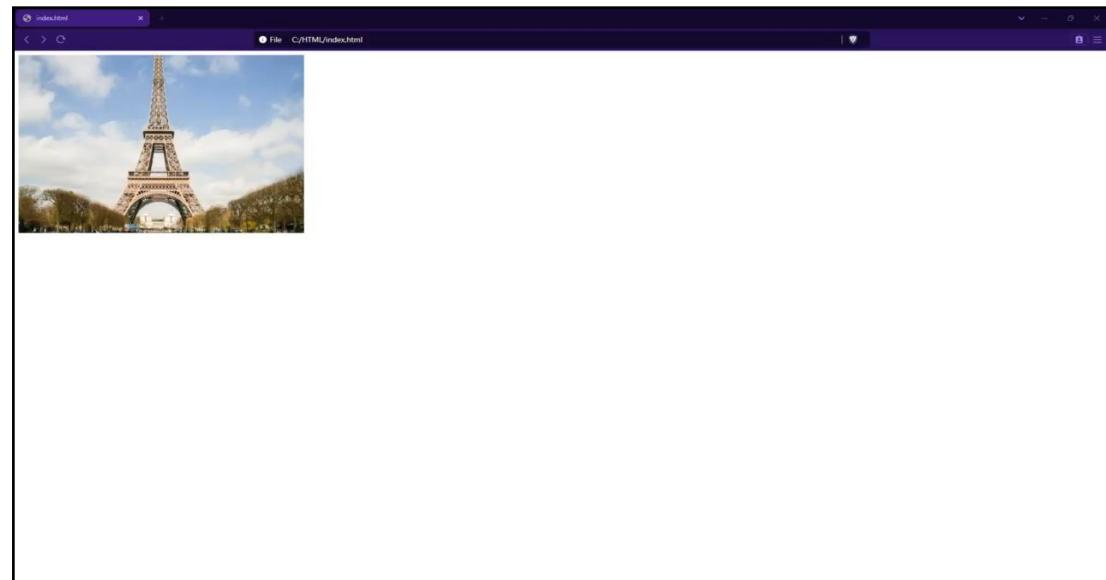
Images HTML

Image comme lien:

Pour utiliser une image comme lien, placez la balise `` à l'intérieur de la balise `<a>`:

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <a href="https://www.toureiffel.paris/fr">
5       
6     </a>
7   </body>
8 </html>
```



Images HTML

Formats d'image courants:

Voici les types de fichiers image les plus courants, qui sont pris en charge dans tous les navigateurs (Chrome, Edge, Firefox, Safari, Opera):

Abréviation	Format de fichier	Extension de fichier
APNG	Animated Portable Network Graphics(Graphiques réseau portables animés)	.apng
GIF	Graphics Interchange Format(Format d'échange de graphiques)	.gif
ICO	Microsoft Icon(Icône Microsoft)	.ico, .cur
JPEG	Joint Photographic Expert Group image(Image du Groupe conjoint d'experts photographiques)	.jpg, .jpeg, .jfif, .pjpeg, .pjp

Images HTML

PNG

Portable Network
Graphics(Carte
graphique réseau
portable)

.png

SVG

Scalable Vector
Graphics(Graphiques
vectoriels évolutifs)

.svg

Remarque: le chargement d'images volumineuses prend du temps et peut ralentir votre page Web. Utilisez les images avec précaution.

Icône de favori HTML

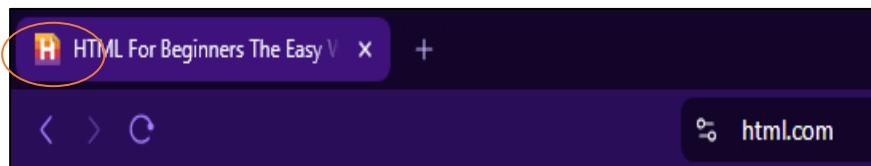
Un favicon est une petite image affichée à côté du titre de la page dans l'onglet du navigateur.

Comment ajouter un favicon en HTML:

Vous pouvez utiliser n'importe quelle image de votre choix comme favicon. Vous pouvez également créer votre propre favicon

Astuce: un favicon est une petite image, il doit donc s'agir d'une image simple avec un contraste élevé.

Une image de favicon s'affiche à gauche du titre de la page dans l'onglet du navigateur, comme ceci:



Pour ajouter un favicon à votre site Web, enregistrez votre image de favicon dans le répertoire racine de votre serveur Web ou créez un dossier dans le répertoire racine appelé `images` et enregistrez votre image de favicon dans ce dossier. Le nom courant d'une image de favicon est "favicon.ico".

Icône de favori HTML

Exemple:

The image shows a comparison between a code editor and a web browser. On the left, a code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ceci est le titre du document</title>
5     <link rel="icon" type="image/x-icon" href="images/favicon.ico" />
6   </head>
7   <body>
8     <h1>Ceci est un titre</h1>
9     <p>Ceci est un paragraphe.</p>
10    </body>
11 </html>
```

On the right, a browser window shows the rendered output. The title bar reads "Ceci est le titre du document". The main content area displays the text "Ceci est un titre" and "Ceci est un paragraphe.". A yellow circle highlights the "5" icon in the title bar.

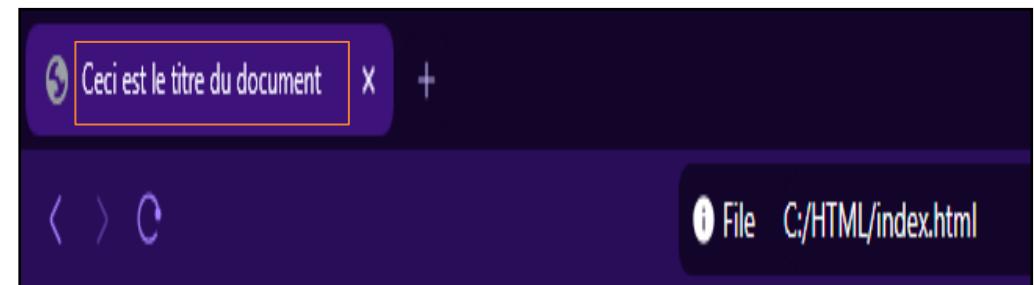
Titre de la page HTML

Chaque page Web doit avoir un titre pour décrire le sens de la page.

L'élément `<title>` ajoute un titre à votre page:

Exemple:

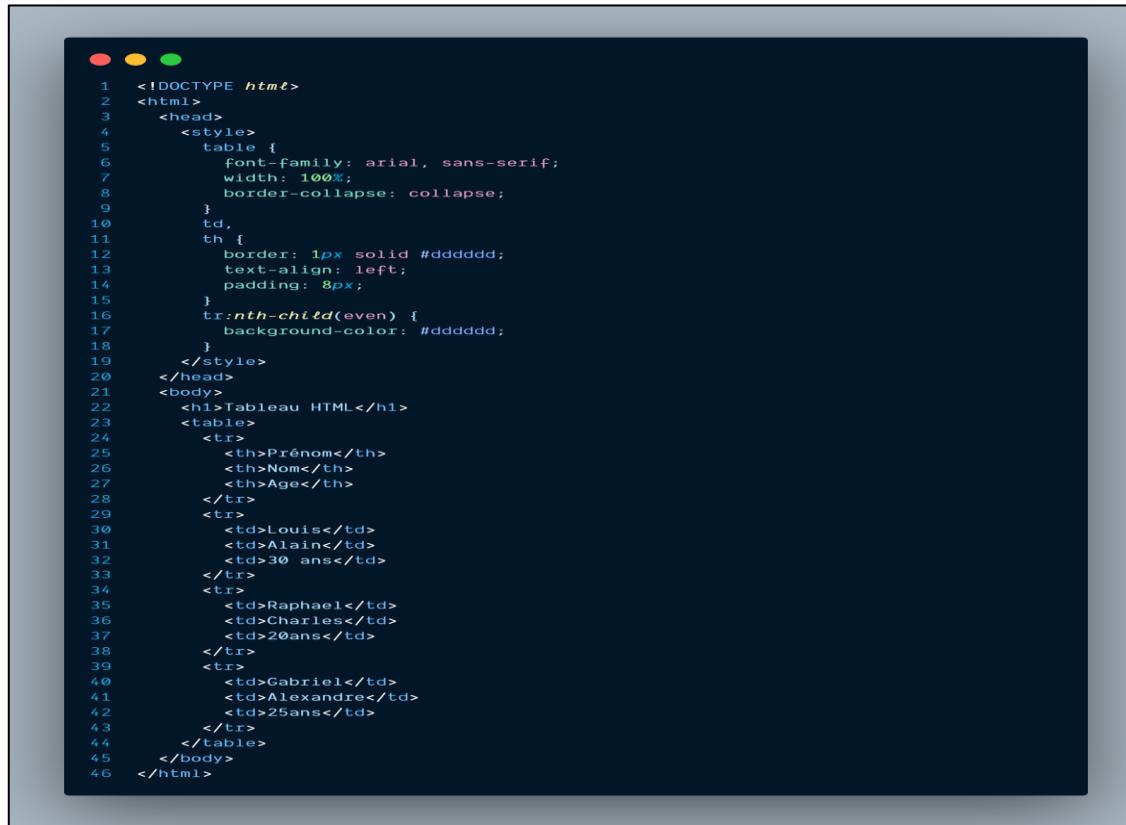
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ceci est le titre du document</title>
5   </head>
6   <body>
7     <h1>Ceci est un titre</h1>
8     <p>Ceci est un paragraphe.</p>
9   </body>
10 </html>
```



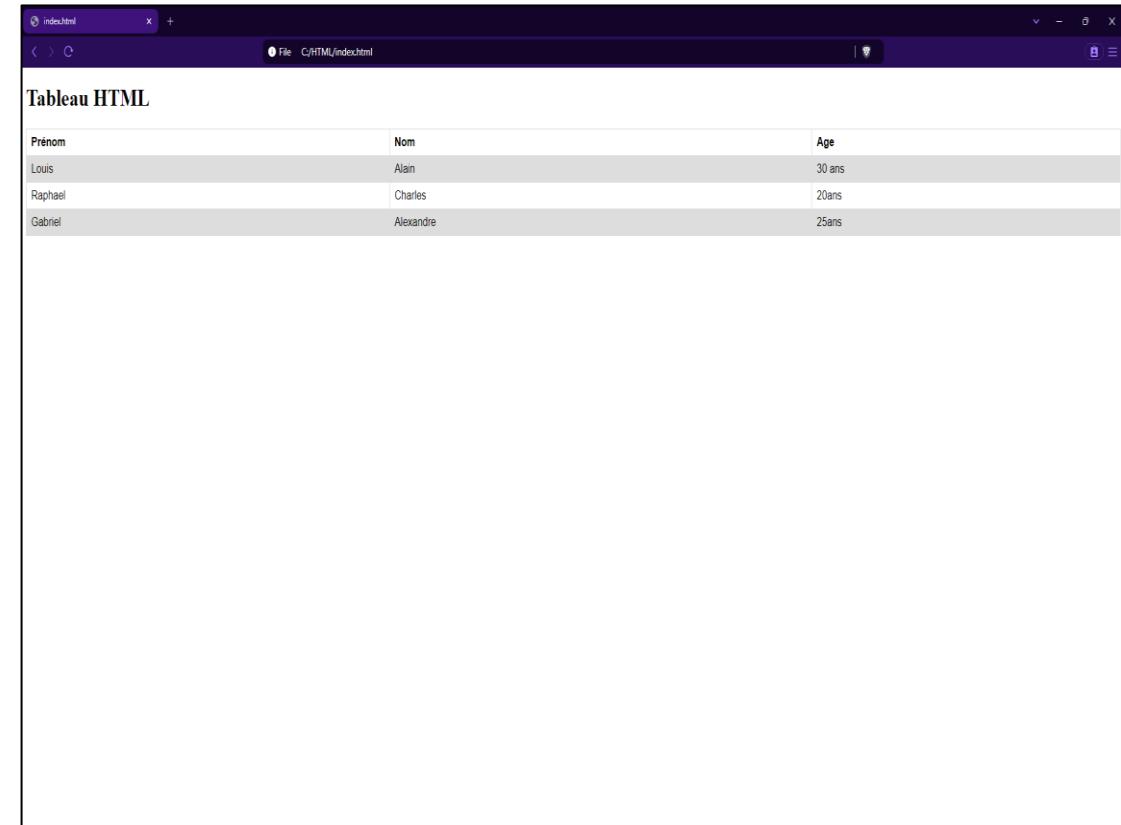
Tableaux HTML

Les tableaux HTML permettent aux développeurs Web d'organiser les données en lignes et en colonnes.

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       table {
6         font-family: arial, sans-serif;
7         width: 100%;
8         border-collapse: collapse;
9       }
10      td,
11      th {
12        border: 1px solid #dddddd;
13        text-align: left;
14        padding: 8px;
15      }
16      tr:nth-child(even) {
17        background-color: #eeeeee;
18      }
19    </style>
20  </head>
21  <body>
22    <h1>Tableau HTML</h1>
23    <table>
24      <tr>
25        <th>Prénom</th>
26        <th>Nom</th>
27        <th>Age</th>
28      </tr>
29      <tr>
30        <td>Louis</td>
31        <td>Alain</td>
32        <td>30 ans</td>
33      </tr>
34      <tr>
35        <td>Raphael</td>
36        <td>Charles</td>
37        <td>20ans</td>
38      </tr>
39      <tr>
40        <td>Gabriel</td>
41        <td>Alexandre</td>
42        <td>25ans</td>
43      </tr>
44    </table>
45  </body>
46 </html>
```



Prénom	Nom	Age
Louis	Alain	30 ans
Raphael	Charles	20ans
Gabriel	Alexandre	25ans

Tableaux HTML

Définir un tableau HTML:

Un tableau en HTML est constitué de cellules de tableau à l'intérieur de lignes et de colonnes.

Exemple:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <style>
5        table, td, th {
6          border: 1px solid #dddddd;
7        }
8      </style>
9    </head>
10   <body>
11     <h1>Tableau HTML</h1>
12     <table style="width: 100%;">
13       <tr>
14         <th>Prénom</th>
15         <th>Nom</th>
16         <th>Age</th>
17       </tr>
18       <tr>
19         <td>Louis</td>
20         <td>Alain</td>
21         <td>30 ans</td>
22       </tr>
23       <tr>
24         <td>Gabriel</td>
25         <td>Alexandre</td>
26         <td>25ans</td>
27       </tr>
28     </table>
29   </body>
30 </html>
```

The screenshot shows a Microsoft Edge browser window with the title "index.html". The page content is titled "Tableau HTML" and displays a table with the following data:

Prénom	Nom	Age
Louis	Alain	30 ans
Gabriel	Alexandre	25ans

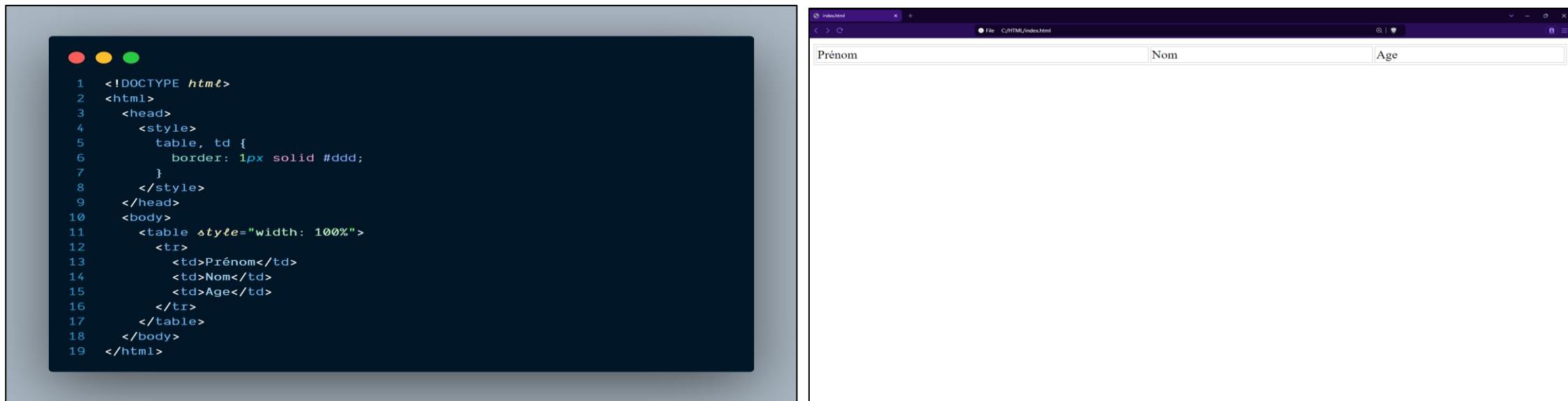
Tableaux HTML

Cellules du tableau:

Chaque cellule du tableau est définie par une balise `<td>` et une balise `</td>`.
`td` signifie table data(données de table).

Tout ce qui se trouve entre `<td>` et `</td>` constitue le contenu de la cellule du tableau.

Exemple:



The image shows a code editor on the left and a web browser on the right. The code editor displays the following HTML and CSS code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       table, td {
6         border: 1px solid #ddd;
7       }
8     </style>
9   </head>
10  <body>
11    <table style="width: 100%">
12      <tr>
13        <td>Prénom</td>
14        <td>Nom</td>
15        <td>Age</td>
16      </tr>
17    </table>
18  </body>
19 </html>
```

The browser window shows a simple table with three columns. The first column is labeled "Prénom", the second "Nom", and the third "Age".

Remarque: une cellule de tableau peut contenir toutes sortes d'éléments HTML : texte, images, listes, liens, autres tableaux, etc.

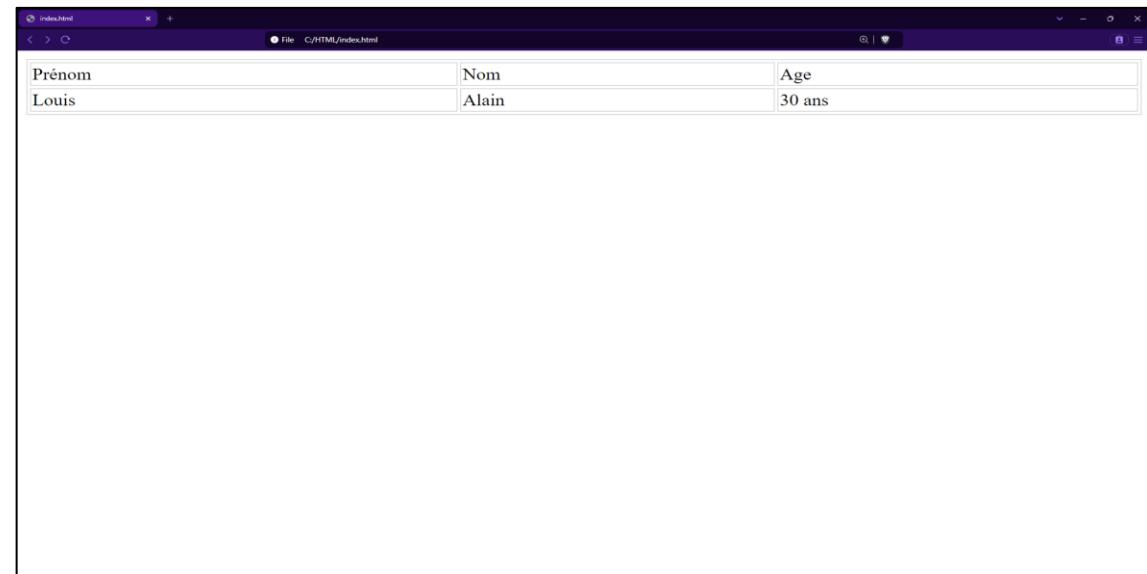
Tableaux HTML

Lignes du tableau:

Chaque ligne du tableau commence par une balise `<tr>` et se termine par une balise `</tr>`.
`tr` signifie table row(ligne de tableau).

Exemple:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <style>
5        table, td {
6          border: 1px solid #ddd;
7        }
8      </style>
9    </head>
10   <body>
11     <table style="width: 100%">
12       <tr>
13         <td>Prénom</td>
14         <td>Nom</td>
15         <td>Age</td>
16       </tr>
17       <tr>
18         <td>Louis</td>
19         <td>Alain</td>
20         <td>30 ans</td>
21       </tr>
22     </table>
23   </body>
24 </html>
```



Prénom	Nom	Age
Louis	Alain	30 ans

Vous pouvez avoir autant de lignes que vous le souhaitez dans un tableau ; assurez-vous simplement que le nombre de cellules est le même dans chaque ligne.

Remarque: il arrive parfois qu'une ligne ait moins ou plus de cellules qu'une autre.

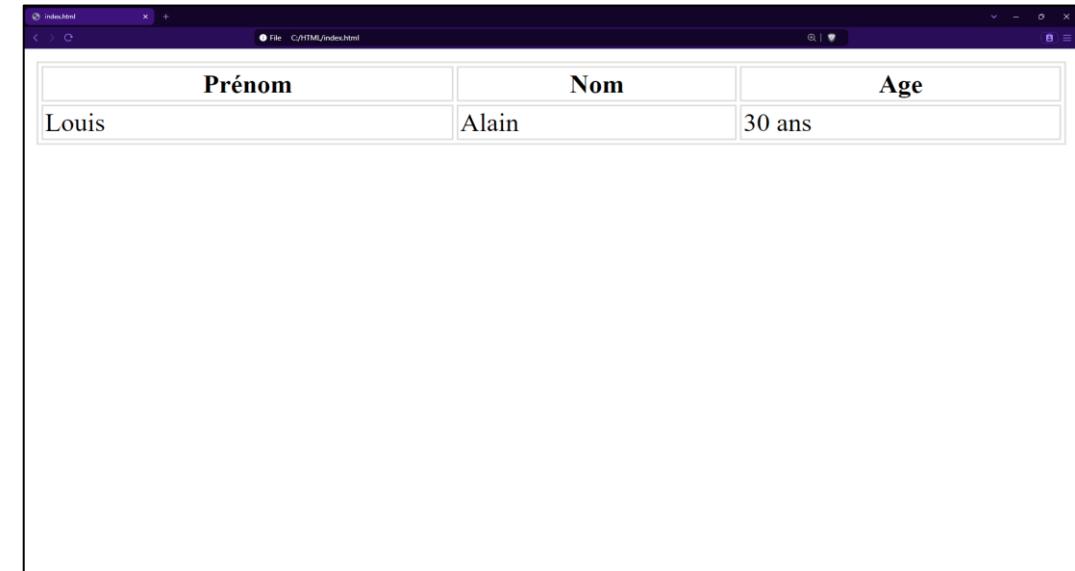
Tableaux HTML

En-têtes de tableau:

Parfois, vous souhaitez que vos cellules soient des cellules d'en-tête de tableau. Dans ce cas, utilisez la balise `<th>` au lieu de la balise `<td>` :
`th` signifie table header(en-tête de tableau).

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       table, td, th {
6         border: 1px solid #dddddd;
7       }
8     </style>
9   </head>
10  <body>
11    <table style="width: 100%;">
12      <tr>
13        <th>Prénom</th>
14        <th>Nom</th>
15        <th>Age</th>
16      </tr>
17      <tr>
18        <td>Louis</td>
19        <td>Alain</td>
20        <td>30 ans</td>
21      </tr>
22    </table>
23  </body>
24 </html>
```



Prénom	Nom	Age
Louis	Alain	30 ans

Par défaut, le texte dans les éléments `<th>` est en gras et centré

Tableaux HTML

Légende du tableau:

Vous pouvez ajouter une légende qui sert de titre à l'ensemble du tableau.

Pour ajouter une légende à un tableau, utilisez la balise `<caption>`:

Remarque: la balise `<caption>` doit être insérée immédiatement après la balise `<table>`.

The image shows two side-by-side screenshots. On the left is a code editor window displaying the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       table, th, td {
6         border: 1px solid #ddd;
7       }
8     </style>
9   </head>
10  <body>
11    <table style="width: 100%">
12      <caption>Informations sur les utilisateurs</caption>
13      <tr>
14        <th>Prénom</th>
15        <th>Nom</th>
16        <th>Age</th>
17      </tr>
18      <tr>
19        <td>Louis</td>
20        <td>Alain</td>
21        <td>30 ans</td>
22      </tr>
23    </table>
24  </body>
25 </html>
```

On the right is a screenshot of a web browser window titled "index.html" showing the rendered table. The table has a caption "Informations sur les utilisateurs". It contains three columns: "Prénom", "Nom", and "Age". The data rows show "Louis", "Alain", and "30 ans" respectively.

Par défaut, le texte dans les éléments `<th>` est en gras et centré

Tableaux HTML

Padding et Spacing:

Les tableaux HTML peuvent ajuster le remplissage à l'intérieur des cellules, ainsi que l'espace entre les cellules.

Avec Padding

Bonjour	Bonjour	Bonjour
Bonjour	Bonjour	Bonjour
Bonjour	Bonjour	Bonjour

Avec Spacing

Bonjour	Bonjour	Bonjour
Bonjour	Bonjour	Bonjour
Bonjour	Bonjour	Bonjour

Padding:

Le remplissage des cellules correspond à l'espace entre les bords des cellules et le contenu de la cellule.

Par défaut, le remplissage est défini sur 0.

Pour ajouter un remplissage aux cellules du tableau, utilisez la propriété CSS **padding** sur les éléments **th** et **td**.

Spacing:

L'espacement des cellules correspond à l'espace entre chaque cellule.

Par défaut, l'espace est défini sur 2 pixels.

Pour modifier l'espace entre les cellules d'un tableau, utilisez la propriété CSS **border-spacing** sur l'élément **tableau**.

Tableaux HTML

Colspan & Rowspan:

Les tableaux HTML peuvent contenir des cellules qui s'étendent sur plusieurs lignes et/ou colonnes.

Exemple	

Exemple		

Exemple		
Exemple		

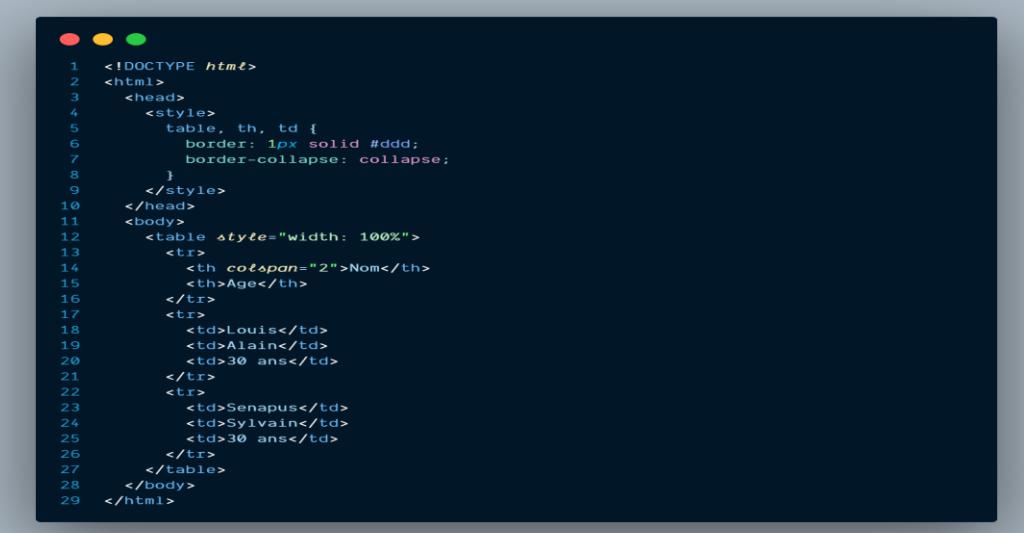
Colspan:

Pour étendre une cellule sur plusieurs colonnes, utilisez l'attribut **colspan**.

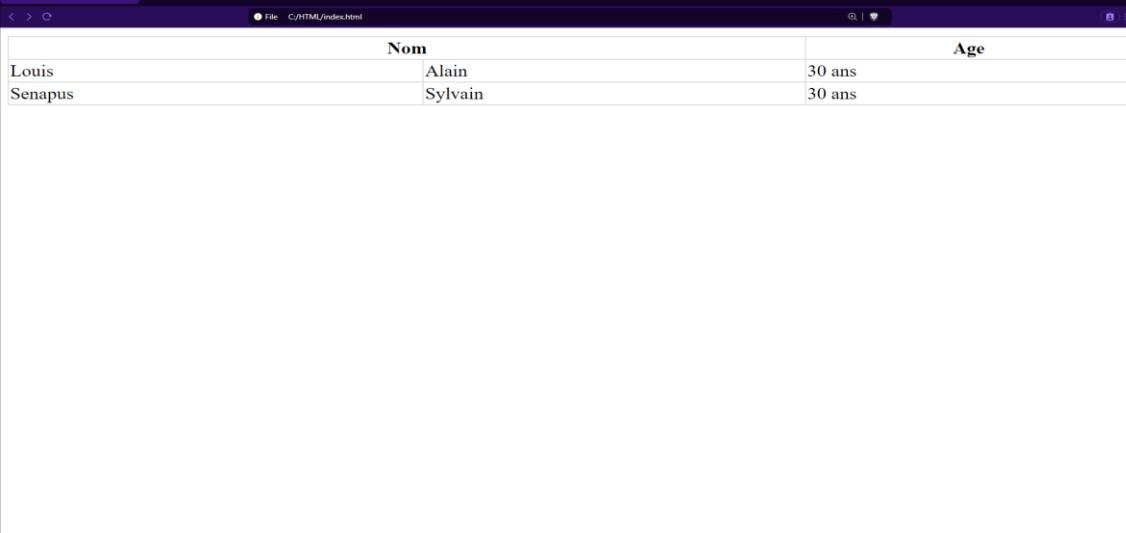
Remarque: la valeur de l'attribut **colspan** représente le nombre de colonnes à couvrir.

Exemple:

Tableaux HTML



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       table, th, td {
6         border: 1px solid #ddd;
7         border-collapse: collapse;
8       }
9     </style>
10   </head>
11   <body>
12     <table style="width: 100%">
13       <tr>
14         <th colspan="2">Nom</th>
15         <th>Age</th>
16       </tr>
17       <tr>
18         <td>Louis</td>
19         <td>Alain</td>
20         <td>30 ans</td>
21       </tr>
22       <tr>
23         <td>Senapus</td>
24         <td>Sylvain</td>
25         <td>30 ans</td>
26       </tr>
27     </table>
28   </body>
29 </html>
```



Nom	Age
Louis	30 ans
Senapus	30 ans

Rowspan:

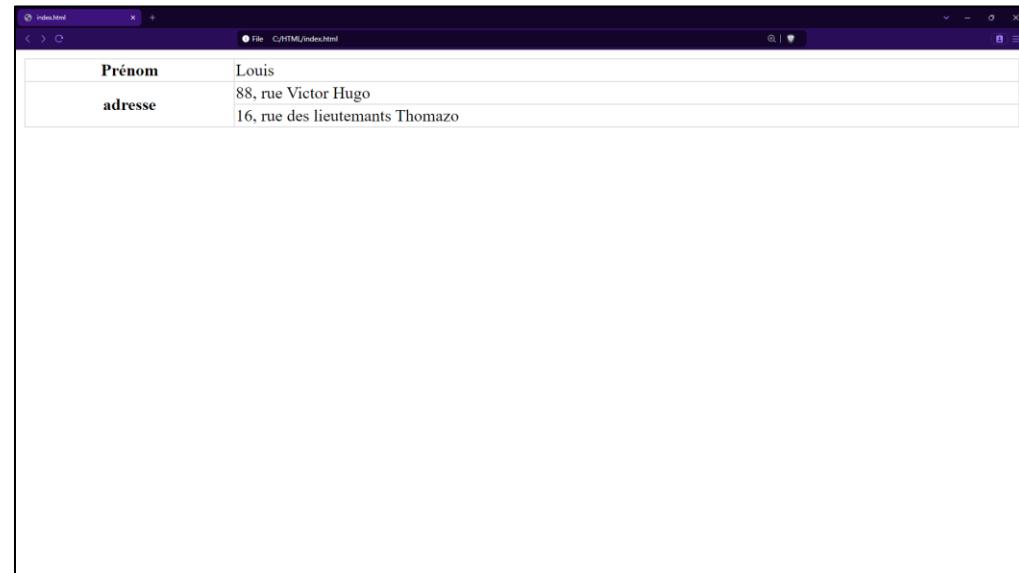
Pour créer une étendue de cellule sur plusieurs lignes, utilisez l'attribut **rowspan**.

Remarque: la valeur de l'attribut **rowspan** représente le nombre de lignes à couvrir.

Exemple:

Tableaux HTML

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <style>
5        table, th, td {
6          border: 1px solid #ddd;
7          border-collapse: collapse;
8        }
9      </style>
10   </head>
11   <body>
12     <table style="width: 100%">
13       <tr>
14         <th>Prénom</th>
15         <td>Louis</td>
16       </tr>
17       <tr>
18         <th rowspan="2">adresse</th>
19         <td>88, rue Victor Hugo</td>
20       </tr>
21       <tr>
22         <td>16, rue des lieutemants Thomazo</td>
23       </tr>
24     </table>
25   </body>
26 </html>
```



The screenshot shows a Microsoft Edge browser window displaying the rendered HTML table. The table has three rows. The first row contains one column with the header "Prénom" and the value "Louis". The second row contains one column with the header "adresse" and the value "88, rue Victor Hugo". The third row contains one column with the value "16, rue des lieutemants Thomazo". The browser interface includes a title bar, a toolbar with icons, and a status bar at the bottom.

Prénom	Louis
adresse	88, rue Victor Hugo
16, rue des lieutemants Thomazo	

Tableaux HTML

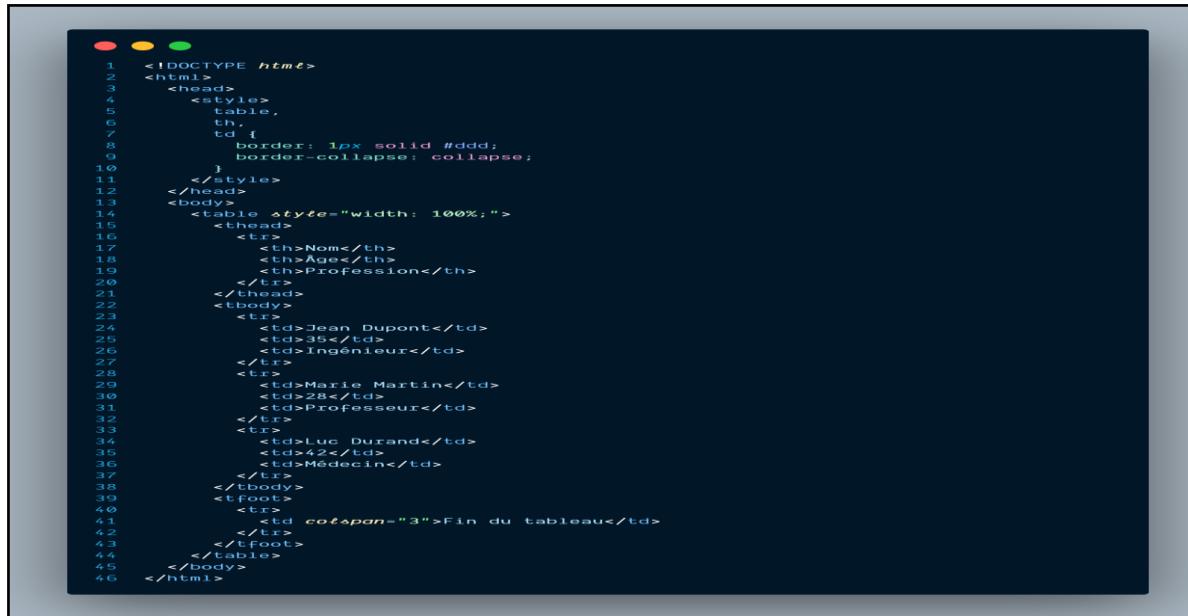
<thead>, <tbody>, <tfoot>:

<thead>: Contient l'en-tête du tableau.

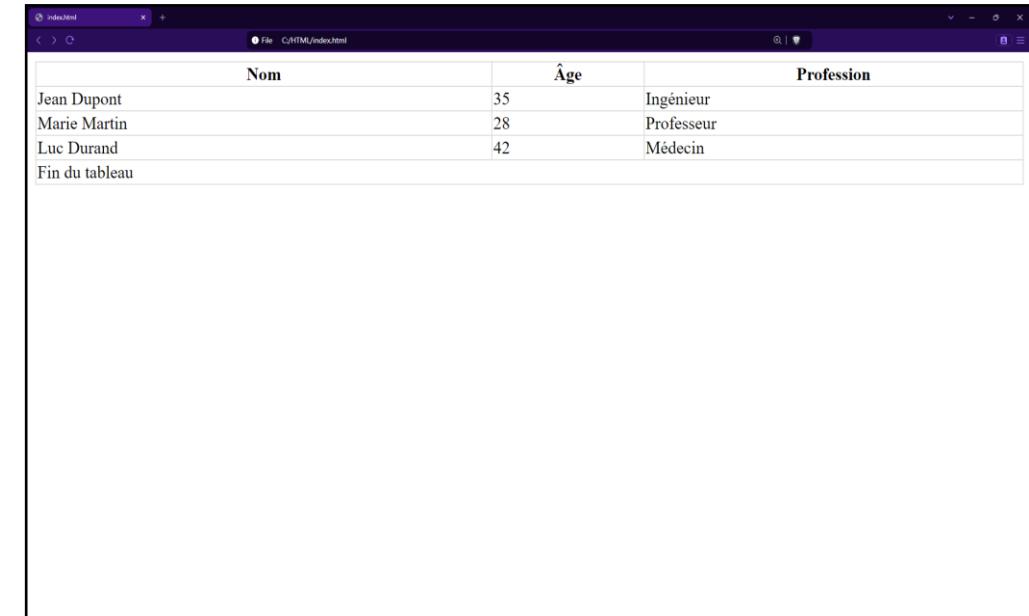
<tbody>: Représente le corps du tableau avec les données.

<tfoot>: C'est le pied de page du tableau.

Exemple:



```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <style>
5        table,
6        th,
7        td {
8          border: 1px solid #ddd;
9          border-collapse: collapse;
10         }
11     </style>
12   </head>
13   <body>
14     <table style="width: 100%;">
15       <thead>
16         <tr>
17           <th>Nom</th>
18           <th>Âge</th>
19           <th>Profession</th>
20         </tr>
21       </thead>
22       <tbody>
23         <tr>
24           <td>Jean Dupont</td>
25           <td>35</td>
26           <td>Ingénieur</td>
27         </tr>
28         <tr>
29           <td>Marie Martin</td>
30           <td>28</td>
31           <td>Professeur</td>
32         </tr>
33         <tr>
34           <td>Luc Durand</td>
35           <td>42</td>
36           <td>Médecin</td>
37         </tr>
38       </tbody>
39       <tfoot>
40         <tr>
41           <td colspan="3">Fin du tableau</td>
42         </tr>
43       </tfoot>
44     </table>
45   </body>
46 </html>
```



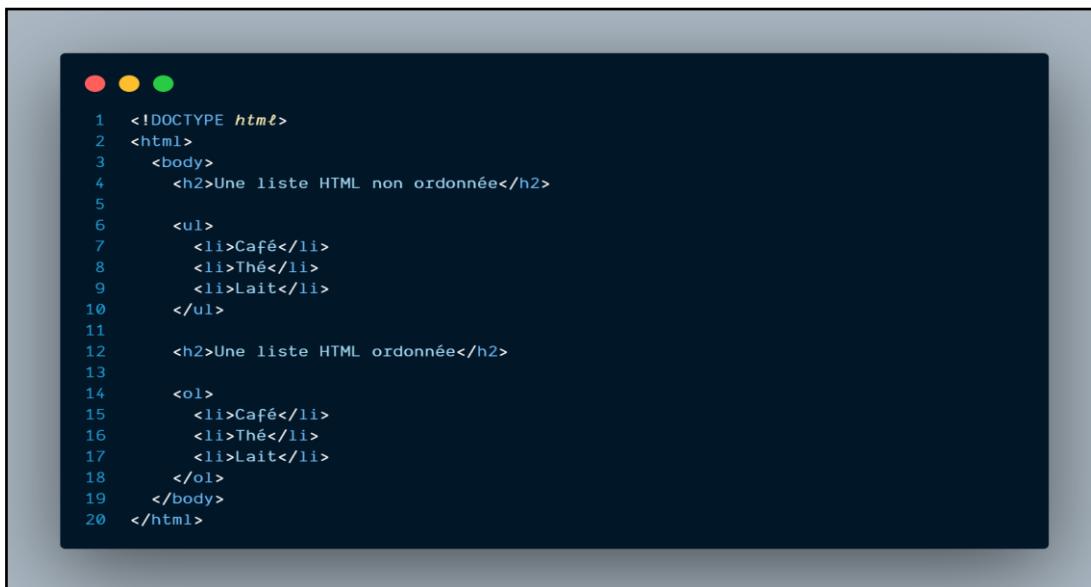
Nom	Âge	Profession
Jean Dupont	35	Ingénieur
Marie Martin	28	Professeur
Luc Durand	42	Médecin
Fin du tableau		

Remarque: même si vous placez les balises dans un ordre non conventionnel, comme mettre <tfoot> avant <thead> et <tbody>, le tableau s'affichera correctement visuellement dans le navigateur.

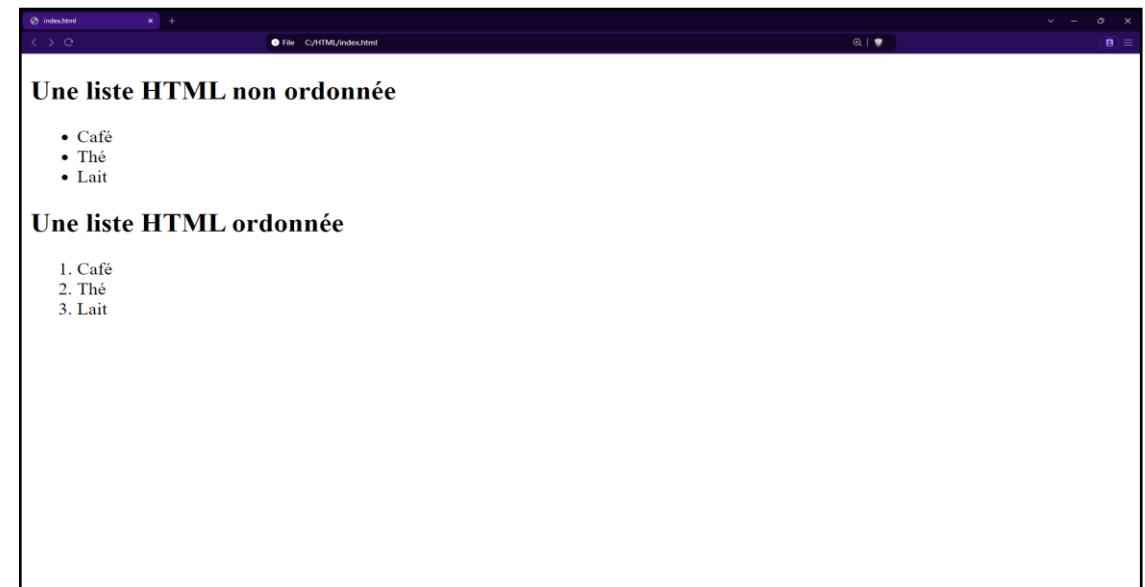
Listes HTML

Les listes HTML permettent aux développeurs Web de regrouper un ensemble d'éléments connexes dans des listes.

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Une liste HTML non ordonnée</h2>
5
6     <ul>
7       <li>CAFÉ</li>
8       <li>Thé</li>
9       <li>Lait</li>
10    </ul>
11
12    <h2>Une liste HTML ordonnée</h2>
13
14    <ol>
15      <li>CAFÉ</li>
16      <li>Thé</li>
17      <li>Lait</li>
18    </ol>
19  </body>
20 </html>
```



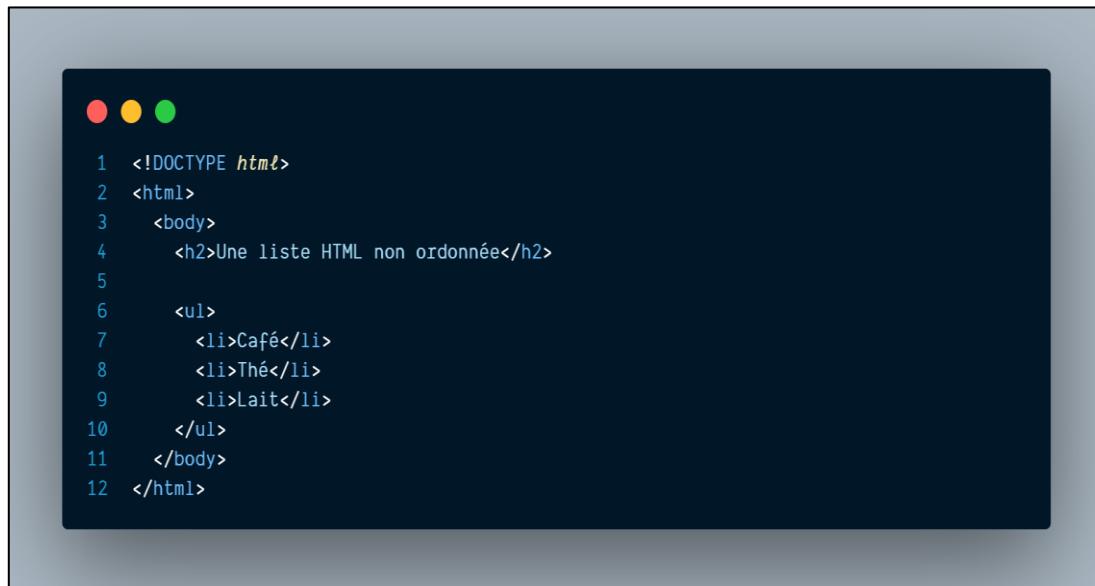
Listes HTML

Liste HTML non ordonnée:

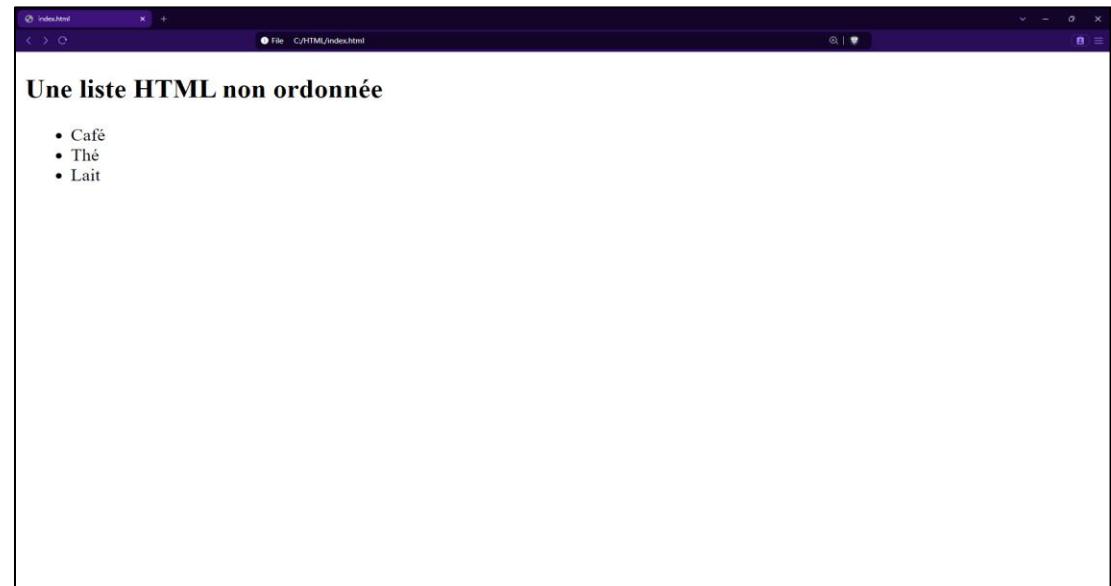
Une liste non ordonnée commence par la balise ``. Chaque élément de la liste commence par la balise ``.

Les éléments de la liste seront marqués par des puces (petits cercles noirs) par défaut.

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Une liste HTML non ordonnée</h2>
5
6     <ul>
7       <li>CAFÉ</li>
8       <li>Thé</li>
9       <li>Lait</li>
10    </ul>
11  </body>
12 </html>
```



Listes HTML

Marqueur d'élément de liste:

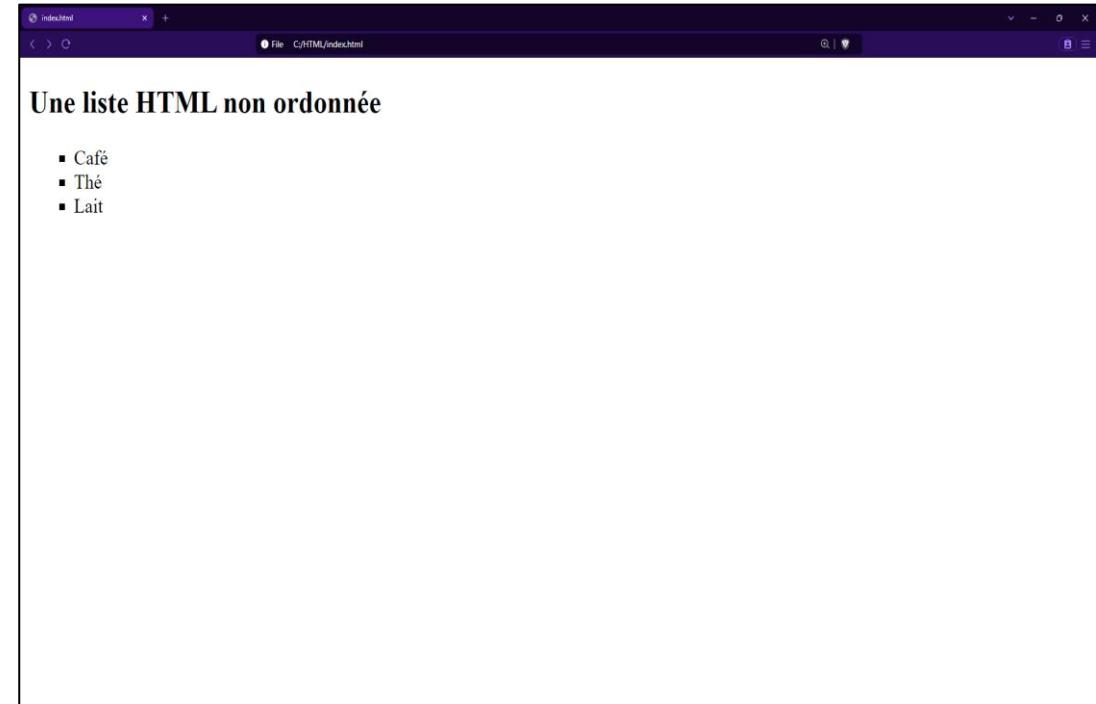
La propriété CSS `list-style-type` permet de définir le style du marqueur d'élément de liste. Elle peut avoir l'une des valeurs suivantes:

Valeur	Description
disc	Définit le marqueur d'élément de liste sur une puce (par défaut)
circle	Définit le marqueur d'élément de liste sur un cercle
square	Définit le marqueur d'élément de liste sur un carré
none	Les éléments de la liste ne seront pas marqués

Exemple:

Listes HTML

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Une liste HTML non ordonnée</h2>
5
6     <ul style="list-style-type: square;">
7       <li>CAFÉ</li>
8       <li>Thé</li>
9       <li>Lait</li>
10    </ul>
11  </body>
12 </html>
```



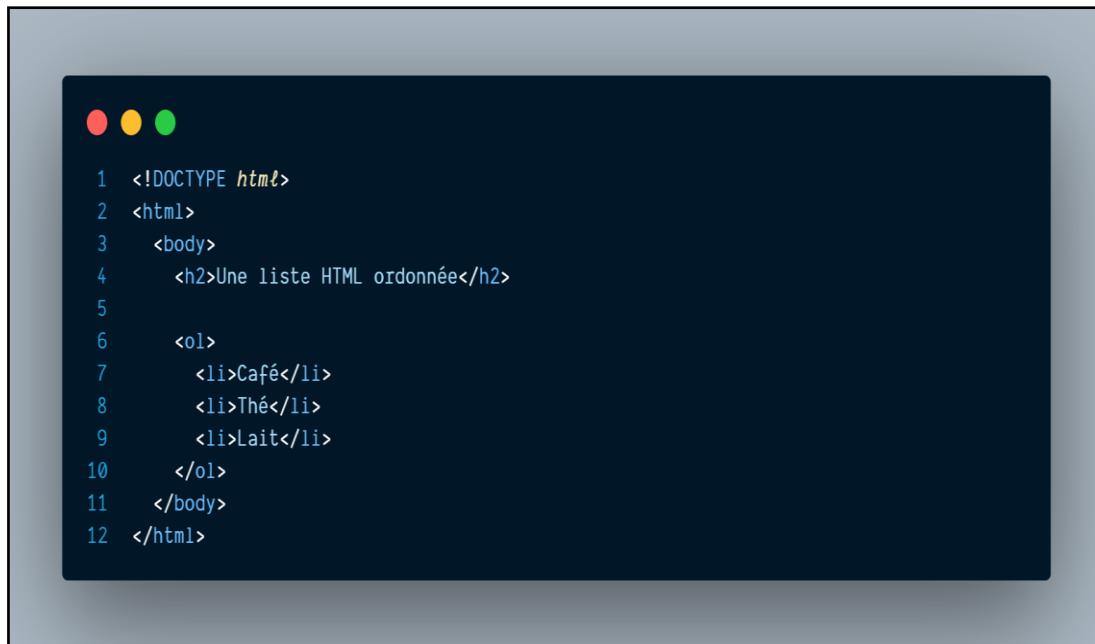
Listes HTML

Liste HTML ordonnée:

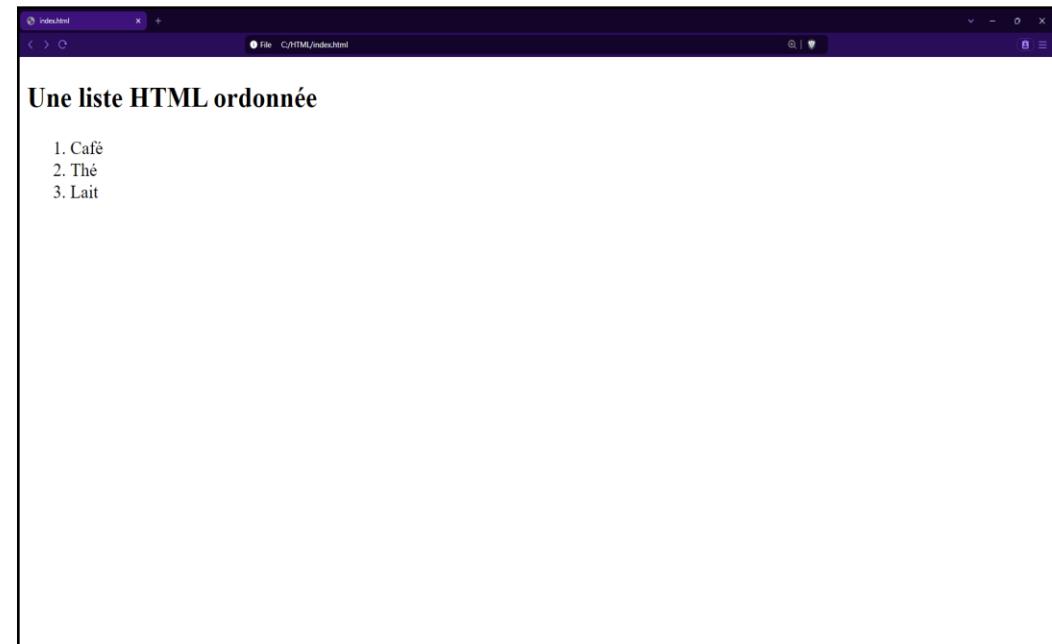
Une liste ordonnée commence par la balise ``. Chaque élément de la liste commence par la balise ``.

Les éléments de la liste seront marqués par défaut avec des numéros.

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Une liste HTML ordonnée</h2>
5
6     <ol>
7       <li>CAFÉ</li>
8       <li>Thé</li>
9       <li>Lait</li>
10    </ol>
11  </body>
12 </html>
```



Listes HTML

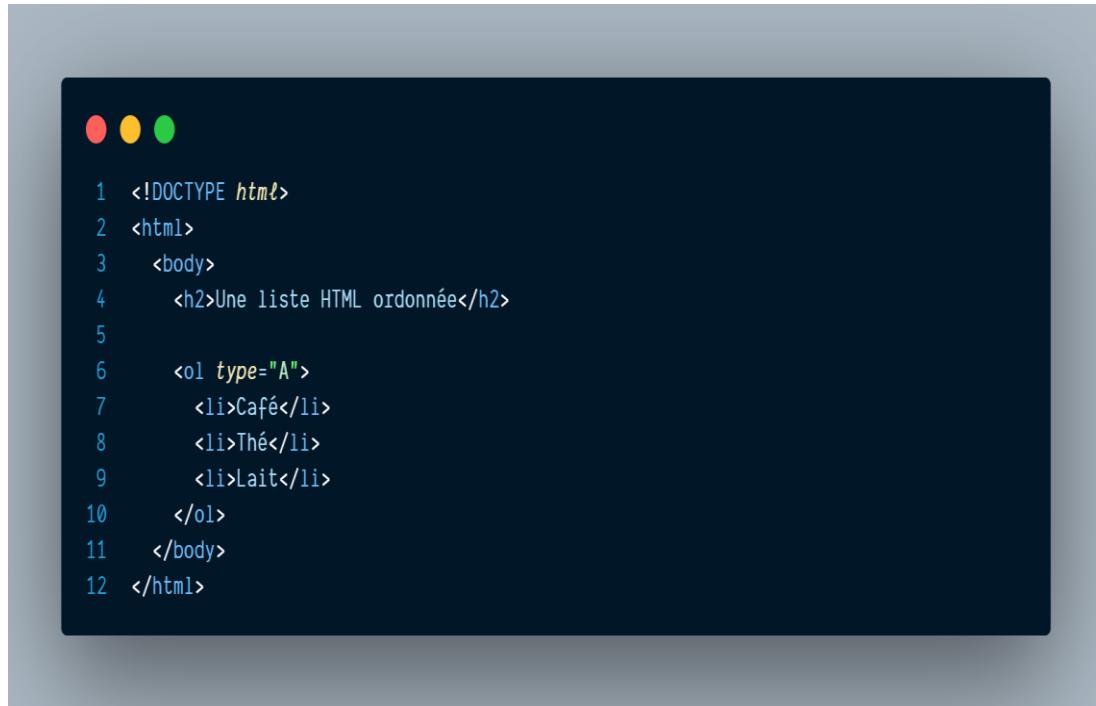
L'attribut type:

L'attribut **type** de la balise `` définit le type du marqueur d'élément de liste:

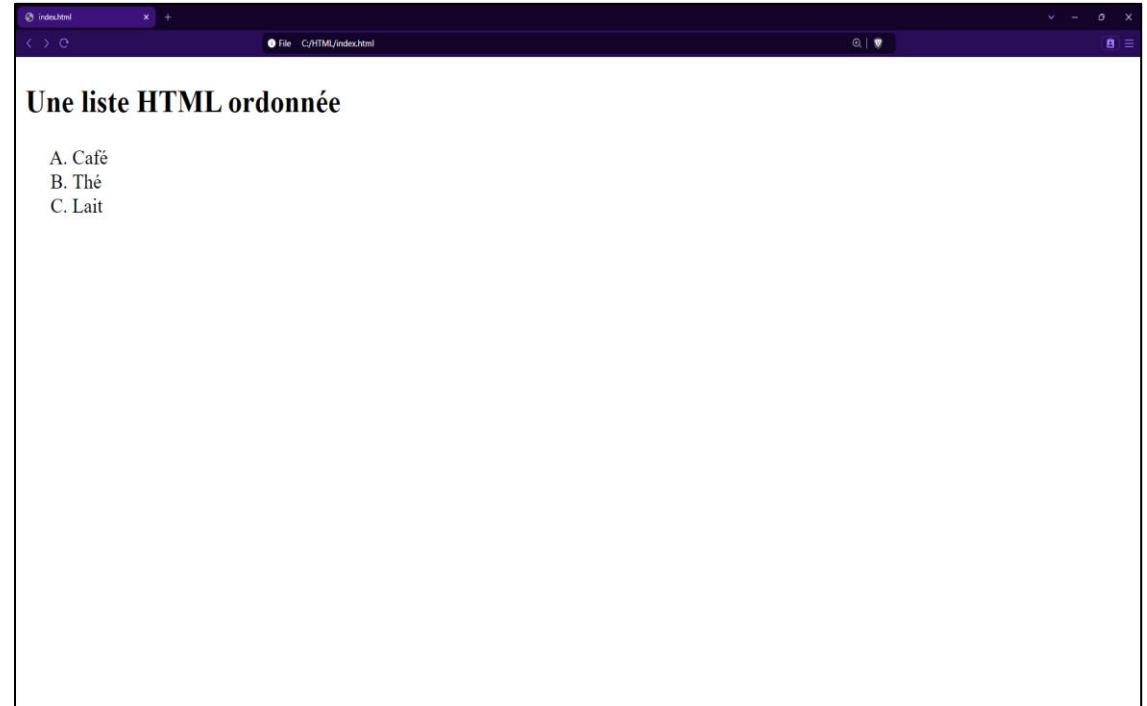
Type	Description
type="1"	Les éléments de la liste seront numérotés avec des numéros (par défaut)
type="A"	Les éléments de la liste seront numérotés avec des lettres majuscules
type="a"	Les éléments de la liste seront numérotés avec des lettres minuscules
type="I"	Les éléments de la liste seront numérotés avec des chiffres romains majuscules
type="i"	Les éléments de la liste seront numérotés avec des chiffres romains minuscules

Exemple:

Listes HTML



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Une liste HTML ordonnée</h2>
5
6     <ol type="A">
7       <li>CAFÉ</li>
8       <li>Thé</li>
9       <li>Lait</li>
10    </ol>
11  </body>
12 </html>
```



Comptage de la liste de contrôle:

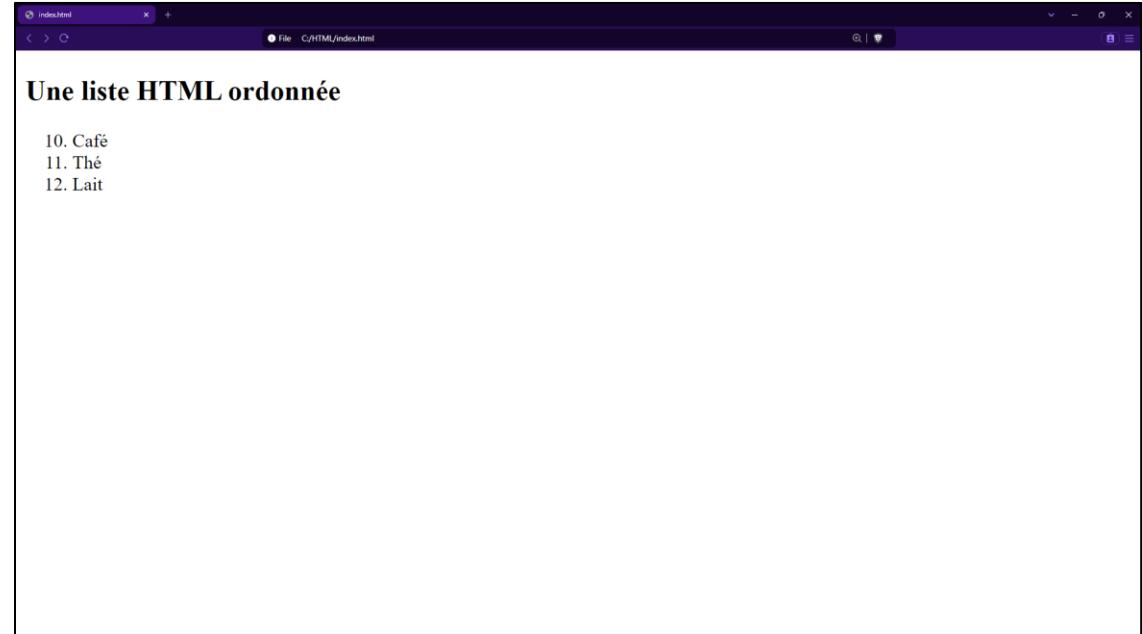
Par défaut, une liste ordonnée commence à compter à partir de 1. Si vous souhaitez commencer à compter à partir d'un nombre spécifié, vous pouvez utiliser l'attribut **start**:

Exemple:

Listes HTML



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Une liste HTML ordonnée</h2>
5
6     <ol start="10">
7       <li>CAFÉ</li>
8       <li>Thé</li>
9       <li>Lait</li>
10    </ol>
11  </body>
12 </html>
```



Les éléments block et inline dans HTML

Chaque élément HTML possède une valeur d'affichage par défaut, en fonction du type d'élément dont il s'agit.

Les deux valeurs d'affichage les plus courantes sont **block** et **inline**.

Éléments de niveau bloc:

Un élément de niveau bloc commence toujours sur une nouvelle ligne et les navigateurs ajoutent automatiquement un espace (une marge) avant et après l'élément. Un élément de niveau bloc occupe toujours toute la largeur disponible (s'étend vers la gauche et la droite aussi loin que possible).

Deux éléments de bloc couramment utilisés sont: `<p>` et `<div>`.

L'élément `<p>` définit un paragraphe dans un document HTML.

L'élément `<div>` définit une division ou une section dans un document HTML.

L'élément `<p>` est un élément de niveau bloc.

L'élément `<div>` est un élément de niveau bloc.

Exemple:

Les éléments block et inline dans HTML

The image shows a code editor on the left and a browser window on the right. The code editor contains the following HTML and CSS:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       p, div {
6         border: 1px solid black;
7       }
8     </style>
9   </head>
10  <body>
11    <p>L'élément p est un élément de niveau bloc.</p>
12    <div>L'élément div est un élément de niveau bloc.</div>
13  </body>
14 </html>
```

The browser window shows two separate paragraphs, each with a black border, demonstrating that both `p` and `div` elements are block-level elements.

Voici quelques éléments de niveau bloc en HTML:

`<address>`, `<article>`, `<aside>`, `<blockquote>`, `<dd>`, `<div>`, `<dl>`, `<dt>`,
`<fieldset>`, `<figcaption>`, `<figure>`, `<footer>`, `<form>`, `<h1>-<h6>`, `<header>`,
`<hr>`, ``, `<main>`, `<nav>`, ``, `<p>`, `<pre>`, `<section>`, `<table>`, ``,
`<video>`,...

Les éléments bloc et en ligne dans HTML

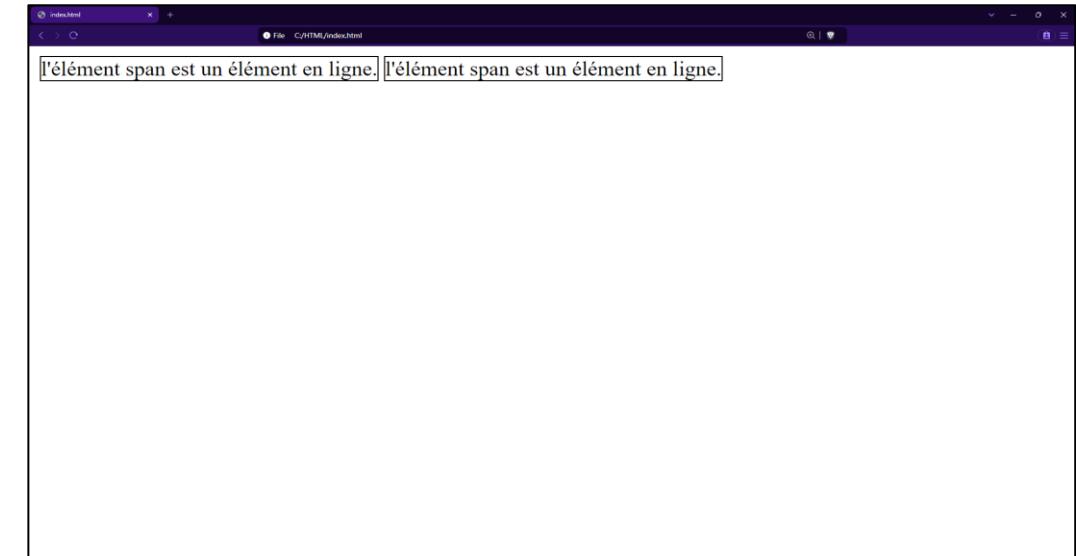
Éléments en ligne:

Un élément en ligne ne commence pas sur une nouvelle ligne.

Un élément en ligne occupe uniquement la largeur nécessaire.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       span {
6         border: 1px solid black;
7       }
8     </style>
9   </head>
10  <body>
11    <span>l'élément span est un élément en ligne.</span>
12    <span>l'élément span est un élément en ligne.</span>
13  </body>
14 </html>
```



Voici quelques éléments en ligne en HTML:

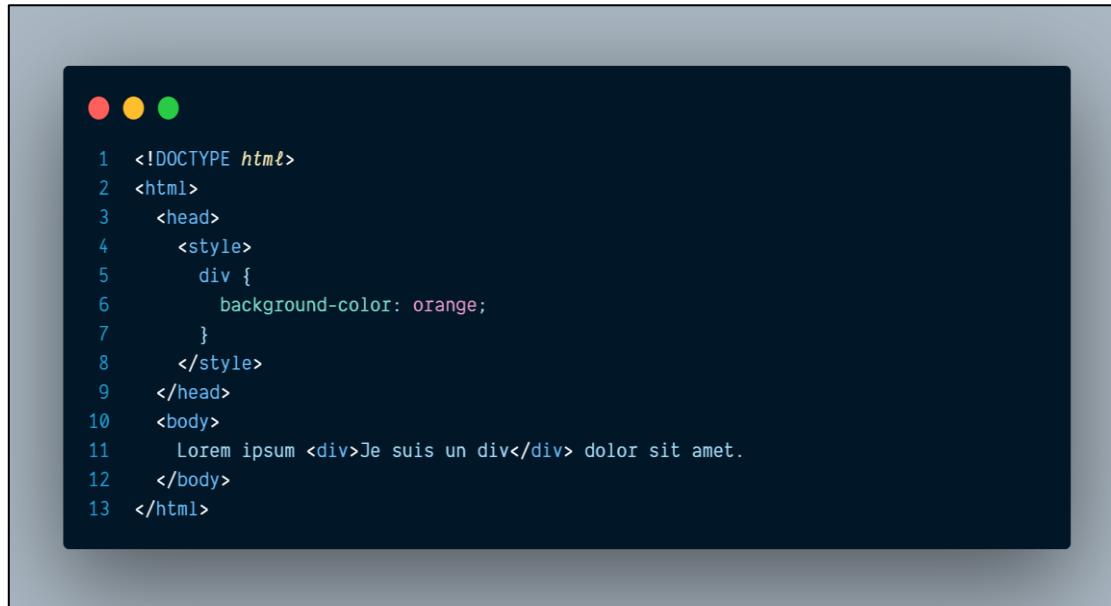
<a>, <abbr>, ,
, <bdo>, <button>, <cite>, , <i>, , <input>, <label>, <q>, <select>, <small>, , , <sub>, <sup>, <textarea>,...

Élément HTML Div

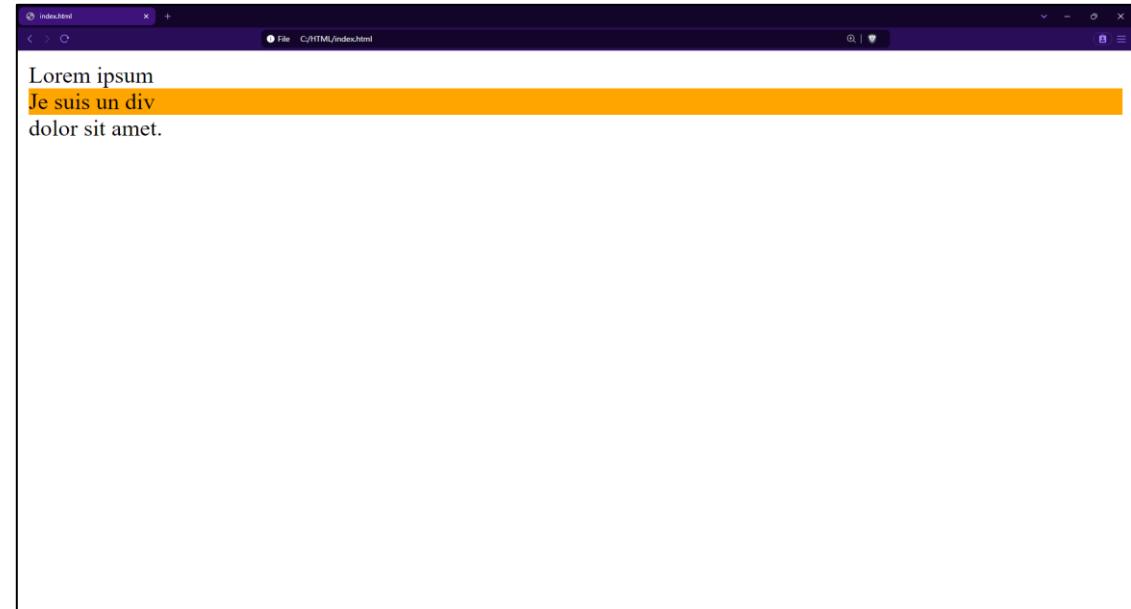
L'élément `<div>` est utilisé comme conteneur pour d'autres éléments HTML.

L'élément `<div>`:

L'élément `<div>` est par défaut un élément de bloc, ce qui signifie qu'il occupe toute la largeur disponible et qu'il est accompagné de sauts de ligne avant et après.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       div {
6         background-color: orange;
7       }
8     </style>
9   </head>
10  <body>
11    Lorem ipsum <div>Je suis un div</div> dolor sit amet.
12  </body>
13 </html>
```



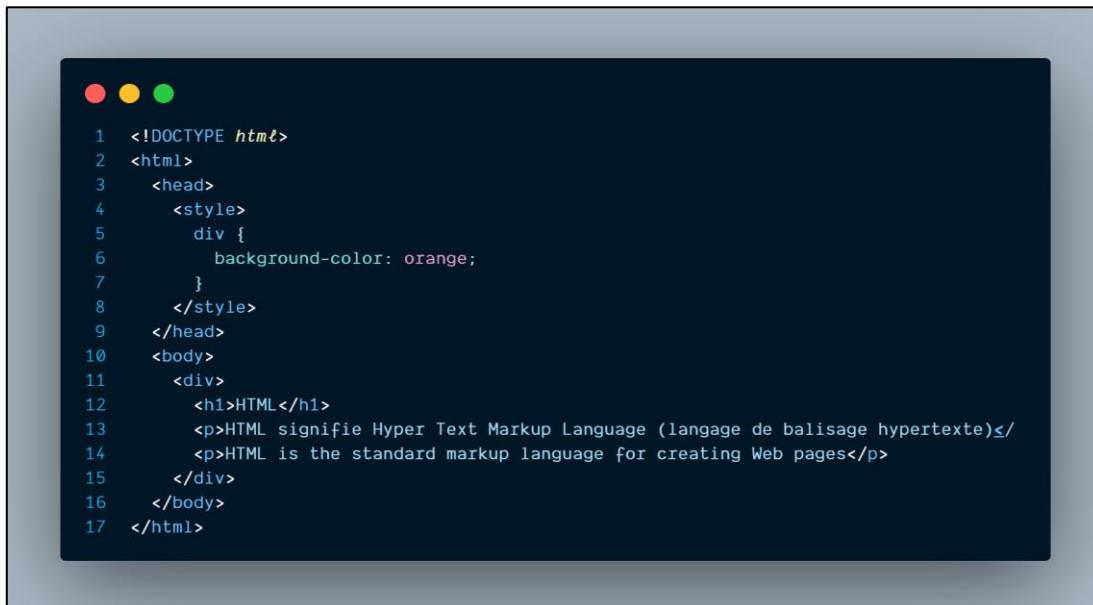
L'élément `<div>` n'a pas d'attributs obligatoires, mais le `style`, la `classe` et l'`id` sont courants.

Élément HTML Div

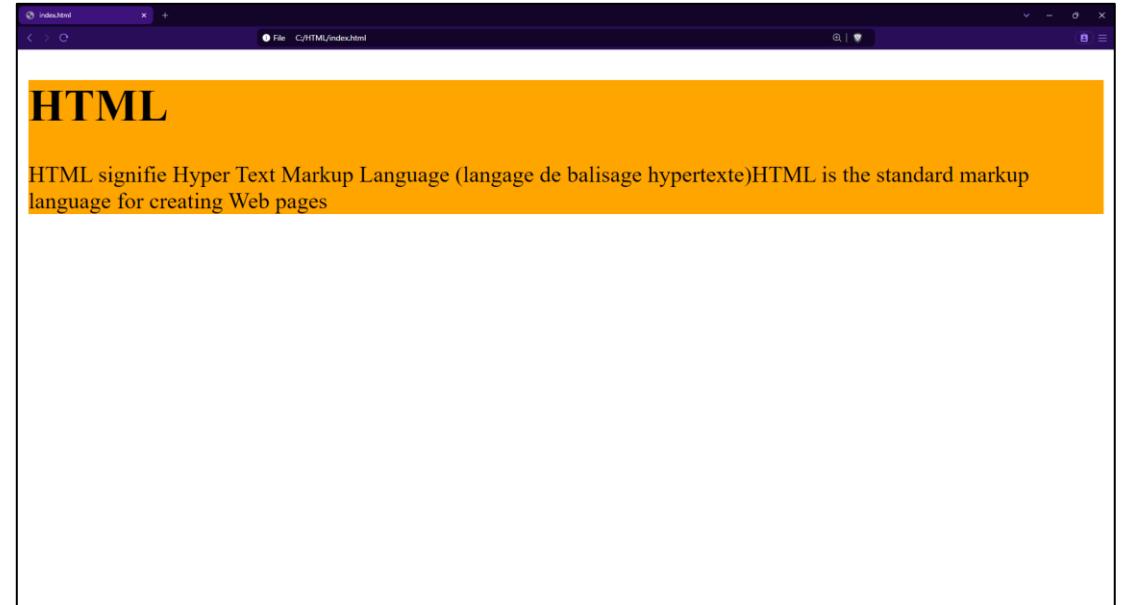
<div> comme conteneur:

L'élément **<div>** est souvent utilisé pour regrouper des sections d'une page Web.

Exemple: Un élément **<div>** avec des éléments HTML:



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       div {
6         background-color: orange;
7       }
8     </style>
9   </head>
10  <body>
11    <div>
12      <h1>HTML</h1>
13      <p>HTML signifie Hyper Text Markup Language (langage de balisage hypertexte)</p>
14      <p>HTML is the standard markup language for creating Web pages</p>
15    </div>
16  </body>
17 </html>
```



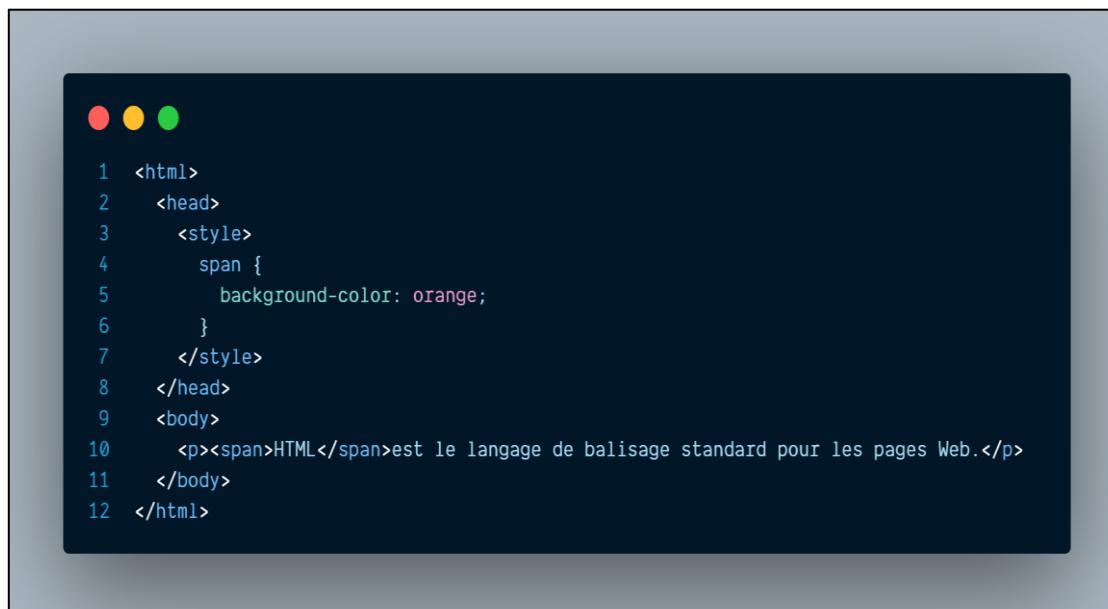
Élément HTML Span

L'élément `` est un conteneur en ligne utilisé pour baliser une partie d'un texte ou une partie d'un document.

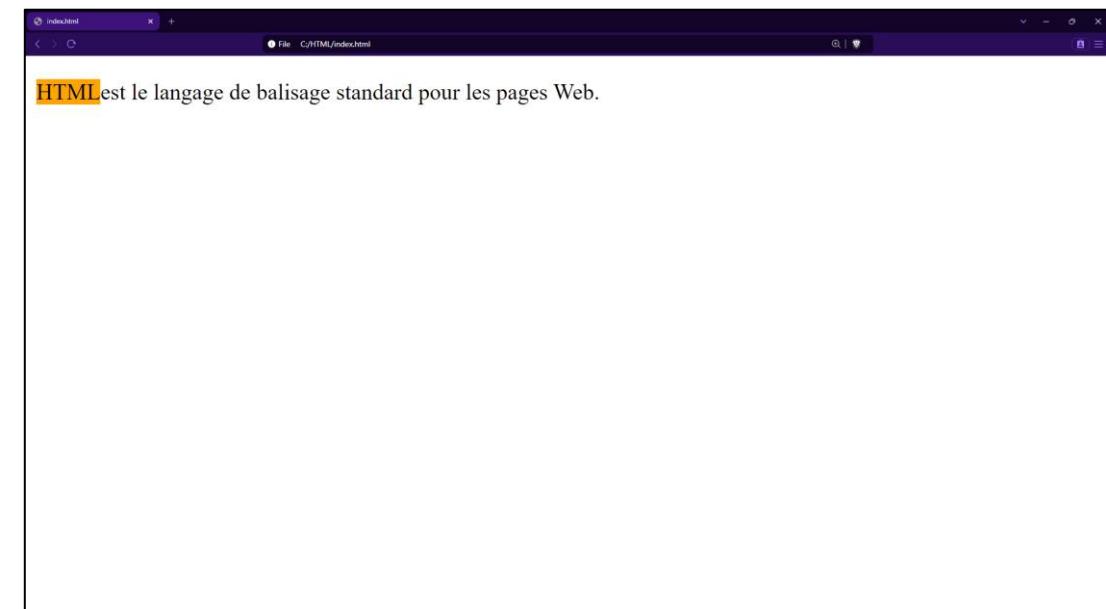
L'élément `` n'a pas d'attributs obligatoires, mais le `style`, la `classe` et l'`id` sont courants.

Lorsqu'il est utilisé avec CSS, l'élément `` peut être utilisé pour styliser des parties du texte:

Exemple:



```
1 <html>
2   <head>
3     <style>
4       span {
5         background-color: orange;
6       }
7     </style>
8   </head>
9   <body>
10    <p><span>HTML</span>est le langage de balisage standard pour les pages Web.</p>
11  </body>
12 </html>
```



L'attribut HTML class

L'attribut HTML `class` permet de spécifier une classe pour un élément HTML. Plusieurs éléments HTML peuvent partager la même classe.

Utilisation de l'attribut de classe:

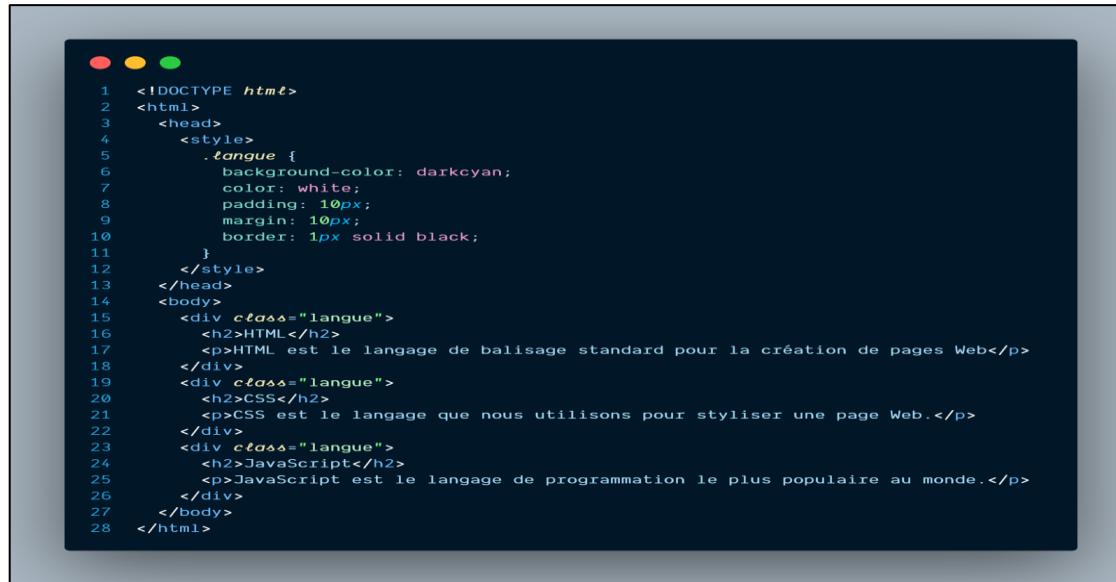
L'attribut `class` est souvent utilisé pour pointer vers un nom de classe dans une feuille de style. Il peut également être utilisé par un JavaScript pour accéder et manipuler des éléments portant le nom de classe spécifique.

Pour créer une classe, saisissez un point (.), suivi d'un nom de classe. Ensuite, définissez les propriétés CSS entre accolades {} :

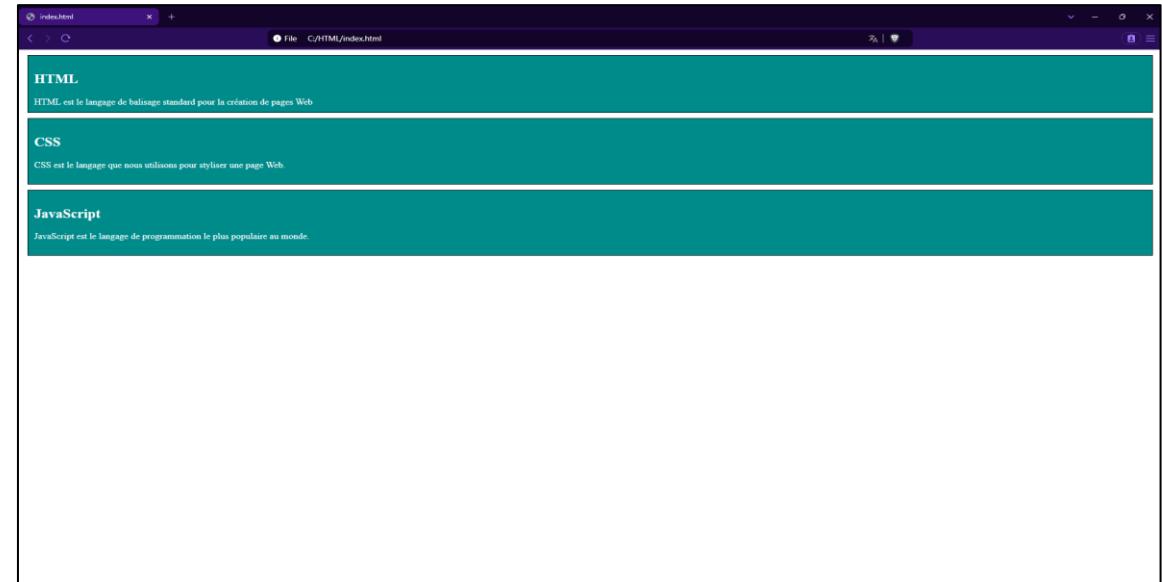
Dans l'exemple suivant, nous avons trois éléments `<div>` avec un attribut `class` ayant pour valeur "langue". Les trois éléments `<div>` seront tous stylisés de la même manière selon la définition de style `.langue` dans la section head:

Exemple:

L'attribut HTML class



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       .langue {
6         background-color: darkcyan;
7         color: white;
8         padding: 10px;
9         margin: 10px;
10        border: 1px solid black;
11      }
12    </style>
13  </head>
14  <body>
15    <div class="langue">
16      <h2>HTML</h2>
17      <p>HTML est le langage de balisage standard pour la création de pages Web</p>
18    </div>
19    <div class="langue">
20      <h2>CSS</h2>
21      <p>CSS est le langage que nous utilisons pour styliser une page Web.</p>
22    </div>
23    <div class="langue">
24      <h2>JavaScript</h2>
25      <p>JavaScript est le langage de programmation le plus populaire au monde.</p>
26    </div>
27  </body>
28 </html>
```



Astuce: l'attribut **class** peut être utilisé sur n'importe quel élément HTML.

Remarque: le nom de la classe est sensible à la casse

L'attribut HTML class

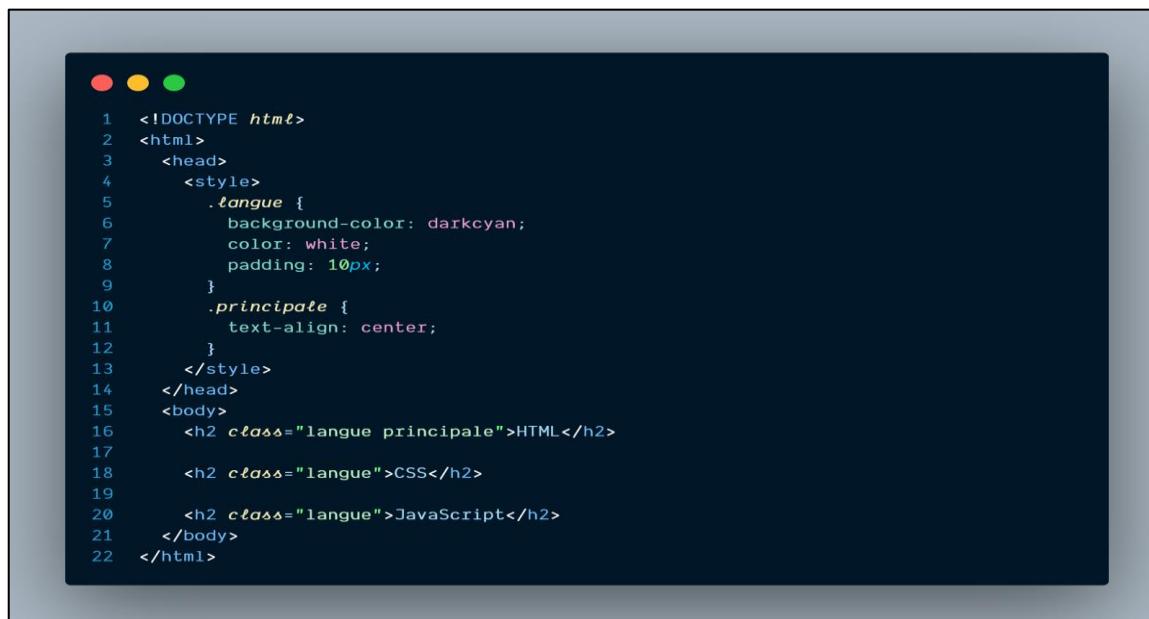
Classes multiples:

Les éléments HTML peuvent appartenir à plusieurs classes.

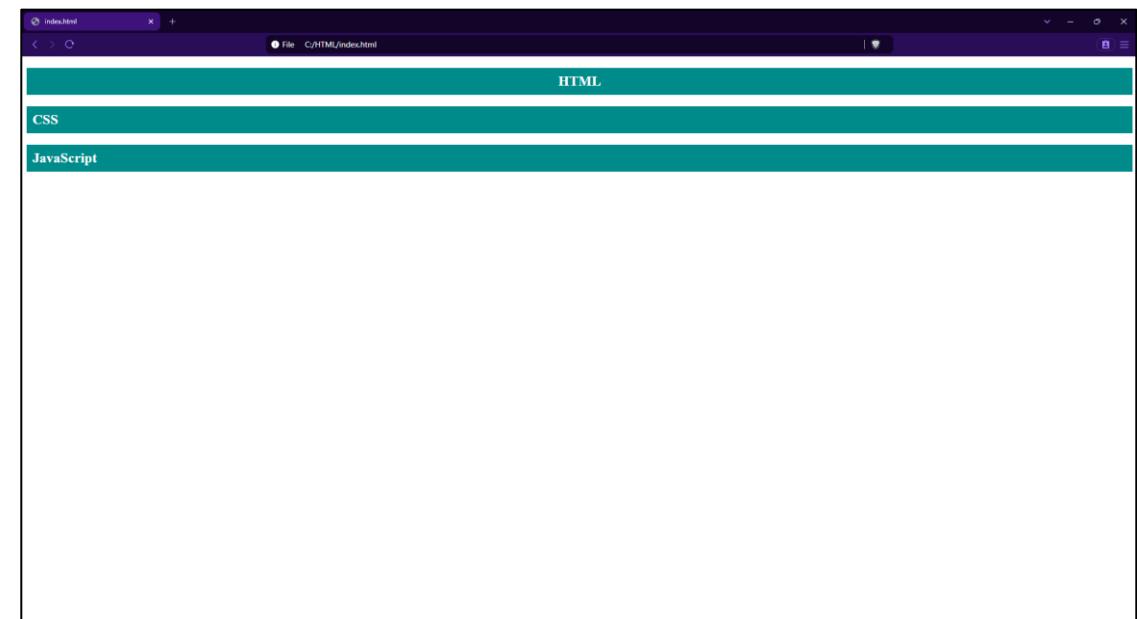
Pour définir plusieurs classes, séparez les noms de classe par un espace, par exemple `<div class="langue principale ">`. L'élément sera stylisé en fonction de toutes les classes spécifiées.

Dans l'exemple suivant, le premier élément `<h2>` appartient à la fois à la classe langue et à la classe principale, et obtiendra les styles CSS des deux classes:

Exemple:



```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <style>
5        .langue {
6          background-color: darkcyan;
7          color: white;
8          padding: 10px;
9        }
10       .principale {
11         text-align: center;
12       }
13     </style>
14   </head>
15   <body>
16     <h2 class="langue principale">HTML</h2>
17
18     <h2 class="langue">CSS</h2>
19
20     <h2 class="langue">JavaScript</h2>
21   </body>
22 </html>
```



L'attribut HTML class

Différents éléments peuvent partager la même classe:

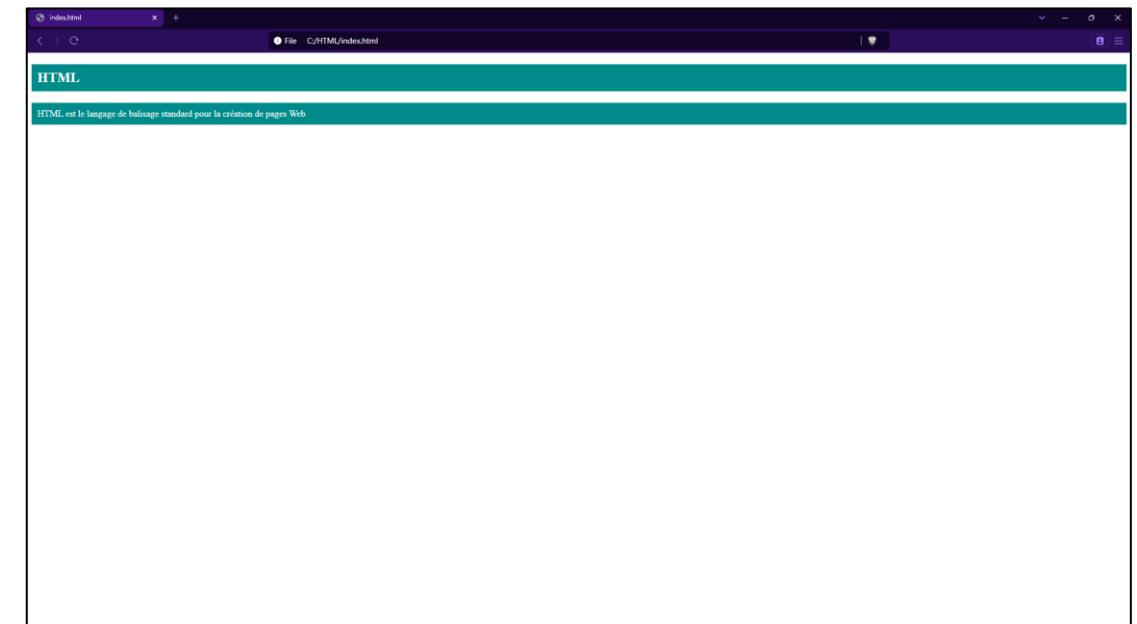
Différents éléments HTML peuvent pointer vers le même nom de classe.

Dans l'exemple suivant, `<h2>` et `<p>` pointent tous deux vers la classe "langue" et partageront le même style:

Exemple:



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       .langue {
6         background-color: darkcyan;
7         color: white;
8         padding: 10px;
9       }
10    </style>
11  </head>
12  <body>
13    <h2 class="langue">HTML</h2>
14    <p class="langue">HTML est le langage de balisage standard pour la création de pages Web</p>
15  </body>
16 </html>
```



L'attribut HTML id

L'attribut HTML `id` permet de spécifier un identifiant unique pour un élément HTML. Vous ne pouvez pas avoir plusieurs éléments avec le même identifiant dans un document HTML.

Utilisation de l'attribut `id`:

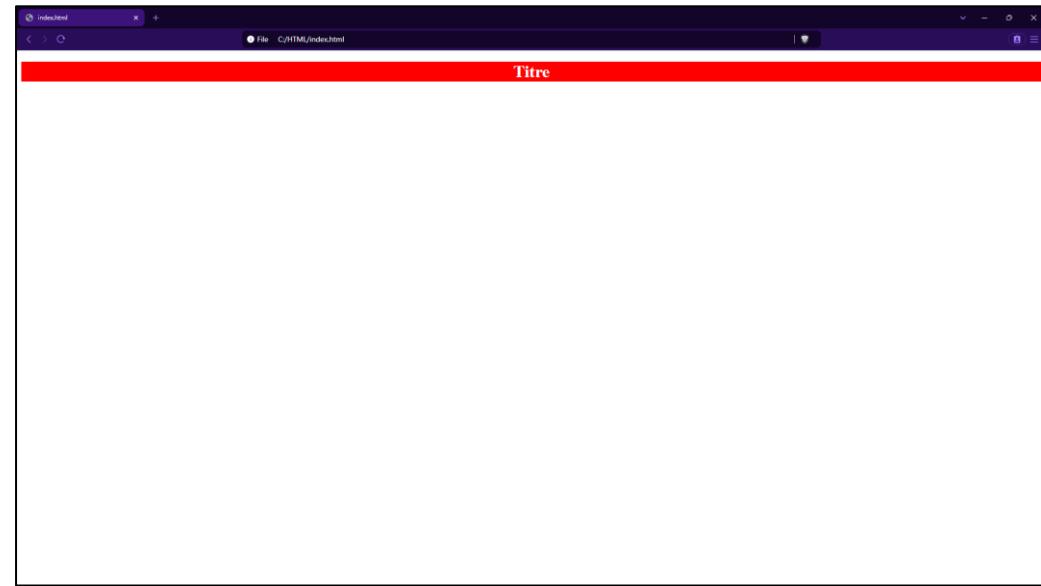
L'attribut `id` spécifie un identifiant unique pour un élément HTML. La valeur de l'attribut `id` doit être unique dans le document HTML.

L'attribut `id` est utilisé pour pointer vers une déclaration de style spécifique dans une feuille de style. Il est également utilisé par JavaScript pour accéder à l'élément avec l'identifiant spécifique et le manipuler.

La syntaxe de l'attribut `id` est la suivante: écrivez un caractère dièse (#), suivi d'un nom d'identifiant. Ensuite, définissez les propriétés CSS entre accolades {}. Dans l'exemple suivant, nous avons un élément `<h1>` qui pointe vers le nom d'identifiant `titre`. Cet élément `<h1>` sera stylisé selon la définition de style `#titre` dans la section head:

Exemple:

L'attribut HTML id



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       #titre {
6         background-color: red;
7         color: white;
8         text-align: center;
9       }
10      </style>
11    </head>
12    <body>
13      <h1 id="titre">Titre</h1>
14    </body>
15  </html>
```

Remarques:

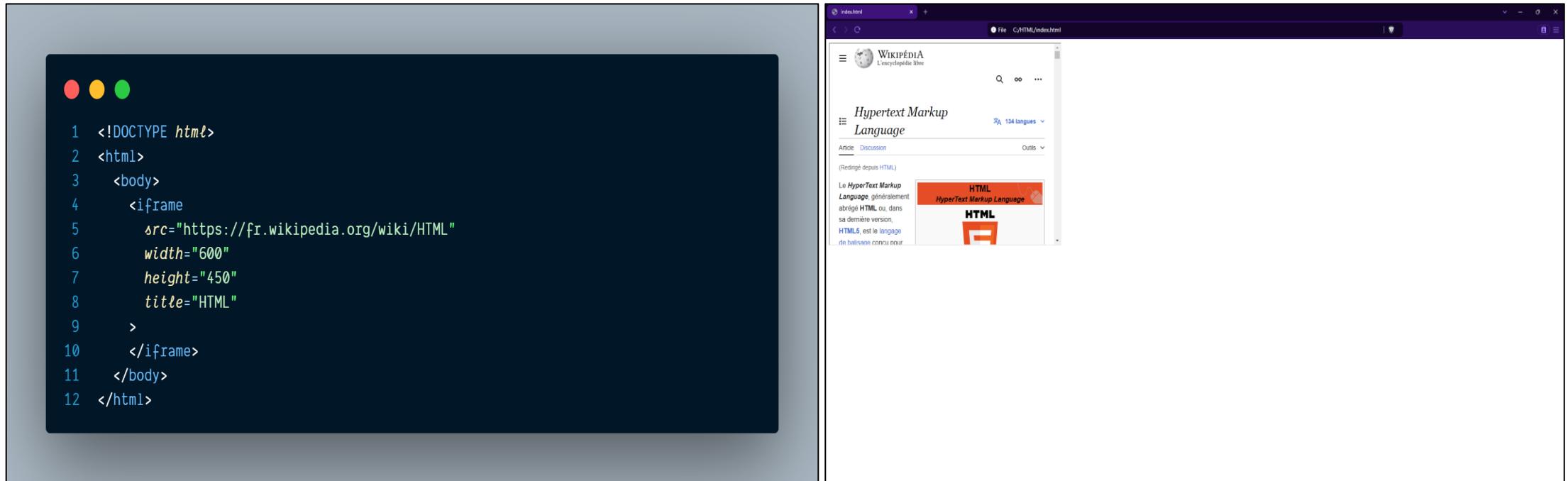
le nom de l'**id** est sensible à la casse!

le nom de l'**id** doit contenir au moins un caractère, ne peut pas commencer par un chiffre et ne doit pas contenir d'espaces (espaces, tabulations, etc.).

La principale différence entre les attributs **class** et **id** est qu'un nom de l'attribut **class** peut être utilisé par plusieurs éléments HTML, tandis qu'un nom de l'**id** ne doit être utilisé que par un seul élément HTML dans la page

HTML Iframes

Un iframe HTML est utilisé pour afficher une page Web dans une page Web.
Exemple: l'exemple ci-dessous intègre Wikipédia dans notre page Web:



The image shows a code editor on the left and a web browser on the right. The code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <iframe
5       src="https://fr.wikipedia.org/wiki/HTML"
6       width="600"
7       height="450"
8       title="HTML"
9     >
10    </iframe>
11  </body>
12 </html>
```

The browser window shows a Wikipedia article titled "Hypertext Markup Language". The page content includes a red banner at the top with the text "HyperText Markup Language" and "HTML".

HTML Iframes

Syntaxe HTML Iframe:

La balise HTML `<iframe>` spécifie un cadre en ligne.

Un cadre en ligne est utilisé pour intégrer un autre document dans le document HTML actuel.

Syntaxe:

```
<iframe src="url" title="description"></iframe>
```

Astuce: il est recommandé d'inclure systématiquement un attribut `title` pour l'élément `<iframe>`. Il est utilisé par les lecteurs d'écran pour lire le contenu de l'iframe.

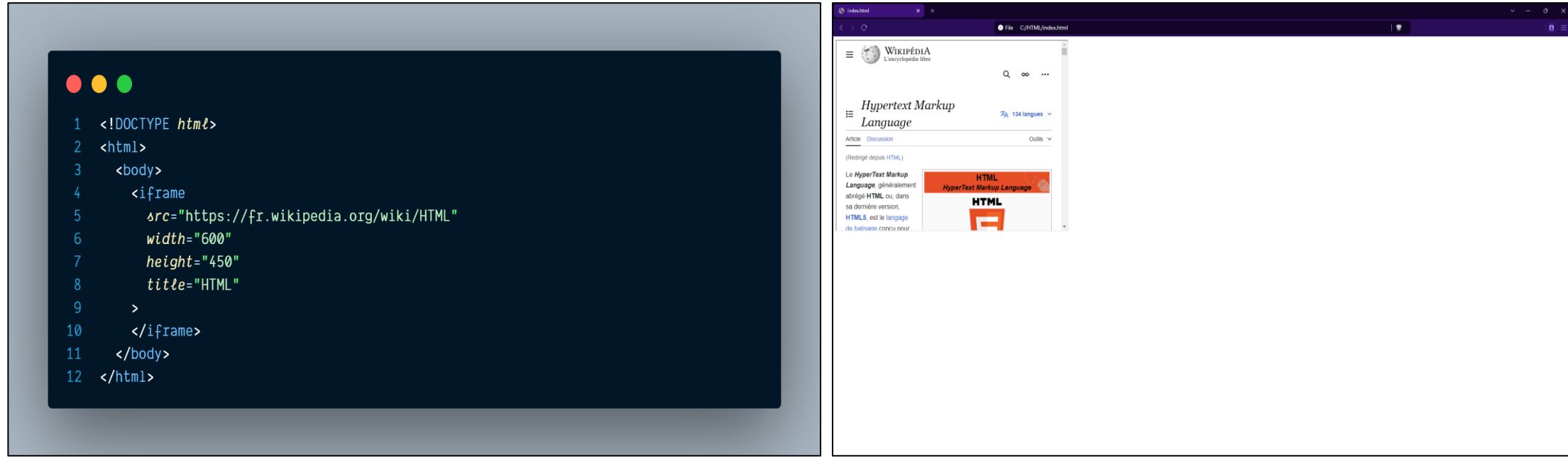
Iframe - Définir la hauteur et la largeur:

Utilisez les attributs `width` et `height` pour spécifier la taille de l'iframe ou vous pouvez ajouter l'attribut `style` et utiliser les propriétés CSS `width` et `height`.

La hauteur et la largeur sont spécifiées en pixels par défaut:

Exemple:

HTML Iframes



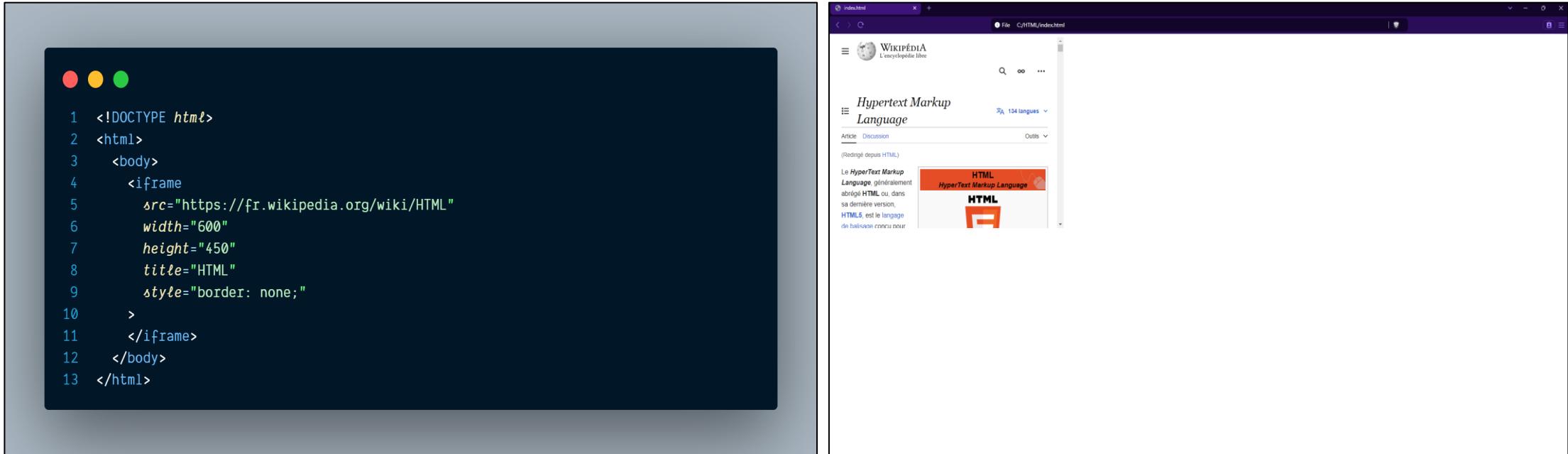
Iframe - Supprimer la bordure:

Par défaut, un iframe est entouré d'une bordure.

Pour supprimer la bordure, ajoutez l'attribut **style** et utilisez la propriété CSS **border**:

Exemple:

HTML Iframes



The image shows a Mac OS X desktop environment. On the left, a terminal window displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <iframe
5       src="https://fr.wikipedia.org/wiki/HTML"
6       width="600"
7       height="450"
8       title="HTML"
9       style="border: none;">
10    </iframe>
11  </body>
12 </html>
```

On the right, a web browser window is open, showing the rendered content of the iframe. The page title is "WIKIPÉDIA L'encyclopédie libre". The main content area displays the Wikipedia article for "Hypertext Markup Language". A red box highlights the "HTML" section of the page.

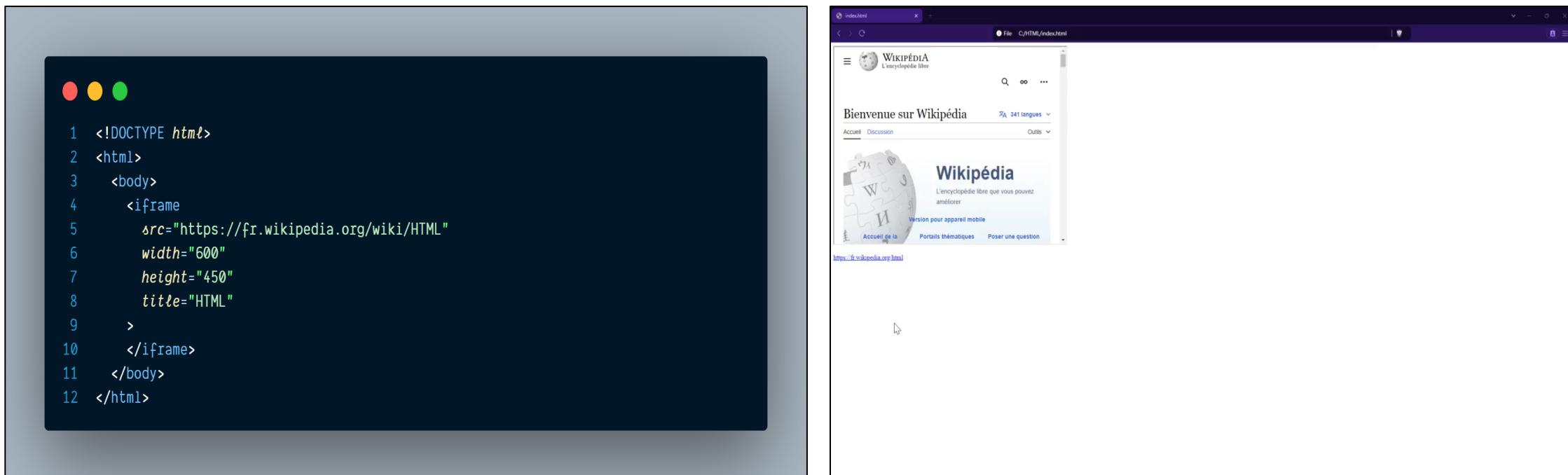
HTML Iframes

Iframe - Cible pour un lien:

Un iframe peut être utilisé comme cadre cible pour un lien.

L'attribut **target** du lien doit faire référence à l'attribut **name** de l'iframe:

Exemple:

A screenshot of a Windows desktop. On the left, a dark-themed code editor window displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <iframe
5       src="https://fr.wikipedia.org/wiki/HTML"
6       width="600"
7       height="450"
8       title="HTML"
9     >
10    </iframe>
11  </body>
12 </html>
```

On the right, a Microsoft Edge browser window shows the Wikipedia homepage. The URL bar at the top of the browser window displays "index.html".

The browser window shows the Wikipedia homepage with the title "Wikipédia" and the subtitle "L'encyclopédie libre que vous pouvez améliorer". Below the title, there is a large image of a brain and the text "Wikipédia". At the bottom of the page, there are links for "Accueil de la Wikipédia", "Portails thématiques", and "Poser une question".

Chemins d'accès aux fichiers HTML

Un chemin de fichier décrit l'emplacement d'un fichier dans la structure de dossiers d'un site Web.

Chemin	Description
	Le fichier "picture.jpg" se trouve dans le même dossier que la page actuelle
	Le fichier "picture.jpg" se trouve dans le dossier images du dossier actuel
	Le fichier "picture.jpg" se trouve dans le dossier images à la racine du site Web actuel.
	Le fichier "picture.jpg" se trouve dans le dossier situé un niveau au-dessus du dossier actuel

Chemins d'accès aux fichiers HTML:

Un chemin d'accès à un fichier décrit l'emplacement d'un fichier dans la structure de dossiers d'un site Web.

Les chemins d'accès aux fichiers sont utilisés lors de la création de liens vers des fichiers externes, tels que:

Pages Web

Images

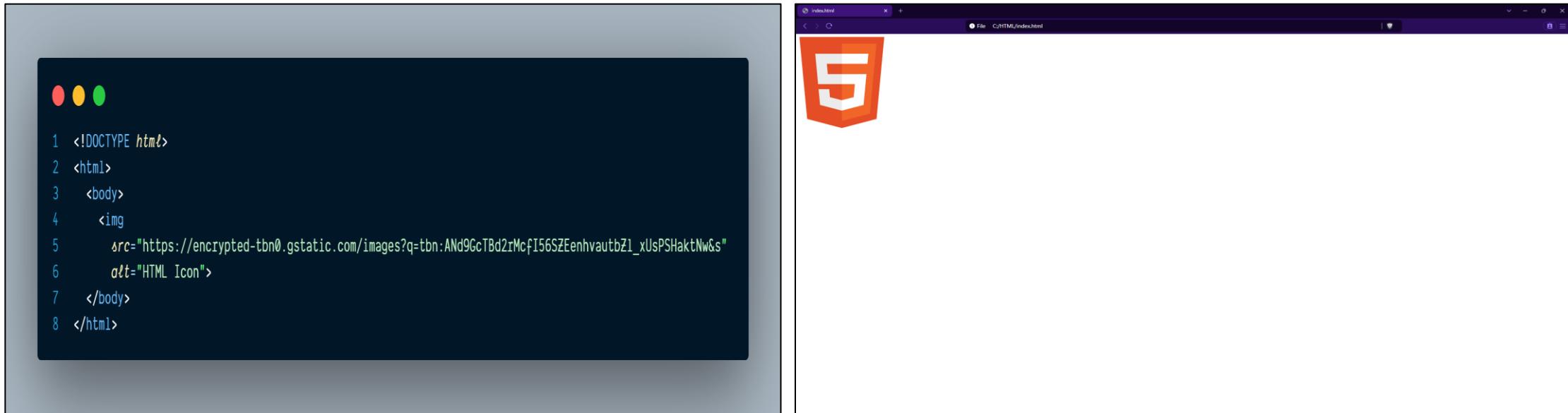
Feuilles de style

JavaScripts

Chemins d'accès aux fichiers HTML

Chemins de fichiers absous:

Un chemin de fichier absolu est l'URL complète d'un fichier:



Bonnes pratiques:

Il est recommandé d'utiliser des chemins de fichiers relatifs (si possible).

Lorsque vous utilisez des chemins de fichiers relatifs, vos pages Web ne seront pas liées à votre URL de base actuelle. Tous les liens fonctionneront sur votre propre ordinateur (localhost) ainsi que sur votre domaine public actuel et vos futurs domaines publics.

HTML - L'élément Head

L'élément HTML `<head>` est un conteneur pour les éléments suivants : `<title>`, `<style>`, `<meta>`, `<link>`, `<script>` et `<base>`.

L'élément HTML `<head>`:

L'élément `<head>` est un conteneur de métadonnées (données sur les données) et est placé entre la balise `<html>` et la balise `<body>` .

Les métadonnées HTML sont des données sur le document HTML. Les métadonnées ne sont pas affichées.

Les métadonnées définissent généralement le titre du document, le jeu de caractères, les styles, les scripts et d'autres méta-informations.

L'élément HTML `<title>`:

L'élément `<title>` définit le titre du document. Le titre doit être uniquement textuel et s'afficher dans la barre de titre du navigateur ou dans l'onglet de la page.

L'élément `<title>` est obligatoire dans les documents HTML!

Le contenu d'un titre de page est très important pour l'optimisation des moteurs de recherche (SEO)! Le titre de la page est utilisé par les algorithmes des moteurs de recherche pour décider de l'ordre dans lequel les pages sont répertoriées dans les résultats de recherche.

L'élément `<title>`:

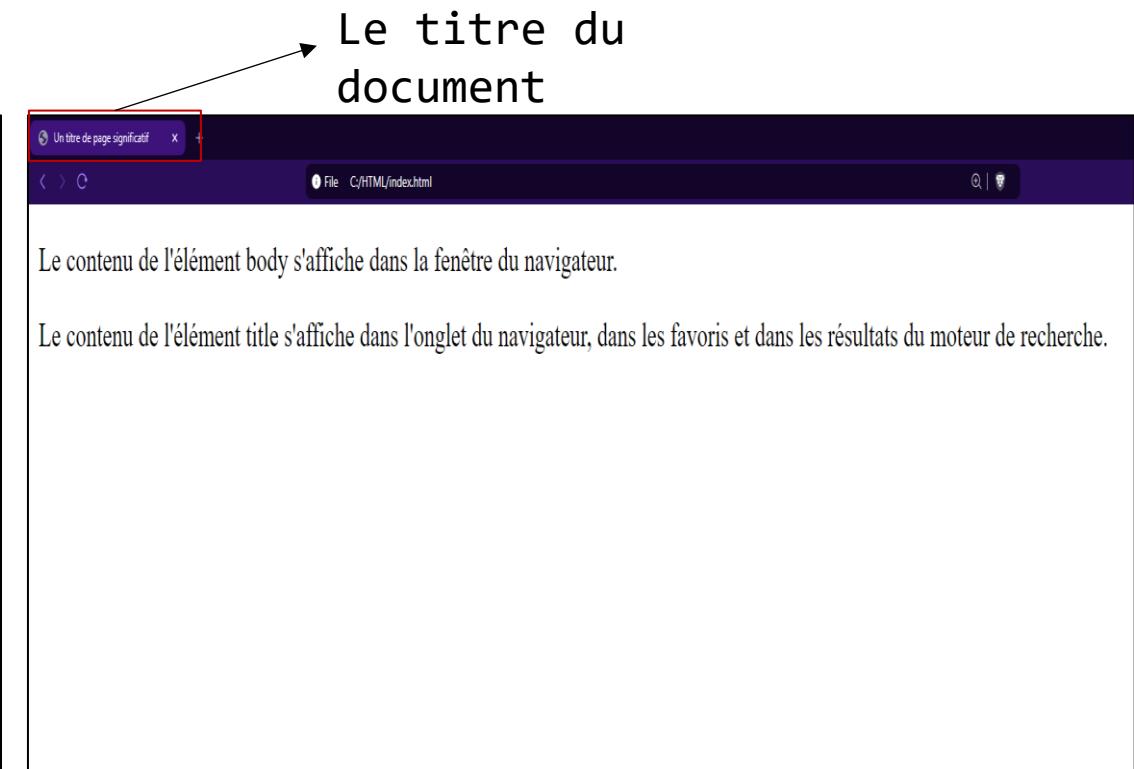
- définit un titre dans la barre d'outils du navigateur
- fournit un titre pour la page lorsqu'elle est ajoutée aux favoris
- affiche un titre pour la page dans les résultats des moteurs de recherche

Essayez donc de rendre le titre aussi précis et significatif que possible !

HTML - L'élément Head

Un document HTML simple:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Un titre de page significatif</title>
5   </head>
6   <body>
7     <p>Le contenu de l'élément body s'affiche dans la fenêtre du navigateur.</p>
8     <p>
9       Le contenu de l'élément title s'affiche dans l'onglet du navigateur, dans
10      les favoris et dans les résultats du moteur de recherche.
11     </p>
12   </body>
13 </html>
```



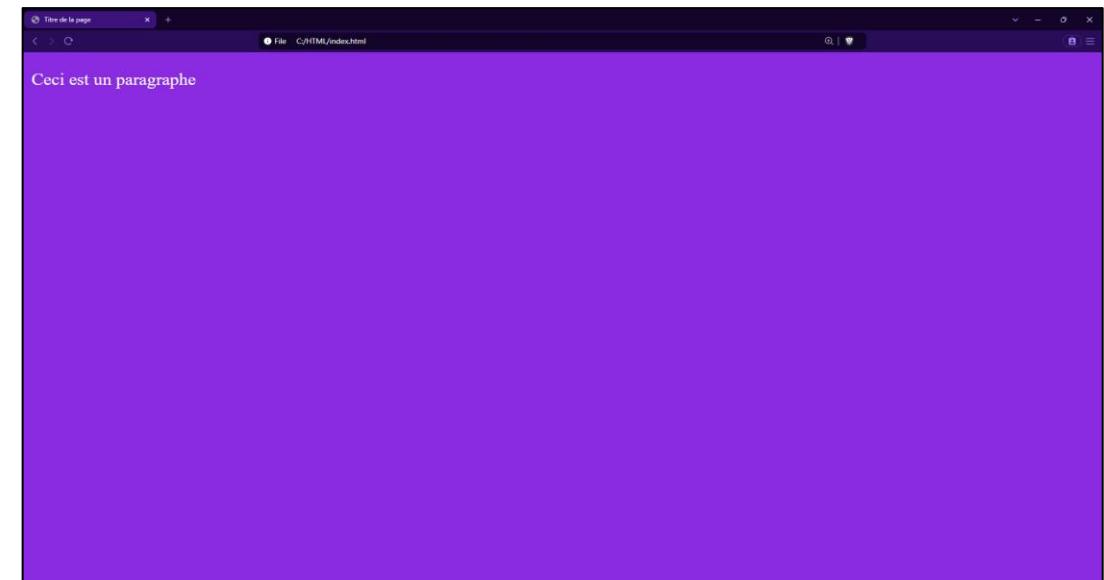
HTML - L'élément Head

L'élément HTML <style>:

L'élément `<style>` est utilisé pour définir les informations de style pour une seule page HTML:



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Titre de la page</title>
5      <style>
6          body { background-color: blueviolet; }
7          p { color: white; }
8      </style>
9  </head>
10 <body>
11     <p>Ceci est un paragraphe</p>
12 </body>
13 </html>
```



HTML - L'élément Head

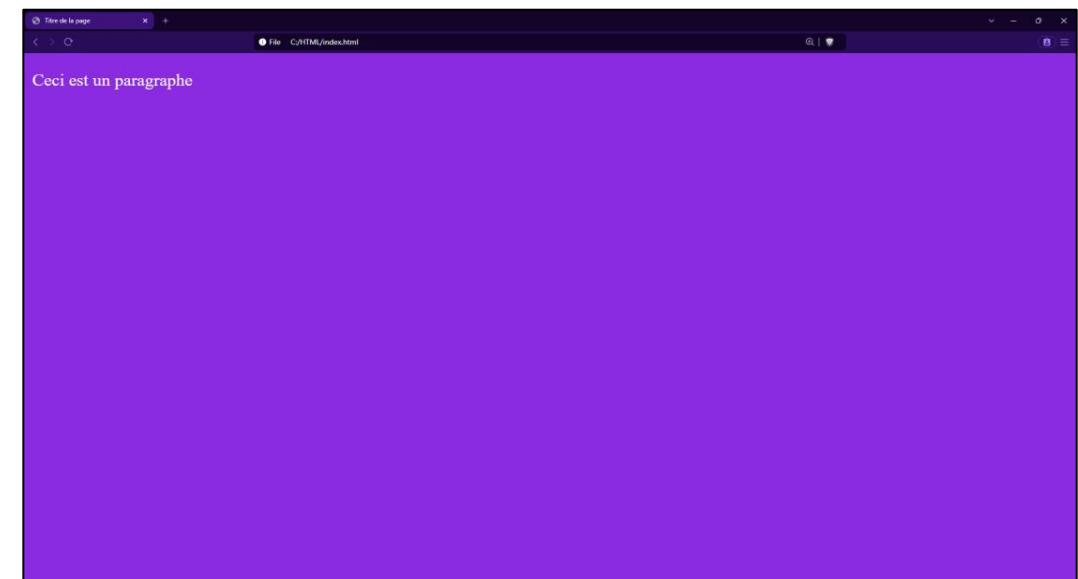
L'élément HTML <link>:

L'élément `<link>` définit la relation entre le document actuel et une ressource externe.

La balise `<link>` est le plus souvent utilisée pour créer un lien vers des feuilles de style externes :



```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Titre de la page</title>
5      <link rel="stylesheet" href="main.css" />
6    </head>
7    <body>
8      <p>Ceci est un paragraphe</p>
9    </body>
10   </html>
```



HTML - L'élément Head

L'élément HTML <meta>:

L'élément `<meta>` est généralement utilisé pour spécifier le jeu de caractères, la description de la page, les mots-clés, l'auteur du document et les paramètres de la fenêtre d'affichage.

Les métadonnées ne seront pas affichées sur la page, mais sont utilisées par les navigateurs (comment afficher le contenu ou recharger la page), par les moteurs de recherche (mots-clés) et d'autres services Web.

Exemples:

Définir le jeu de caractères utilisé:

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

Définir des mots clés pour les moteurs de recherche:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

Définissez une description de votre page Web:

```
<meta name="description" content="une description de votre page Web">
```

Définir l'auteur d'une page:

```
<meta name="author" content="Marcel Savoie">
```

Actualiser le document toutes les 30 secondes:

```
<meta http-equiv="refresh" content="30">
```

Paramétriser la fenêtre d'affichage pour que votre site Web soit beau sur tous les appareils:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

HTML - L'élément Head

L'élément HTML <script>:

L'élément `<script>` est utilisé pour définir des JavaScripts côté client.

L'élément HTML <base>:

L'élément `<base>` spécifie l'URL de base et/ou la cible de toutes les URL relatives d'une page.

La balise `<base>` doit contenir soit un attribut `href`, soit un attribut `target`, soit les deux.

Il ne peut y avoir qu'un seul élément `<base>` dans un document.

Éléments sémantiques HTML

Éléments sémantiques = éléments ayant un sens.

Que sont les éléments sémantiques?

Un élément sémantique décrit clairement sa signification à la fois pour le navigateur et pour le développeur.

Exemples d'éléments non sémantiques : `<div>` et `` - Ne disent rien sur leur contenu.

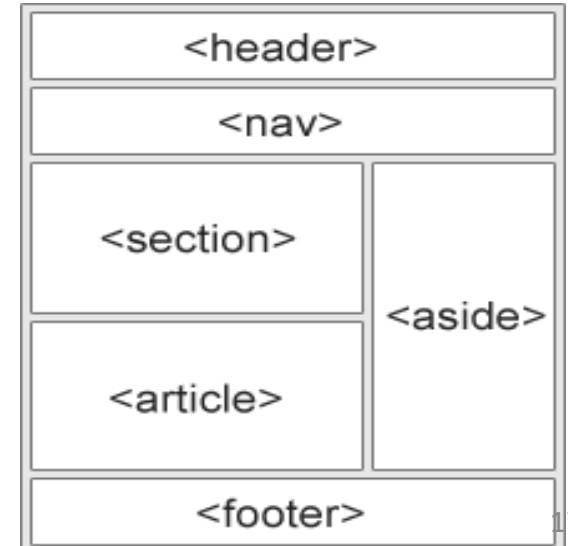
Exemples d'éléments sémantiques : `<form>`, `<table>` et `<article>` - Définissent clairement leur contenu.

Éléments sémantiques en HTML

De nombreux sites Web contiennent du code HTML tel que: `<div id="nav">` `<div class="header">` `<div id="footer">` pour indiquer la navigation, l'en-tête et le pied de page.

En HTML, certains éléments sémantiques peuvent être utilisés pour définir différentes parties d'une page Web :

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`



Éléments sémantiques HTML

Élément HTML <section>:

L'élément <section> définit une section dans un document.

Selon la documentation HTML du W3C: ."Une section est un regroupement thématique de contenu, généralement avec un titre."

Exemples d'utilisation d'un élément <section>:

- Chapitres
- Introduction
- Articles d'actualité
- Coordonnées

Une page Web peut normalement être divisée en sections pour l'introduction, le contenu et les coordonnées.

Élément HTML <article>:

L'élément <article> spécifie un contenu indépendant et autonome.

Un article doit avoir un sens en soi et il doit être possible de le distribuer indépendamment du reste du site Web.

Exemples d'utilisation de l'élément <article>:

- Messages de forum
- Messages de blog
- Commentaires d'utilisateurs
- Fiches de produits
- Article de journal

Éléments sémantiques HTML

Élément HTML <header>:

L'élément <header> représente un conteneur pour le contenu d'introduction ou un ensemble de liens de navigation.

Un élément <header> contient généralement:

- un ou plusieurs éléments d'en-tête (<h1> - <h6>)
- un logo ou une icône
- des informations sur l'auteur

Remarque: vous pouvez avoir plusieurs éléments <header> dans un document HTML.

Élément HTML <footer>:

L'élément <footer> définit un pied de page pour un document ou une section.

Un élément <footer> contient généralement :

- informations sur l'auteur
- informations sur les droits d'auteur
- informations de contact
- plan du site
- retour en haut des liens
- documents associés

Remarque: Vous pouvez avoir plusieurs éléments <footer> dans un même document.

Éléments sémantiques HTML

Élément HTML <nav>:

L'élément <nav> définit un ensemble de liens de navigation.

Remarque: tous les liens d'un document ne doivent pas être à l'intérieur d'un élément <nav>. L'élément <nav> est destiné uniquement aux blocs principaux de liens de navigation.

Les navigateurs, tels que les lecteurs d'écran pour les utilisateurs handicapés, peuvent utiliser cet élément pour déterminer s'il faut omettre le rendu initial de ce contenu.

Élément HTML <aside>:

L'élément <aside> définit un contenu en plus du contenu dans lequel il est placé (comme une barre latérale).

Le contenu <aside> doit être indirectement lié au contenu environnant.

Éléments HTML <figure> et <figcaption>:

La balise <figure> spécifie un contenu autonome, comme des illustrations, des diagrammes, des photos, des listes de codes, etc.

La balise <figcaption> définit une légende pour un élément <figure>. L'élément <figcaption> peut être placé comme premier ou dernier enfant d'un élément <figure>.

Éléments sémantiques HTML

Pourquoi des éléments sémantiques?

Selon le W3C: "Un Web sémantique permet de partager et de réutiliser les données entre les applications, les entreprises et les communautés."

Guide de style HTML

Un code HTML cohérent, propre et ordonné permet aux autres de lire et de comprendre plus facilement votre code.

Voici quelques conseils et astuces pour créer un bon code HTML.

Toujours déclarer le type de document:

Déclarez toujours le type de document comme première ligne de votre document.

Le type de document correct pour HTML est:

```
<!DOCTYPE html>
```

Utiliser des noms d'éléments en minuscules:

Le HTML permet de mélanger des lettres majuscules et minuscules dans les noms d'éléments.

Nous recommandons cependant d'utiliser des noms d'éléments en minuscules, car :

- Le mélange de noms en majuscules et en minuscules est peu esthétique
- Les développeurs utilisent généralement des noms en minuscules
- Les minuscules ont l'air plus propres
- Les minuscules sont plus faciles à écrire

Guide de style HTML

Fermer tous les éléments HTML:

En HTML, vous n'êtes pas obligé de fermer tous les éléments (par exemple l'élément `<p>`). Cependant, la meilleure pratique consiste à fermer tous les éléments HTML.

Utiliser des noms d'attributs en minuscules:

Le HTML permet de mélanger des lettres majuscules et minuscules dans les noms d'attributs. Cependant, la meilleure pratique consiste à utiliser des noms d'attributs en minuscules, car:

- Le mélange de noms en majuscules et en minuscules est peu esthétique
- Les développeurs utilisent généralement des noms en minuscules
- Les minuscules ont l'air plus propres
- Les minuscules sont plus faciles à écrire

Toujours citer les valeurs d'attribut:

Le HTML autorise les valeurs d'attribut sans guillemets.

Cependant, la meilleure pratique consiste à citer les valeurs d'attribut, car:

- Les développeurs citent généralement les valeurs d'attribut
- Les valeurs entre guillemets sont plus faciles à lire
- Vous devez utiliser des guillemets si la valeur contient des espaces

Guide de style HTML

Spécifiez toujours l'alt, la largeur et la hauteur des images:

Spécifiez toujours l'attribut **alt** pour les images. Cet attribut est important si l'image ne peut pas être affichée pour une raison quelconque.

Définissez également toujours la largeur et la hauteur des images. Cela réduit le scintillement, car le navigateur peut réservé de l'espace pour l'image avant le chargement.

Espaces et signes d'égalité:

Le HTML autorise les espaces autour des signes d'égalité. Mais l'absence d'espaces est plus facile à lire et permet de mieux regrouper les entités.

Évitez les lignes de code trop longues:

Lorsque vous utilisez un éditeur HTML, il n'est PAS pratique de faire défiler vers la droite et la gauche pour lire le code HTML.

Essayez d'éviter les lignes de code trop longues.

Lignes vides et retrait:

N'ajoutez pas de lignes vides, d'espaces ou de retraits sans raison.

Pour une meilleure lisibilité, ajoutez des lignes vides pour séparer les blocs de code volumineux ou logiques.

Pour une meilleure lisibilité, ajoutez deux espaces de retrait. N'utilisez pas la touche Tab.

Guide de style HTML

Ne sautez jamais l'élément <title>:

L'élément <title> est obligatoire en HTML.

Le contenu d'un titre de page est très important pour l'optimisation des moteurs de recherche (SEO) ! Le titre de la page est utilisé par les algorithmes des moteurs de recherche pour décider de l'ordre dans lequel les pages sont répertoriées dans les résultats de recherche.

L'élément <title>:

- définit un titre dans la barre d'outils du navigateur
- fournit un titre pour la page lorsqu'elle est ajoutée aux favoris
- affiche un titre pour la page dans les résultats des moteurs de recherche
- Essayez donc de rendre le titre aussi précis et significatif que possible

Vous omettez <html> et <body>?

Une page HTML sera validée sans les balises <html> et <body>

Cependant, la meilleure pratique consiste toujours à ajouter les balises <html> et <body> !

L'omission de <body> peut produire des erreurs dans les navigateurs plus anciens.

L'omission de <html> et <body> peut également faire planter les logiciels DOM et XML.

Guide de style HTML

Omission de `<head>`?

La balise HTML `<head>` peut également être omise.

Les navigateurs ajouteront tous les éléments avant `<body>`, à un élément `<head>` par défaut.

Cependant, la meilleure pratique consiste à utiliser la balise `<head>`.

Fermer les éléments HTML vides?

En HTML, il est facultatif de fermer les éléments vides.

Ajoutez l'attribut lang:

Vous devez toujours inclure l'attribut lang dans la balise `<html>` pour déclarer la langue de la page Web. Cela est destiné à aider les moteurs de recherche et les navigateurs.

Métadonnées:

Pour garantir une interprétation correcte et une indexation correcte des moteurs de recherche, la langue et le codage des caractères `<meta charset="charset">` doivent être définis le plus tôt possible dans un document HTML.

Guide de style HTML

Définition de la viewport:

La viewport est la zone visible d'une page Web par l'utilisateur. Elle varie selon l'appareil - elle sera plus petite sur un téléphone mobile que sur un écran d'ordinateur.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Cela donne au navigateur des instructions sur la façon de contrôler les dimensions et la mise à l'échelle de la page.

La partie `width=device-width` définit la largeur de la page pour qu'elle suive la largeur de l'écran de l'appareil (qui varie en fonction de l'appareil).

La partie `initial-scale=1.0` définit le niveau de zoom initial lorsque la page est chargée pour la première fois par le navigateur.

Guide de style HTML

Commentaires HTML:

Les commentaires courts doivent être écrits sur une seule ligne, comme ceci :

```
<!-- Ceci est un commentaire -->
```

Les commentaires qui s'étendent sur plusieurs lignes doivent être écrits comme ceci :

```
<!--  
    Voici un exemple de commentaire long. Voici un exemple de commentaire long.  
    Voici un exemple de commentaire long. Voici un exemple de commentaire long.  
-->
```

Les commentaires longs sont plus faciles à observer s'ils sont indentés de deux espaces.

Utilisation des feuilles de style:

Utilisez une syntaxe simple pour créer des liens vers des feuilles de style (l'attribut **type** n'est pas nécessaire) :

```
<link rel="stylesheet" href="styles.css">
```

Les règles CSS courtes peuvent être écrites de manière compressée, comme ceci :

Guide de style HTML

```
.class { property: value; }
```

Les règles CSS longues doivent être écrites sur plusieurs lignes:

```
.class {  
    property: value;  
}
```

- Placez la parenthèse ouvrante sur la même ligne que le sélecteur
- Utilisez un espace avant la parenthèse ouvrante
- Utilisez deux espaces d'indentation
- Utilisez un point-virgule après chaque paire propriété-valeur, y compris la dernière
- N'utilisez des guillemets autour des valeurs que si la valeur contient des espaces
- Placez la parenthèse fermante sur une nouvelle ligne, sans espaces de début

Guide de style HTML

Chargement de JavaScript en HTML:

Utilisez une syntaxe simple pour charger des scripts externes (l'attribut **type** n'est pas nécessaire):

```
<script src="main.js">
```

Utilisez des noms de fichiers en minuscules:

Certains serveurs Web (Apache, Unix) sont sensibles à la casse pour les noms de fichiers: "london.jpg" ne peut pas être consulté sous "London.jpg".

D'autres serveurs Web (Microsoft, IIS) ne sont pas sensibles à la casse : "london.jpg" peut être consulté sous "London.jpg".

Si vous utilisez un mélange de majuscules et de minuscules, vous devez en être conscient.

Si vous passez d'un serveur insensible à la casse à un serveur sensible à la casse, même de petites erreurs endommageront votre site Web !

Pour éviter ces problèmes, utilisez toujours des noms de fichiers en minuscules !

Guide de style HTML

Extensions de fichiers:

Les fichiers HTML doivent avoir une extension .html (.htm est autorisé).

Les fichiers CSS doivent avoir une extension .css.

Les fichiers JavaScript doivent avoir une extension .js.

Différences entre .htm et .html?

Il n'y a aucune différence entre les extensions de fichier .htm et .html!

Les deux seront traités comme du HTML par n'importe quel navigateur Web et serveur Web.

Entités HTML

Les caractères réservés en HTML doivent être remplacés par des entités:

< (inférieur à) (less than) = <
> (supérieur à) (greater than) = >

Entités de caractères HTML:

Certains caractères sont réservés en HTML.

Si vous utilisez les signes inférieur à (<) ou supérieur à (>) dans votre texte HTML, le navigateur peut les mélanger avec des balises.

Les noms ou numéros d'entités peuvent être utilisés pour afficher les caractères HTML réservés.

Les noms d'entités ressemblent à ceci:

```
&nom_entité;
```

Les numéros d'entité ressemblent à ceci:

```
&#numéro_entité;
```

Pour afficher un signe inférieur à (<), il faut écrire: < ou <

Les noms d'entités sont plus faciles à retenir que les numéros d'entités.

Entités HTML

Espace insécable:

Une entité HTML couramment utilisée est l'espace insécable:

Un espace insécable est un espace qui ne se divise pas en nouvelle ligne.

Deux mots séparés par un espace insécable restent collés (ils ne se divisent pas en nouvelle ligne). C'est pratique lorsque la séparation des mots risque d'être perturbante.

Exemples :

10 km/h

22 h

Une autre utilisation courante de l'espace insécable est d'empêcher les navigateurs de tronquer les espaces dans les pages HTML.

Si vous écrivez 10 espaces dans votre texte, le navigateur en supprimera 9. Pour ajouter de véritables espaces à votre texte, vous pouvez utiliser l'entité caractère .

Le trait d'union insécable (‑) est utilisé pour définir un caractère trait d'union (-) qui ne se coupe pas en une nouvelle ligne.

Entités HTML

Quelques entités de caractères HTML utiles:

Résultat	Description	Nom	Nombre
	espace insécable	&nbsp	
<	inférieur à	<	<
>	supérieur à	>	>
&	esperluette	&	&
"	guillemets doubles	"	"
'	guillemet simple	'	'
©	droits d'auteur	©	©
®	marque déposée	®	®

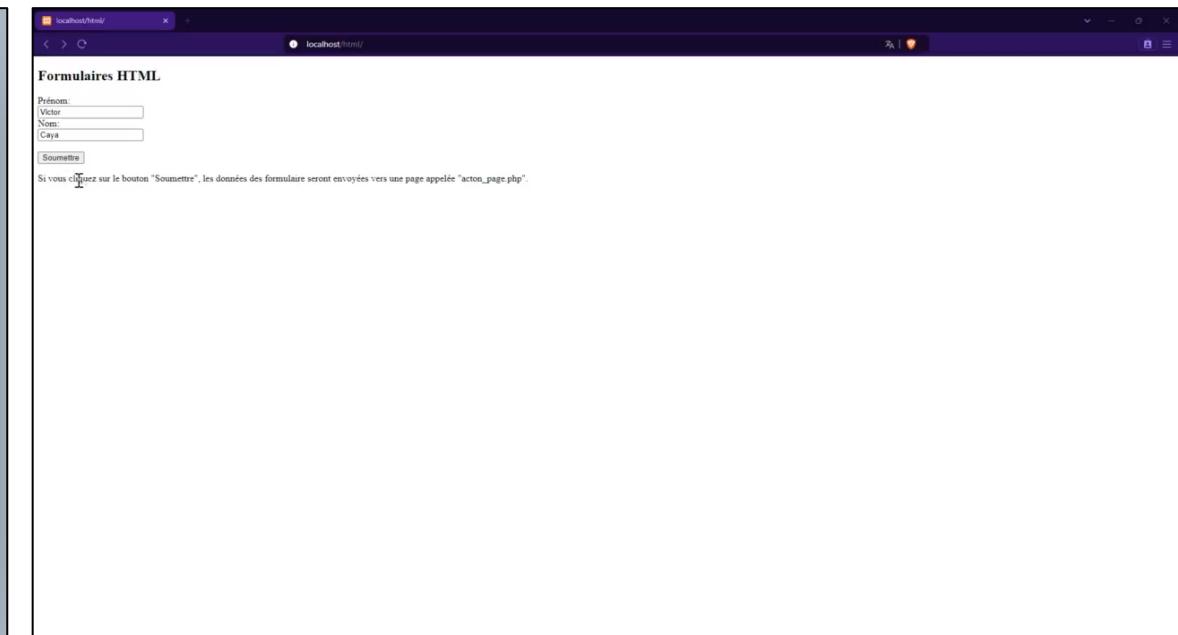
Remarque: Les noms d'entités sont sensibles à la casse.

Formulaires HTML

Un formulaire HTML est utilisé pour recueillir les données saisies par l'utilisateur. Les données saisies par l'utilisateur sont généralement envoyées à un serveur pour traitement.

Exemple:

```
1  <!DOCTYPE html>
2  <html>
3    <body>
4      <h2>Formulaires HTML</h2>
5
6      <form action="action_page.php">
7        <label for="prenom">Prénom:</label><br />
8        <input type="text" name="prenom" id="prenom" value="Victor" /><br />
9        <label for="nom">Nom:</label><br />
10       <input type="text" name="nom" id="nom" value="Caya" /><br /><br />
11       <input type="submit" value="Soumettre" />
12     </form>
13
14   <p>
15     Si vous cliquez sur le bouton "Soumettre", les données du formulaire
16     seront envoyées vers une page appelée "action_page.php" et la page affichera les données.
17   </p>
18   </body>
19 </html>
```



Formulaires HTML

L'élément <form>:

L'élément HTML <form> est utilisé pour créer un formulaire HTML pour la saisie de données par l'utilisateur:

```
<form>
    form elements
</form>
```

L'élément <form> est un conteneur pour différents types d'éléments d'entrée, tels que : des champs de texte, des cases à cocher, des boutons radio, des boutons d'envoi, etc.

L'élément <input>:

L'élément HTML <input> est l'élément de formulaire le plus utilisé.

Un élément <input> peut être affiché de plusieurs manières, selon l'attribut **type**.

Voici quelques exemples :

Formulaires HTML

Type	Description
<input type="text">	Affiche un champ de saisie de texte sur une seule ligne
<input type="radio">	Affiche un bouton radio (pour sélectionner l'une des nombreuses options)
<input type="checkbox">	Affiche une case à cocher (pour sélectionner zéro ou plusieurs choix)
<input type="submit">	Affiche un bouton d'envoi (pour soumettre le formulaire)
<input type="button">	Affiche un bouton cliquable

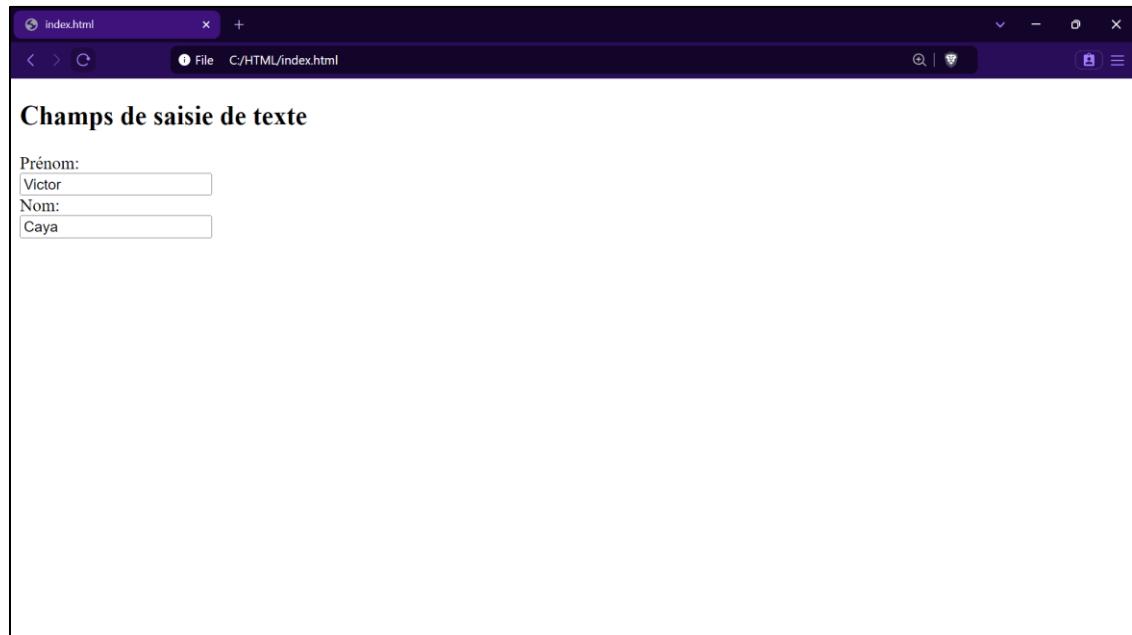
Champs de texte:

Le `<input type="text">` définit un champ de saisie sur une seule ligne pour la saisie de texte.

Exemple: Un formulaire avec des champs de saisie pour le texte:

Formulaires HTML

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Champs de saisie de texte</h2>
5
6     <form>
7       <label for="prenom">Prénom:</label><br />
8       <input type="text" id="prenom" name="prenom" value="Victor" /><br />
9       <label for="nom">Nom:</label><br />
10      <input type="text" id="nom" name="nom" value="Caya" /><br /><br />
11    </form>
12  </body>
13 </html>
```



Remarque: le formulaire lui-même n'est pas visible. Notez également que la largeur par défaut d'un champ de saisie est de 20 caractères.

Formulaires HTML

L'élément <label>:

La balise `<label>` définit une étiquette pour de nombreux éléments de formulaire.

L'élément `<label>` est utile pour les utilisateurs de lecteurs d'écran, car le lecteur d'écran lira à haute voix l'étiquette lorsque l'utilisateur se concentre sur l'élément d'entrée.

L'élément `<label>` aide également les utilisateurs qui ont des difficultés à cliquer sur de très petites zones (telles que des boutons radio ou des cases à cocher) - car lorsque l'utilisateur clique sur le texte dans l'élément `<label>`, il bascule le bouton radio/la case à cocher.

L'attribut `for` de la balise `<label>` doit être égal à l'attribut `id` de l'élément `<input>` pour les lier ensemble.

Boutons radio:

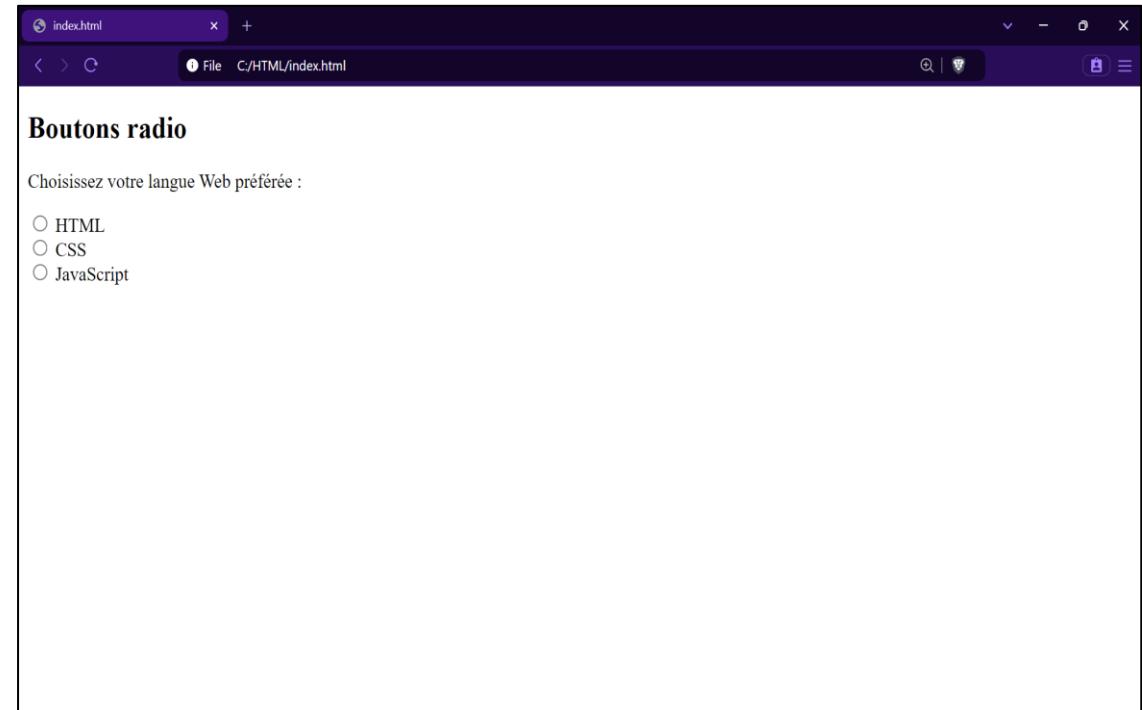
Le `<input type="radio">` définit un bouton radio.

Les boutons radio permettent à l'utilisateur de sélectionner un choix parmi un nombre limité.

Exemple: Un formulaire avec des boutons radio:

Formulaires HTML

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Boutons radio</h2>
5
6     <p>Choisissez votre langue Web préférée :</p>
7
8     <form>
9       <input type="radio" id="html" name="langue_preferee" value="HTML">
10      <label for="html">HTML</label><br>
11      <input type="radio" id="css" name="langue_preferee" value="CSS">
12      <label for="css">CSS</label><br>
13      <input type="radio" id="javascript" name="langue_preferee" value="JavaScript">
14      <label for="javascript">JavaScript</label>
15    </form>
16  </body>
17 </html>
```



Formulaires HTML

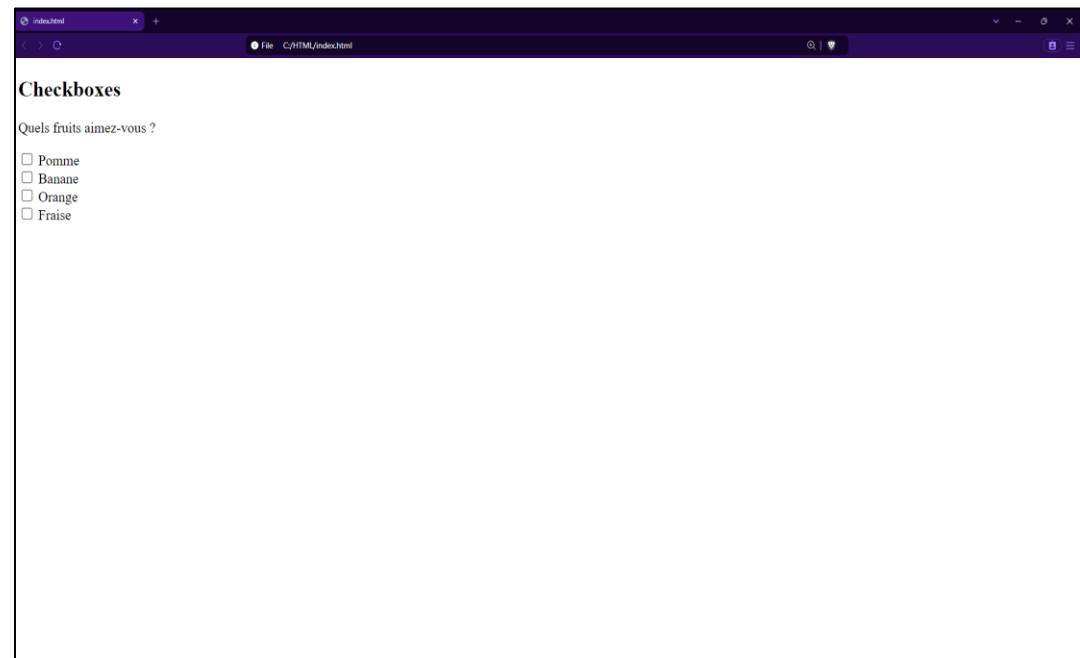
Checkboxes :

`<input type="checkbox">` définit une case à cocher.

Les cases à cocher permettent à un utilisateur de sélectionner ZÉRO ou PLUS d'options parmi un nombre limité de choix.

Exemple :

```
1  <!DOCTYPE html>
2  <html>
3    <body>
4      <h2>Checkboxes</h2>
5
6      <p>Quels fruits aimez-vous ?</p>
7
8      <form>
9        <input type="checkbox" id="pomme" name="fruit" value="pomme" />
10       <label for="pomme">Pomme</label><br />
11       <input type="checkbox" id="banane" name="fruit" value="banane" />
12       <label for="banane">Banane</label><br />
13       <input type="checkbox" id="orange" name="fruit" value="orange" />
14       <label for="orange">Orange</label><br />
15       <input type="checkbox" id="fraise" name="fruit" value="fraise" />
16       <label for="fraise">Fraise</label><br />
17     </form>
18   </body>
19 </html>
```



Formulaires HTML

Le bouton submit:

Le `<input type="submit">` définit un bouton permettant de soumettre les données du formulaire à un gestionnaire de formulaire.

Le gestionnaire de formulaire est généralement un fichier sur le serveur contenant un script pour le traitement des données d'entrée.

Le gestionnaire de formulaire est spécifié dans l'attribut `action` du formulaire.

Exemple: Un formulaire avec un bouton submit:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Formulaires HTML</h2>
5
6     <form action="action_page.php">
7       <label for="prenom">Prénom:</label><br />
8       <input type="text" name="prenom" id="prenom" value="Victor" /><br />
9       <label for="nom">Nom:</label><br />
10      <input type="text" name="nom" id="nom" value="Caya" /><br /><br />
11      <input type="submit" value="Soumettre" />
12    </form>
13
14    <p>
15      Si vous cliquez sur le bouton "Soumettre", les données du formulaire
16      seront envoyées vers une page appelée "action_page.php" et la page affichera les données.
17    </p>
18  </body>
19 </html>
```



Formulaires HTML

L'attribut `name` pour `<input>`:

Notez que chaque input doit avoir un attribut `name` pour être envoyé.

Si l'attribut `name` est omis, la valeur du input ne sera pas envoyée du tout.

Formulaires HTML

Attributs du formulaire HTML:

L'attribut action:

L'attribut **action** définit l'action à effectuer lors de l'envoi du formulaire.

En général, les données du formulaire sont envoyées vers un fichier sur le serveur lorsque l'utilisateur clique sur le bouton d'envoi.

Dans l'exemple ci-dessous, les données du formulaire sont envoyées vers un fichier appelé "action_page.php". Ce fichier contient un script côté serveur qui gère les données du formulaire :

Exemple: lors de la soumission, envoyer les données du formulaire à "action_page.php":

Formulaires HTML

```
1  <!DOCTYPE html>
2  <html>
3    <body>
4      <h2>Formulaires HTML</h2>
5
6      <form action="action_page.php">
7        <label for="prenom">Prénom:</label><br />
8        <input type="text" name="prenom" id="prenom" value="Victor" /><br />
9        <label for="nom">Nom:</label><br />
10       <input type="text" name="nom" id="nom" value="Caya" /><br /><br />
11       <input type="submit" value="Soumettre" />
12     </form>
13
14     <p>
15       Si vous cliquez sur le bouton "Soumettre", les données du formulaire
16       seront envoyées vers une page appelée "action_page.php" et la page affichera les données.
17     </p>
18   </body>
19 </html>
```



Astuce: si l'attribut `action` est omis, l'action est définie sur la page actuelle.

Formulaires HTML

L'attribut target:

L'attribut **target** spécifie où afficher la réponse reçue après l'envoi du formulaire.

L'attribut **target** peut avoir l'une des valeurs suivantes :

Valeur	Description
_blank	La réponse s'affiche dans une nouvelle fenêtre ou un nouvel onglet
_self	La réponse est affichée dans la fenêtre actuelle
_parent	La réponse est affichée dans le cadre parent
_top	La réponse est affichée dans le corps entier de la fenêtre
framename	The response is displayed in a named iframe

La valeur par défaut est **_self**, ce qui signifie que la réponse s'ouvrira dans la fenêtre actuelle.

Formulaires HTML

L'attribut `method`:

L'attribut `method` spécifie la méthode HTTP à utiliser lors de l'envoi des données du formulaire.

Les données du formulaire peuvent être envoyées sous forme de variables URL (avec `method="get"`) ou sous forme de transaction HTTP post (avec `method="post"`).

La méthode HTTP par défaut lors de l'envoi des données du formulaire est GET.

Remarques sur GET:

- Ajoute les données du formulaire à l'URL, dans des paires nom/valeur
- N'utilisez JAMAIS GET pour envoyer des données sensibles ! (les données du formulaire envoyées sont visibles dans l'URL !)
- La longueur d'une URL est limitée (2048 caractères)
- Utile pour les soumissions de formulaires lorsqu'un utilisateur souhaite ajouter le résultat à ses favoris
- GET est utile pour les données non sécurisées, comme les chaînes de requête dans Google

Remarques sur POST:

- Ajoute les données du formulaire à l'intérieur du corps de la requête HTTP (les données du formulaire envoyées ne sont pas affichées dans l'URL)
- POST n'a aucune limite de taille et peut être utilisé pour envoyer de grandes quantités de données.
- Les soumissions de formulaires avec POST ne peuvent pas être ajoutées à leurs favoris

Formulaires HTML

Astuce: utilisez toujours POST si les données du formulaire contiennent des informations sensibles ou personnelles !

L'attribut `autocomplete`:

L'attribut `autocomplete` spécifie si la saisie semi-automatique doit être activée ou désactivée pour un formulaire.

Lorsque la saisie semi-automatique est activée, le navigateur complète automatiquement les valeurs en fonction des valeurs saisies précédemment par l'utilisateur.

L'attribut `novalidate`:

L'attribut `novalidate` est un attribut booléen.

lorsqu'il est présent, il précise que les données du formulaire (entrée) ne doivent pas être validées lors de leur soumission.

Formulaires HTML

Éléments de formulaire HTML:

Les éléments HTML <form>:

L'élément HTML <form> peut contenir un ou plusieurs des éléments de formulaire suivants:

- <input>
- <label>
- <select>
- <textarea>
- <button>
- <fieldset>
- <legend>
- <datalist>
- <output>
- <option>
- <optgroup>

Formulaires HTML

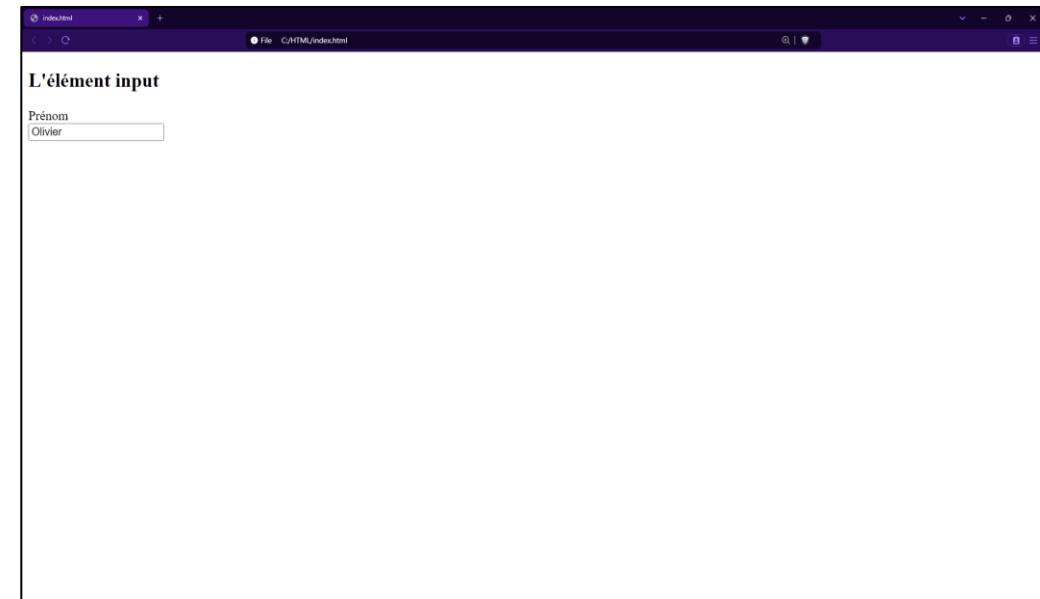
L'élément <input>:

L'un des éléments de formulaire les plus utilisés est l'élément `<input>`.

L'élément `<input>` peut être affiché de plusieurs manières, selon l'attribut `type`.

Exemple:

```
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <h2>L'élément input</h2>
5
6          <form>
7              <label for="prenom">Prénom</label><br />
8              <input type="text" id="prenom" name="prenom" value="Olivier" /><br /><br />
9          </form>
10         </body>
11     </html>
```



Formulaires HTML

L'élément <label>:

La balise `<label>` définit une étiquette pour de nombreux éléments de formulaire.

L'élément `<label>` est utile pour les utilisateurs de lecteurs d'écran, car le lecteur d'écran lira à haute voix l'étiquette lorsque l'utilisateur se concentre sur l'élément d'entrée.

L'élément `<label>` aide également les utilisateurs qui ont des difficultés à cliquer sur de très petites zones (telles que des boutons radio ou des cases à cocher) - car lorsque l'utilisateur clique sur le texte dans l'élément `<label>`, il bascule le bouton radio/la case à cocher.

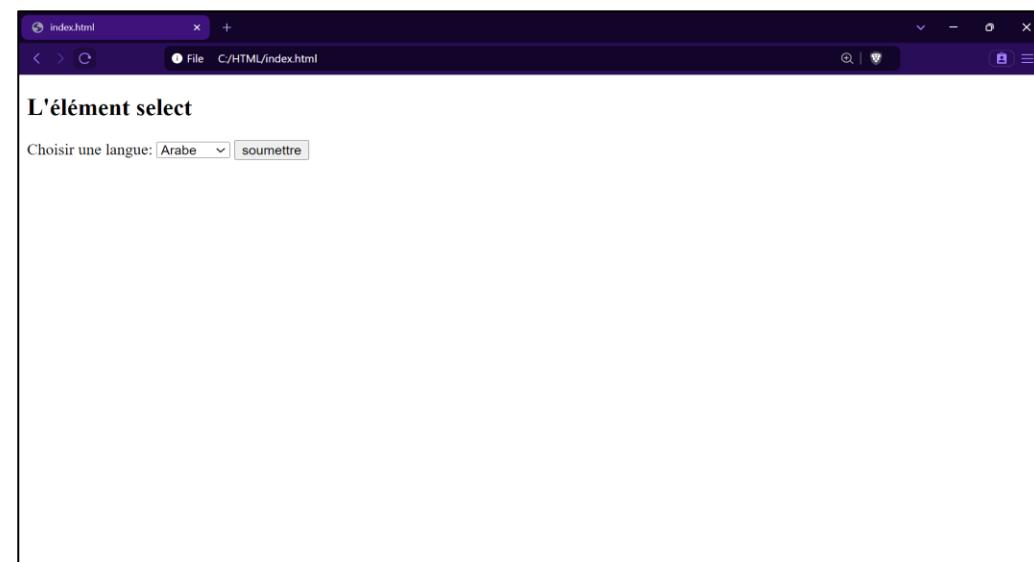
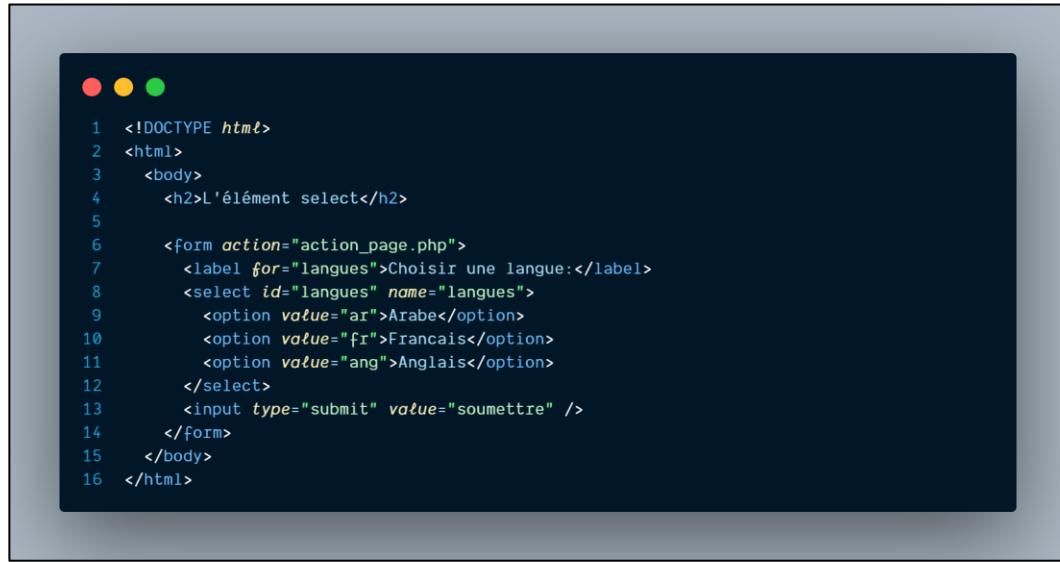
L'attribut `for` de la balise `<label>` doit être égal à l'attribut `id` de l'élément `<input>` pour les lier ensemble.

L'élément <select>:

L'élément `<select>` définit une liste déroulante.

Exemple:

Formulaires HTML



The left image shows a code editor with an HTML file named index.html. The code defines a dropdown menu with three options: Arabic, French, and English. The first option, Arabic, is selected by default. The right image shows a browser window displaying the same HTML code. A dropdown menu titled "L'élément select" is shown, with "Arabe" selected and a "soumettre" button below it.

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>L'élément select</h2>
5
6     <form action="action_page.php">
7       <label for="langues">Choisir une langue:</label>
8       <select id="langues" name="langues">
9         <option value="ar">Arabe</option>
10        <option value="fr">Français</option>
11        <option value="ang">Anglais</option>
12      </select>
13      <input type="submit" value="soumettre" />
14    </form>
15  </body>
16 </html>
```

L'élément **<option>** définit une option qui peut être sélectionnée.

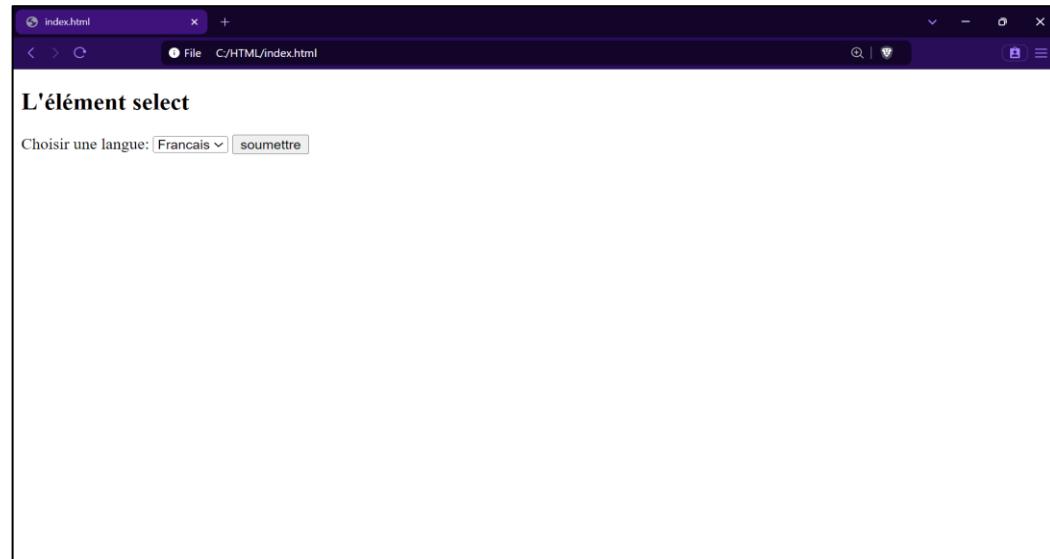
Par défaut, le premier élément de la liste déroulante est sélectionné.

Pour définir une option présélectionnée, ajoutez l'attribut **selected** à l'option:

Exemple:

Formulaires HTML

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>L'élément select</h2>
5
6     <form action="action_page.php">
7       <label for="langues">Choisir une langue:</label>
8       <select id="langues" name="langues">
9         <option value="ar">Arabe</option>
10        <option value="fr" selected>Français</option>
11        <option value="ang">Anglais</option>
12      </select>
13      <input type="submit" value="soumettre" />
14    </form>
15  </body>
16 </html>
```



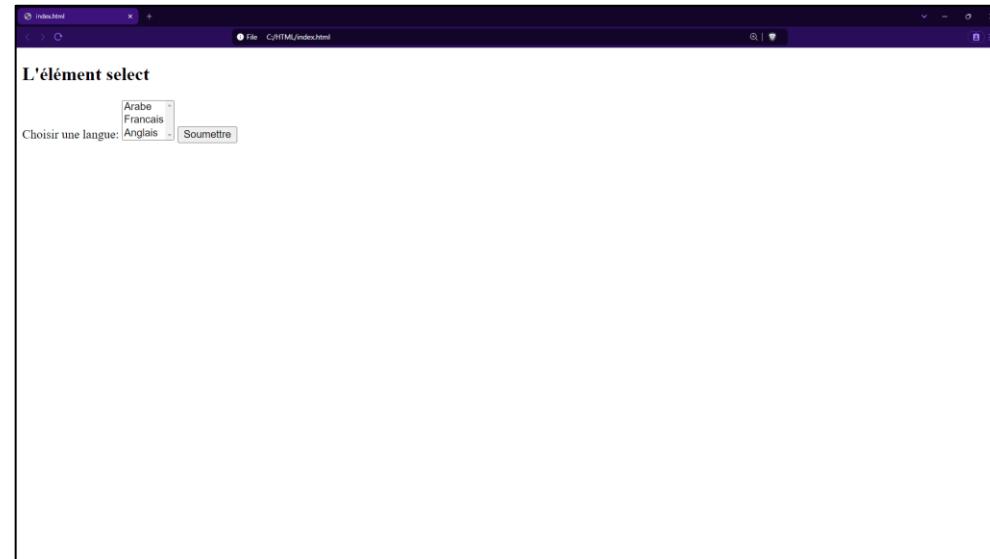
Valeurs visibles:

Utilisez l'attribut **size** pour spécifier le nombre de valeurs visibles.

Exemple:

Formulaires HTML

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>L'élément select</h2>
5
6     <form action="action_page.php">
7       <label for="langues">Choisir une langue:</label>
8       <select id="langues" name="langues" size="3">
9         <option value="ar">Arabe</option>
10        <option value="fr">Francais</option>
11        <option value="ang">Anglais</option>
12      </select>
13      <input type="submit" value="soumettre" />
14    </form>
15  </body>
16 </html>
```

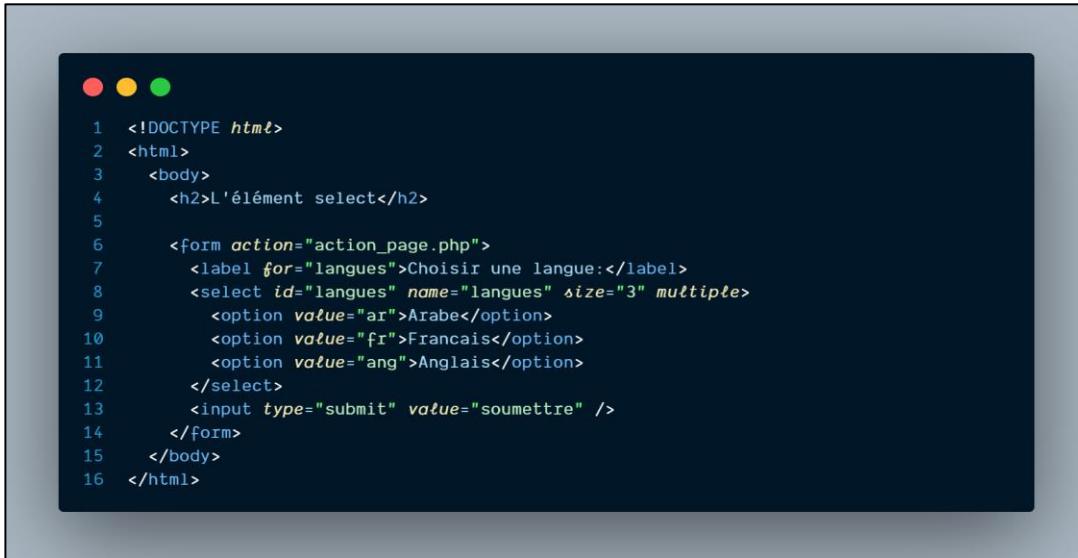


Autoriser les sélections multiples:

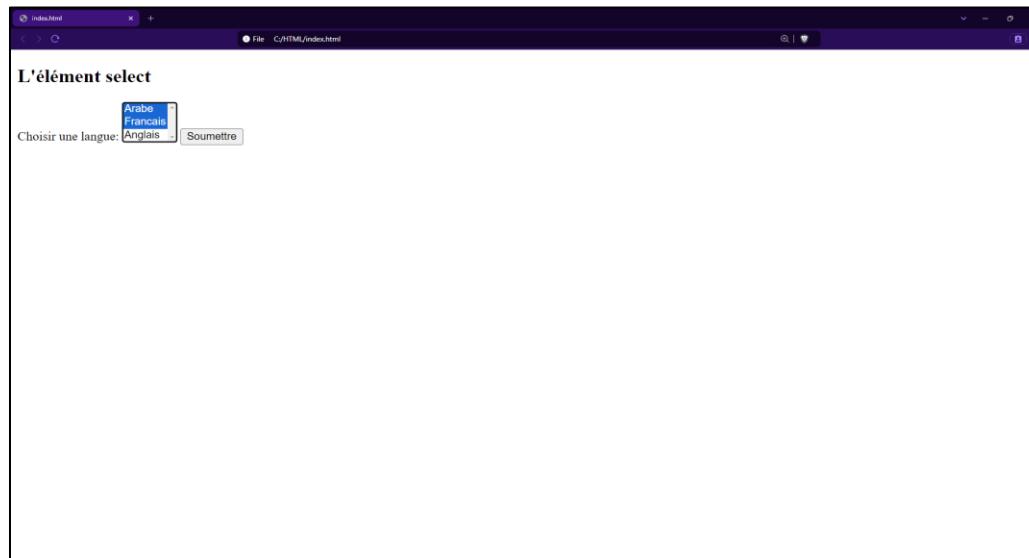
Utilisez l'attribut **multiple** pour permettre à l'utilisateur de sélectionner plusieurs valeurs:

Exemple:

Formulaires HTML



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>L'élément select</h2>
5
6     <form action="action_page.php">
7       <label for="langues">Choisir une langue:</label>
8       <select id="langues" name="langues" size="3" multiple>
9         <option value="ar">Arabe</option>
10        <option value="fr">Francais</option>
11        <option value="ang">Anglais</option>
12      </select>
13      <input type="submit" value="soumettre" />
14    </form>
15  </body>
16 </html>
```

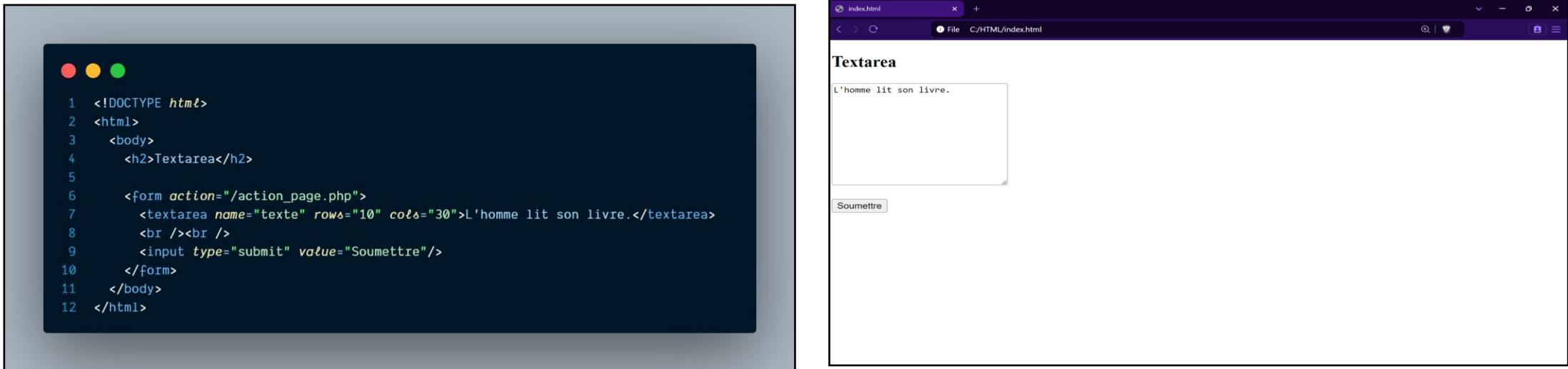


L'élément <textarea>:

L'élément **<textarea>** définit un champ de saisie multiligne (une zone de texte):

Exemple:

Formulaires HTML



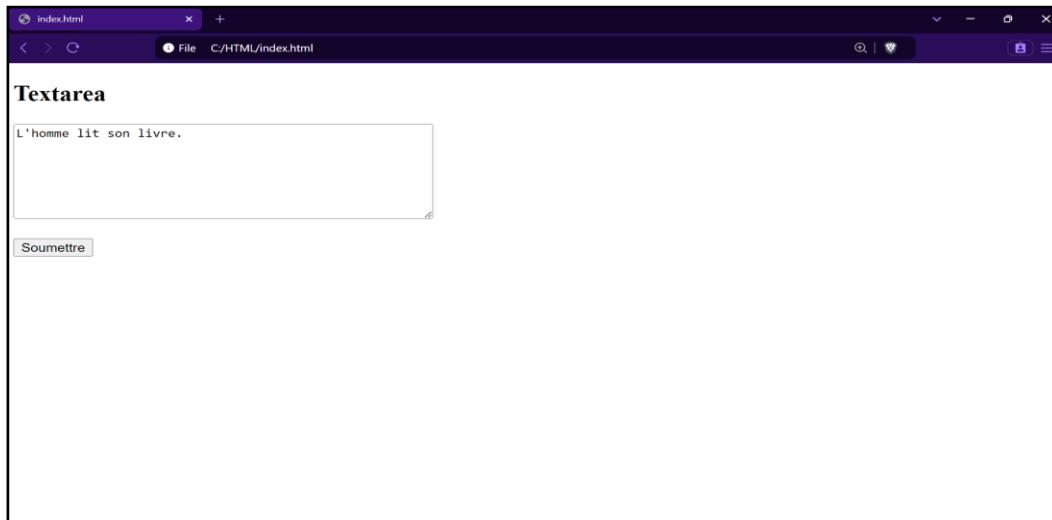
L'attribut **rows** spécifie le nombre de lignes visibles dans une zone de texte.
L'attribut **cols** spécifie la largeur visible d'une zone de texte.

Vous pouvez également définir la taille de la zone de texte en utilisant CSS:

Exemple:

Formulaires HTML

```
● ● ●
1  <!DOCTYPE html>
2  <html>
3    <body>
4      <h2>Textarea</h2>
5
6      <form action="/action_page.php">
7        <textarea name="texte" rows="10" cols="30" style="width: 400px; height: 100px;">L'homme lit son livre.</textarea>
8        <br /><br />
9        <input type="submit" value="Soumettre"/>
10
11      </form>
12    </body>
13  </html>
```

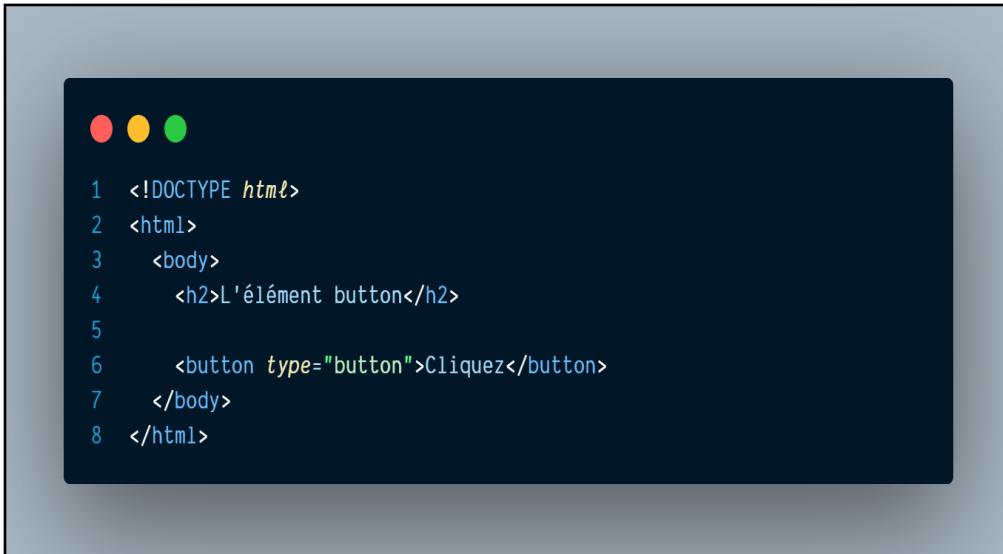


L'élément **<button>**:

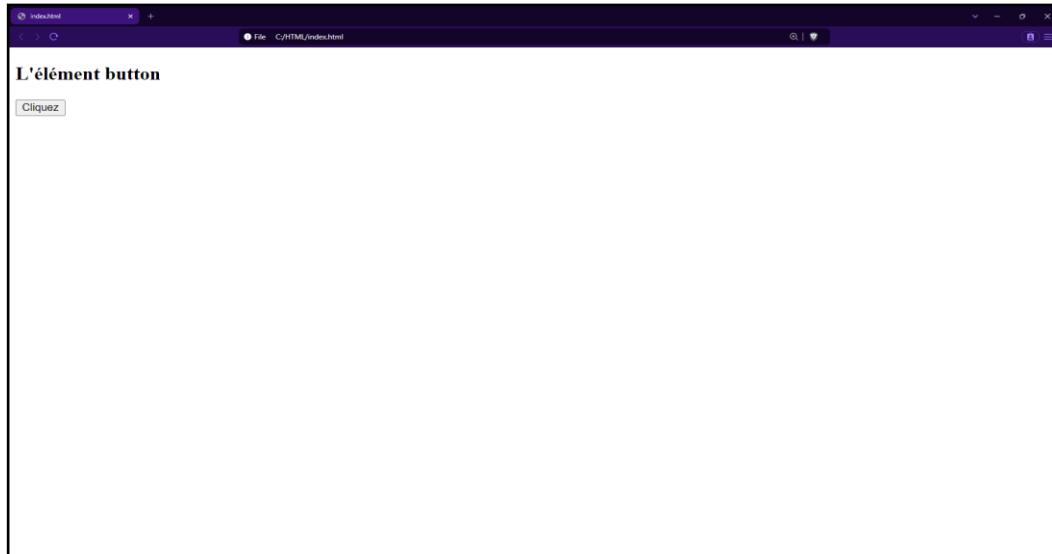
L'élément **<button>** définit un bouton cliquable:

Exemple:

Formulaires HTML



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>L'élément button</h2>
5
6     <button type="button">Cliquez</button>
7   </body>
8 </html>
```



Remarque: spécifiez toujours l'attribut **type** pour l'élément bouton. Différents navigateurs peuvent utiliser différents types par défaut pour l'élément bouton.

Formulaires HTML

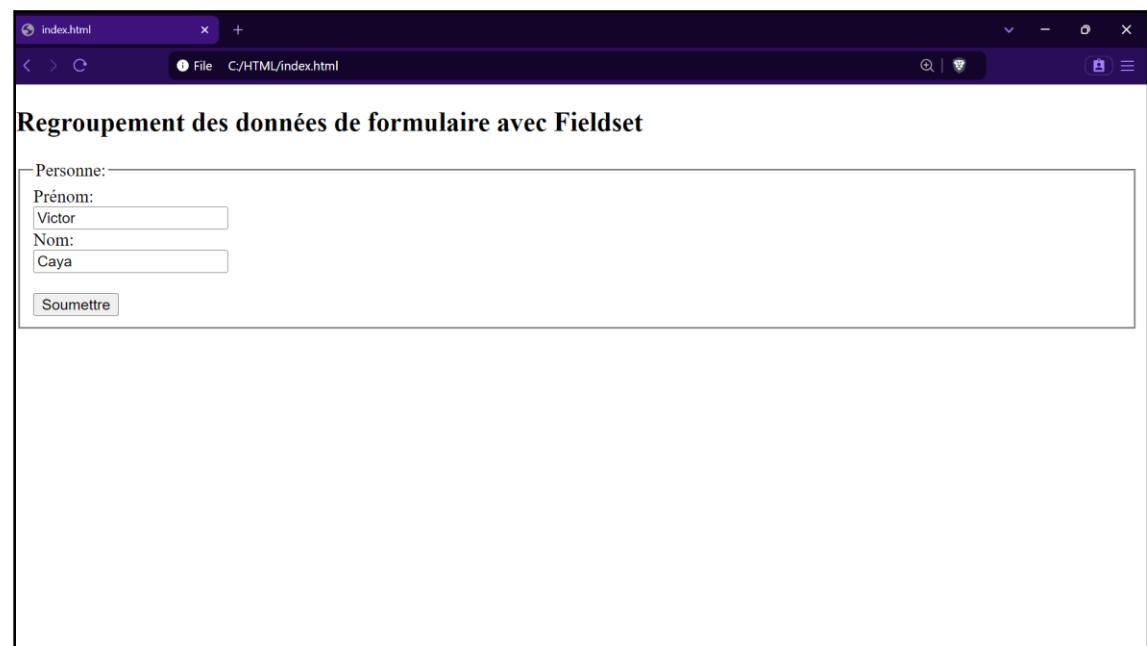
Les éléments <fieldset> et <legend>:

L'élément <fieldset> est utilisé pour regrouper des données liées dans un formulaire.

L'élément <legend> définit une légende pour l'élément <fieldset>.

Exemple:

```
1  <!DOCTYPE html>
2  <html>
3    <body>
4      <h2>Regroupement des données de formulaire avec Fieldset</h2>
5
6      <form action="action_page.php">
7        <fieldset>
8          <legend>Personne:</legend>
9          <label for="prenom">Prénom:</label><br />
10         <input type="text" id="prenom" name="prenom" value="Victor" /><br />
11         <label for="nom">Nom:</label><br />
12         <input type="text" id="nom" name="nom" value="Caya" /><br /><br />
13         <input type="submit" value="Soumettre" />
14     </fieldset>
15   </form>
16 </body>
17 </html>
```



Formulaires HTML

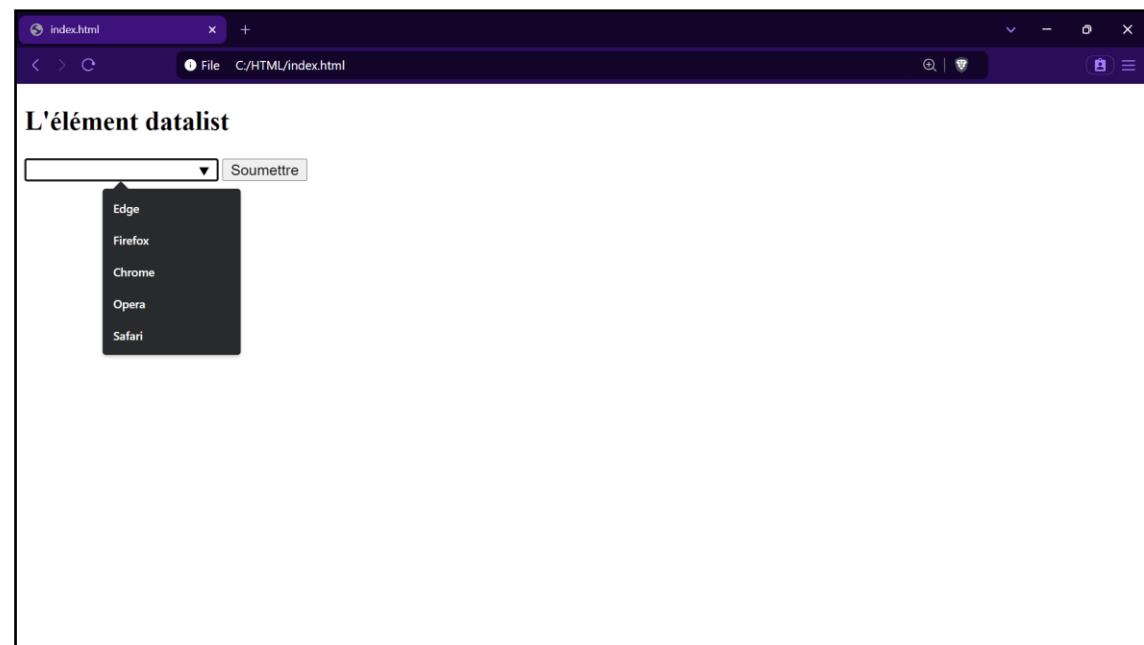
L'élément <datalist>:

L'élément `<datalist>` spécifie une liste d'options prédéfinies pour un élément `<input>`. Les utilisateurs verront une liste déroulante des options prédéfinies lorsqu'ils saisissent des données.

L'attribut `list` de l'élément `<input>` doit faire référence à l'attribut `id` de l'élément `<datalist>`.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>L'élément datalist</h2>
5
6     <form action="action_page.php">
7       <input list="navigateurs" name="navigateurs" />
8       <datalist id="navigateurs">
9         <option value="Edge"></option>
10        <option value="Firefox"></option>
11        <option value="Chrome"></option>
12        <option value="Opera"></option>
13        <option value="Safari"></option>
14      </datalist>
15      <input type="submit" value="Soumettre" />
16    </form>
17  </body>
18 </html>
```



Formulaires HTML

L'élément **<output>**:

L'élément **<output>** représente le résultat d'un calcul.

Formulaires HTML

Les types des HTML inputs:

Les types des HTML inputs:

Voici les différents types d'entrée que vous pouvez utiliser en 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">

Formulaires HTML

- <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">

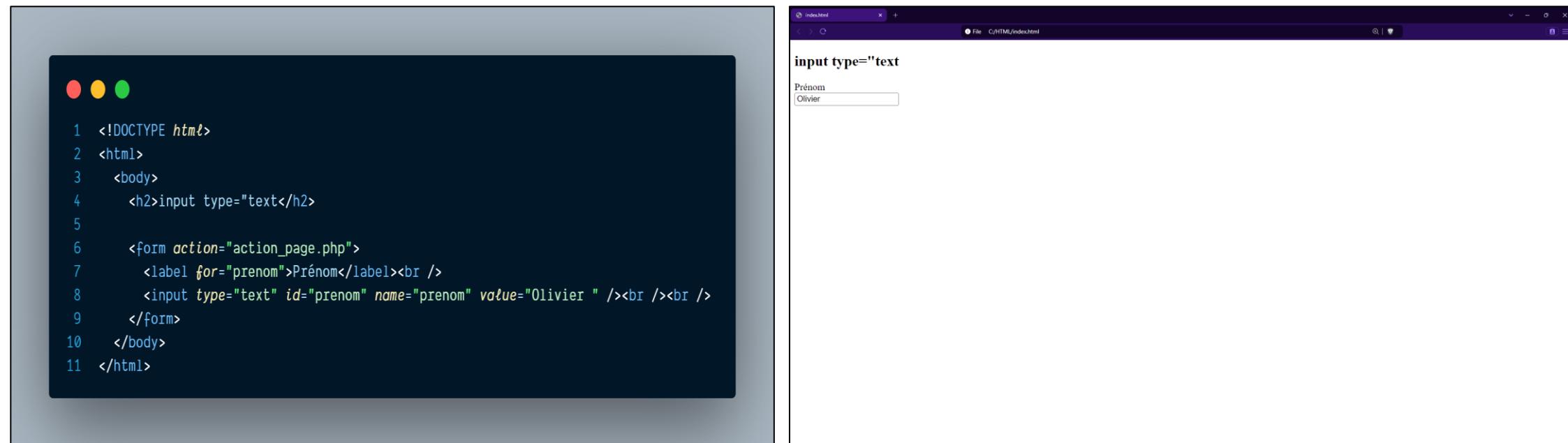
Astuce: la valeur par défaut de l'attribut **type** est "texte".

Formulaires HTML

Input type text:

<input type="text"> définit un champ de saisie de texte sur une seule ligne:

Exemple:



The image shows a split-screen view. On the left, a code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>input type="text"</h2>
5
6     <form action="action_page.php">
7       <label for="prenom">Prénom</label><br />
8       <input type="text" id="prenom" name="prenom" value="Olivier " /><br /><br />
9     </form>
10   </body>
11 </html>
```

On the right, a web browser window titled "index.html" shows the resulting page. It features a heading "input type="text"" and a text input field containing the value "Olivier".

Remarque: le formulaire lui-même n'est pas visible. Notez également que la largeur par défaut d'un champ de saisie est de 20 caractères.

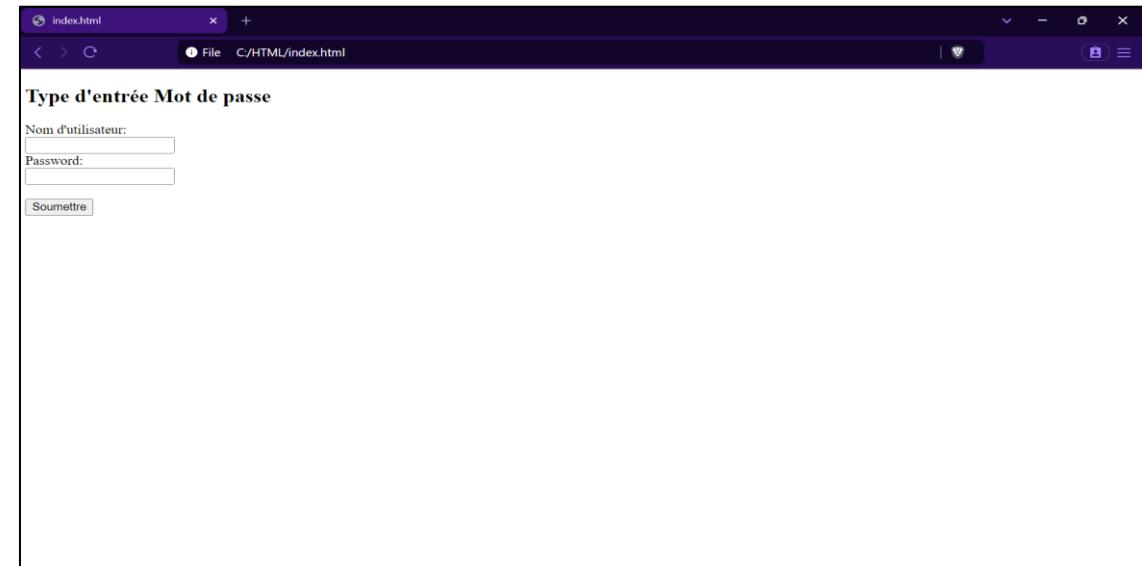
Formulaires HTML

Input type password:

<input type="password"> définit un champ de mot de passe:

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Type d'entrée Mot de passe</h2>
5
6     <form action="action_page.php">
7       <label for="nom_utilisateur">Nom d'utilisateur:</label><br />
8       <input type="text" id="nom_utilisateur" name="nom_utilisateur" /><br />
9       <label for="mtp">Password:</label><br />
10      <input type="password" id="mtp" name="mtp" /><br /><br />
11      <input type="submit" value="Soumettre" />
12    </form>
13  </body>
14 </html>
```



Les caractères d'un champ de mot de passe sont masqués (affichés sous forme d'astérisques ou de cercles).

Formulaires HTML

Input type Submit:

`<input type="submit">` définit un bouton pour soumettre des données de formulaire à un gestionnaire de formulaire.

Le gestionnaire de formulaire est généralement une page de serveur avec un script pour le traitement des données d'entrée.

Le gestionnaire de formulaire est spécifié dans l'attribut `action` du formulaire:

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Formulaires HTML</h2>
5
6     <form action="action_page.php">
7       <label for="prenom">Prénom:</label><br />
8       <input type="text" name="prenom" id="prenom" value="Victor" /><br />
9       <label for="nom">Nom:</label><br />
10      <input type="text" name="nom" id="nom" value="Caya" /><br /><br />
11      <input type="submit" value="Soumettre" />
12    </form>
13
14    <p>
15      Si vous cliquez sur le bouton "Soumettre", les données du formulaire
16      seront envoyées vers une page appelée "action_page.php" et la page affichera les données.
17    </p>
18  </body>
19 </html>
```



Si vous omettez l'attribut `value` du bouton d'envoi, le bouton obtiendra la valeur `submit`.

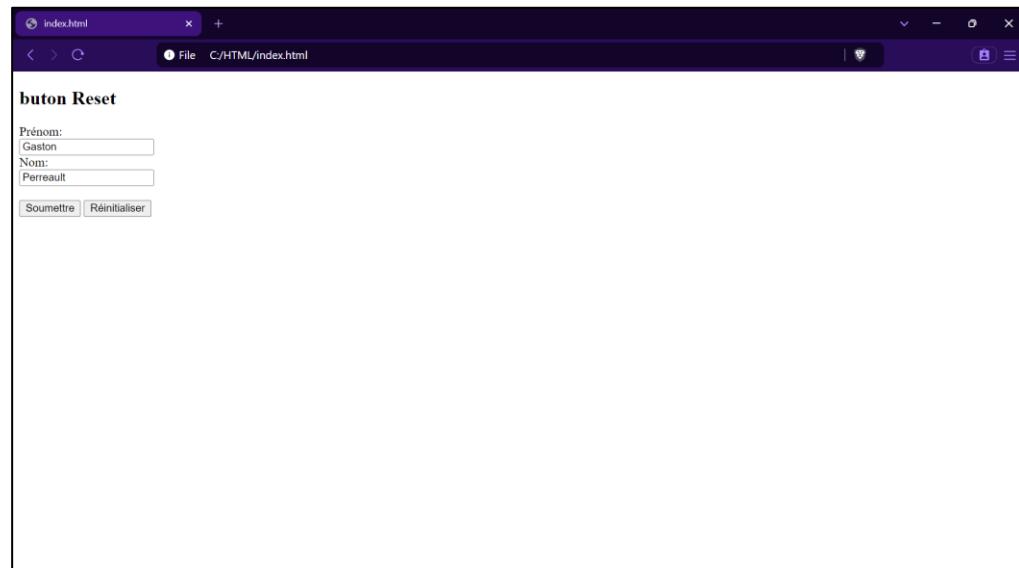
Formulaires HTML

Input type reset:

`<input type="reset">` définit un bouton de réinitialisation qui réinitialisera toutes les valeurs du formulaire à leurs valeurs par défaut:

Exemple:

```
1  <!DOCTYPE html>
2  <html>
3    <body>
4      <h2>button Reset</h2>
5
6      <form action="action_page.php">
7        <label for="prenom">Prénom:</label><br>
8        <input type="text" id="prenom" name="prenom" value="Gaston"><br>
9        <label for="nom">Nom:</label><br>
10       <input type="text" id="nom" name="nom" value="Perreault"><br><br>
11       <input type="submit" value="Soumettre" />
12       <input type="reset" value="Réinitialiser" />
13     </form>
14   </body>
15 </html>
```



Si vous modifiez les valeurs d'entrée et cliquez ensuite sur le bouton "Réinitialiser", les données du formulaire seront réinitialisées aux valeurs par défaut.

Formulaires HTML

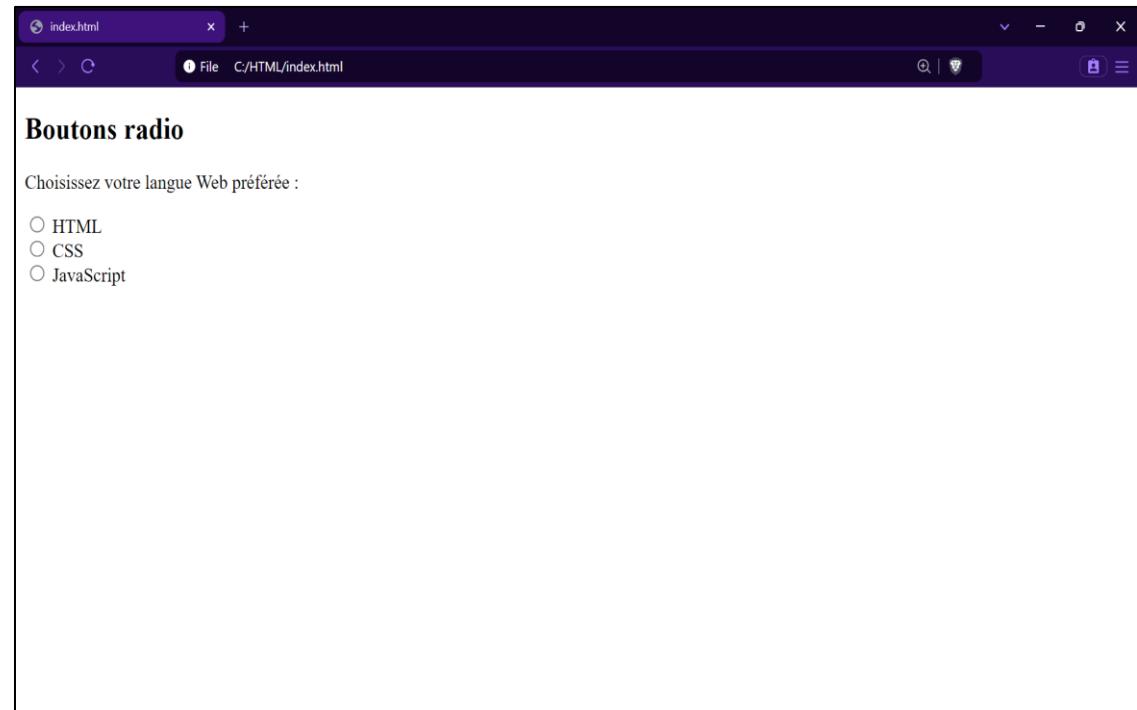
Input type radio:

`<input type="radio">` définit un bouton radio.

Les boutons radio permettent à l'utilisateur de sélectionner un seul choix parmi un nombre limité:

Exemple:

```
1  <!DOCTYPE html>
2  <html>
3    <body>
4      <h2>Boutons radio</h2>
5
6      <p>Choisissez votre langue Web préférée :</p>
7
8      <form>
9        <input type="radio" id="html" name="langue_preferee" value="HTML">
10       <label for="html">HTML</label><br>
11        <input type="radio" id="css" name="langue_preferee" value="CSS">
12       <label for="css">CSS</label><br>
13        <input type="radio" id="javascript" name="langue_preferee" value="JavaScript">
14       <label for="javascript">JavaScript</label>
15     </form>
16   </body>
17 </html>
```

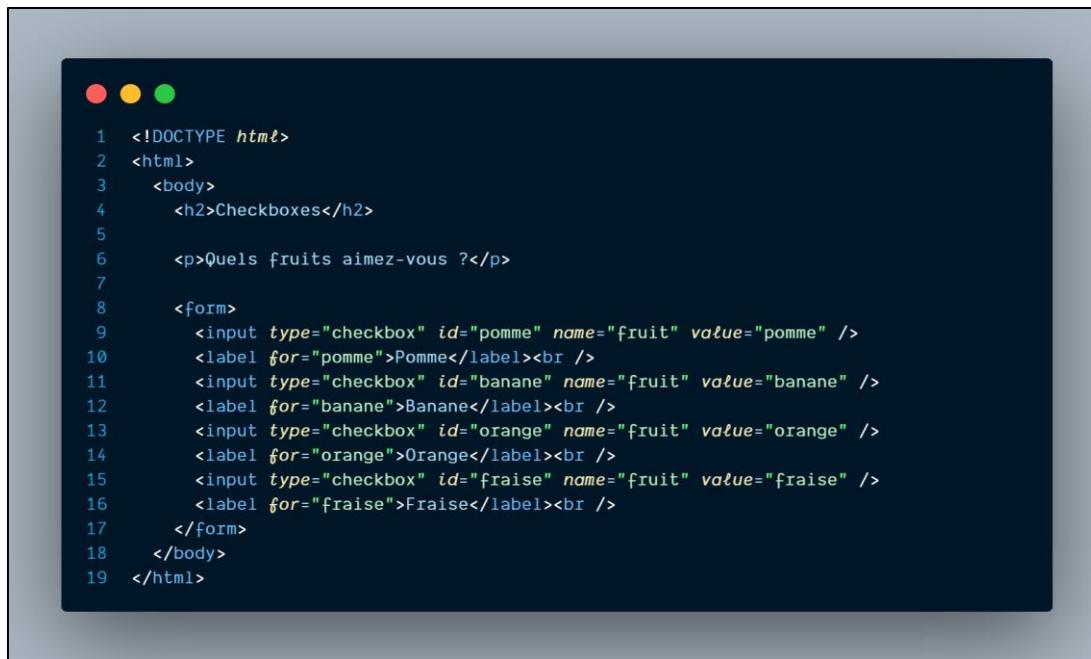


Formulaires HTML

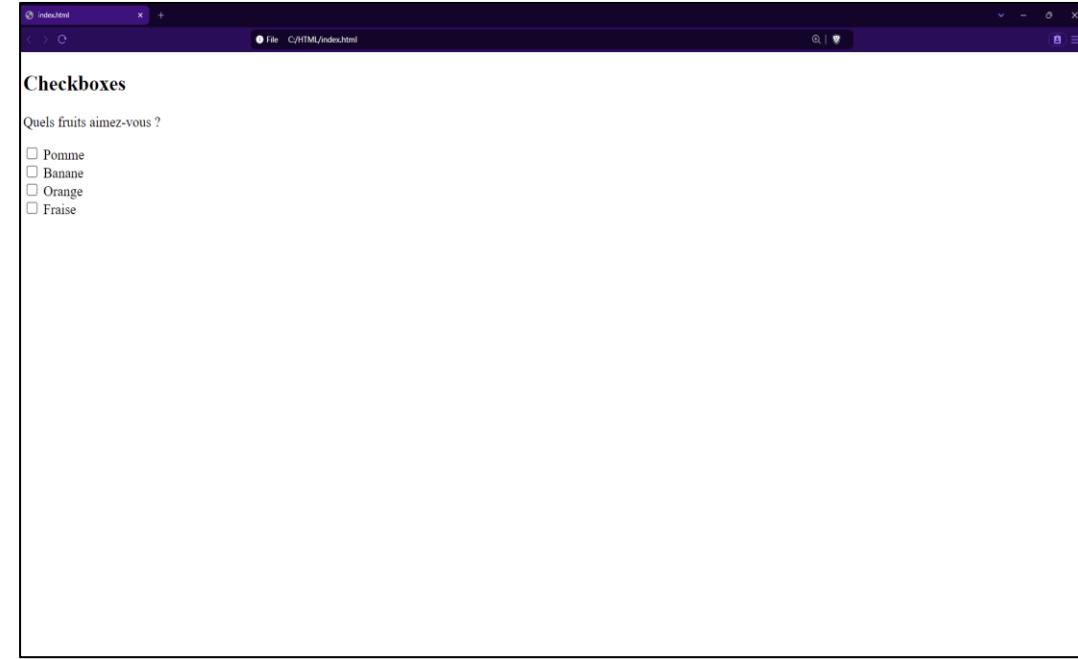
Checkboxes :

<input type="checkbox"> définit une case à cocher.

Les cases à cocher permettent à un utilisateur de sélectionner ZÉRO ou PLUS d'options parmi un nombre limité de choix.



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Checkboxes</h2>
5
6     <p>Quels fruits aimez-vous ?</p>
7
8     <form>
9       <input type="checkbox" id="pomme" name="fruit" value="pomme" />
10      <label for="pomme">Pomme</label><br />
11      <input type="checkbox" id="banane" name="fruit" value="banane" />
12      <label for="banane">Banane</label><br />
13      <input type="checkbox" id="orange" name="fruit" value="orange" />
14      <label for="orange">Orange</label><br />
15      <input type="checkbox" id="fraise" name="fruit" value="fraise" />
16      <label for="fraise">Fraise</label><br />
17    </form>
18  </body>
19 </html>
```



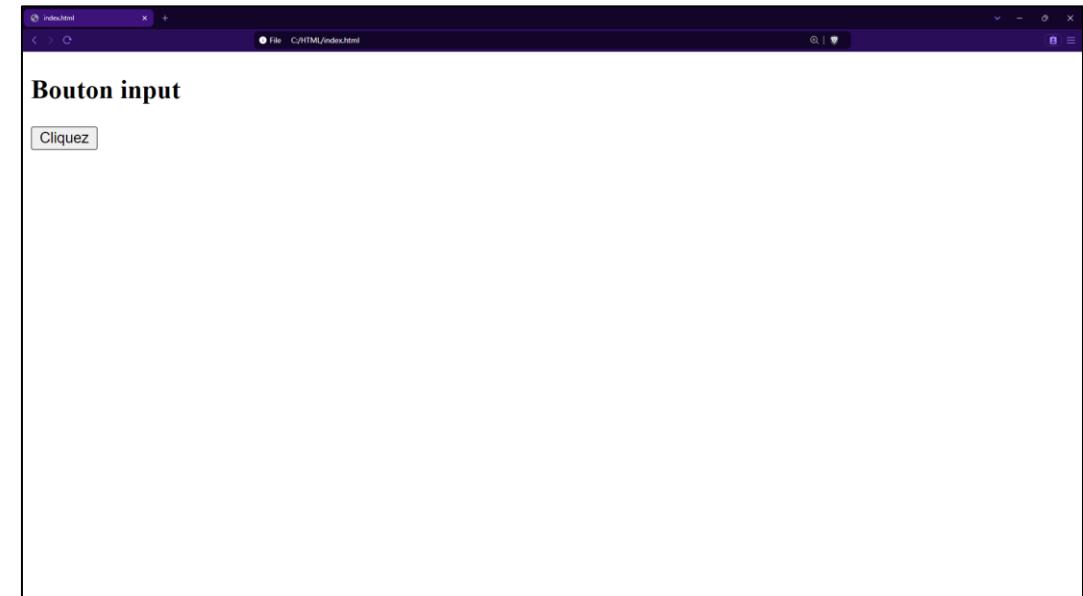
Formulaires HTML

Input type button:

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

Exemple:

```
1  <!DOCTYPE html>
2  <html>
3    <body>
4      <h2>Bouton input</h2>
5
6      <input type="button" value="Cliquez" />
7    </body>
8  </html>
```



Formulaires HTML

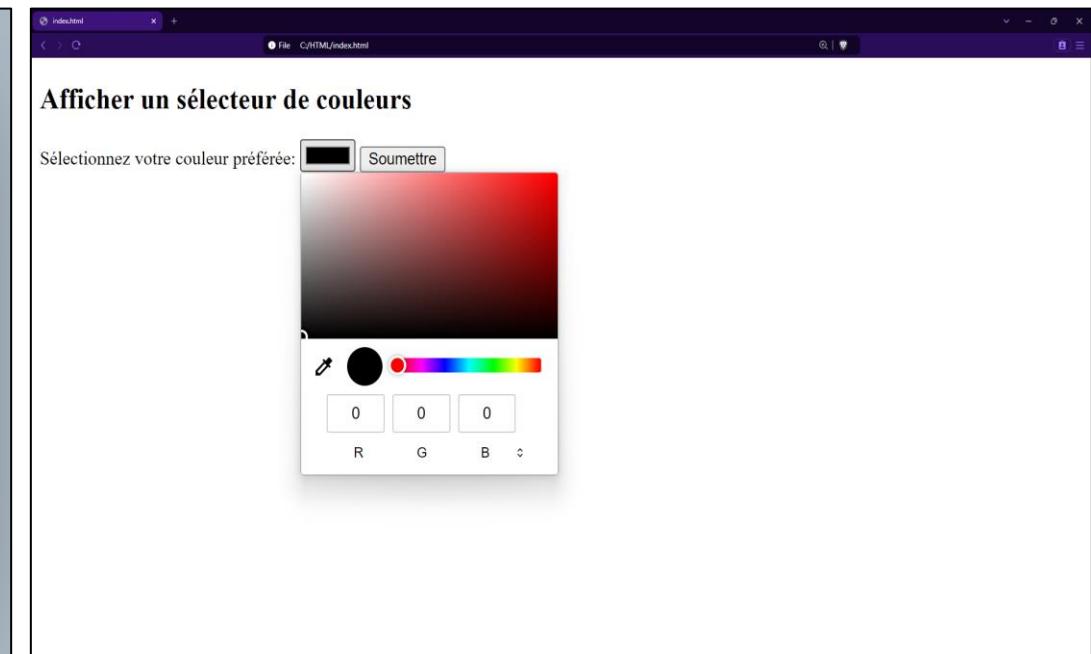
Input type color:

`<input type="color">` est utilisé pour les champs de saisie qui doivent contenir une couleur.

Selon la prise en charge du navigateur, un sélecteur de couleur peut apparaître dans le champ de saisie.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Afficher un sélecteur de couleurs</h2>
5
6     <form action="action_page.php">
7       <label for="couleur_preferee">Sélectionnez votre couleur préférée:</label>
8       <input type="color" id="couleur-preferee" name="couleur-preferee" />
9       <input type="submit" value="Soumettre" />
10    </form>
11  </body>
12 </html>
```



Formulaires HTML

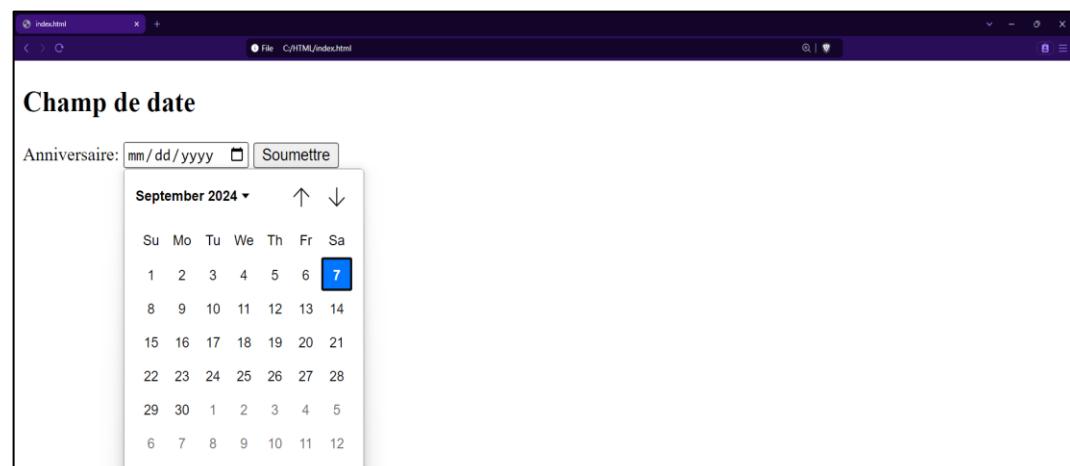
Input type date:

`<input type="date">` est utilisé pour les champs de saisie qui doivent contenir une date.

Selon la prise en charge du navigateur, un sélecteur de date peut apparaître dans le champ de saisie.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Champ de date</h2>
5
6     <form action="action_page.php">
7       <label for="anniversaire">Anniversaire:</label>
8       <input type="date" id="anniversaire" name="anniversaire" />
9       <input type="submit" value="Soumettre" />
10    </form>
11  </body>
12 </html>
```



Formulaires HTML

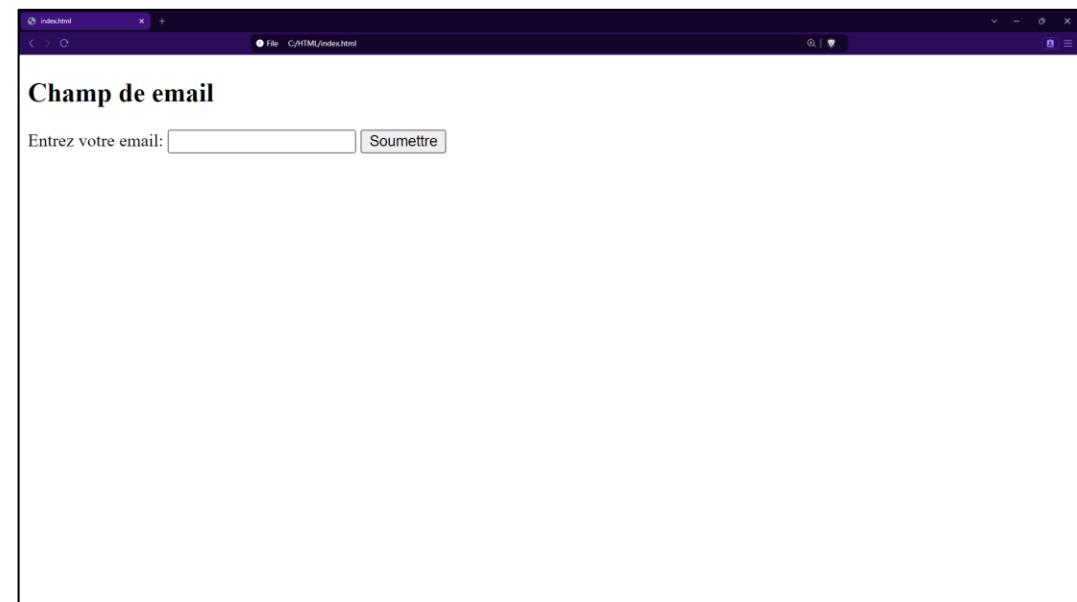
Input type email:

`<input type="email">` est utilisé pour les champs de saisie qui doivent contenir une adresse e-mail.

Selon la prise en charge du navigateur, l'adresse e-mail peut être automatiquement validée lors de sa soumission.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Champ de email</h2>
5
6     <form action="action_page.php">
7       <label for="email">Entrez votre email:</label>
8       <input type="email" id="email" name="email" />
9       <input type="submit" value="Soumettre" />
10    </form>
11  </body>
12 </html>
```



Formulaires HTML

Input type file:

`<input type="file">` définit un champ de sélection de fichier et un bouton "Parcourir" pour les téléchargements de fichiers.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Téléchargement de fichier</h1>
5
6     <form action="action_page.php">
7       <label for="fichier">Sélectionner un fichier:</label>
8       <input type="file" id="fichier" name="fichier" /><br /><br />
9       <input type="submit" value="Soumettre" />
10    </form>
11  </body>
12 </html>
```

The screenshot shows a web browser window titled "index.html" with the URL "C:/HTML/index.html". The page content is "Téléchargement de fichier". Below it is a form with a label "Sélectionner un fichier:" followed by an field. A "Choose File" button is next to the field, and the status is "No file chosen". Below the field is a "Soumettre" button. In the foreground, a Windows "Open" file dialog is displayed over the browser. The dialog shows a folder structure with various files and folders like "action", "cache", "config", "diffs", "Noms", "Places", "ash", "unicode", "3D Objects", "Contacts", "Desktop", "Documents", "Downloads", "Videos", "Favorites", "Links", "Music", "OneDrive", "Pictures", "Postman", "Searches", "ash_histo", "ash_config", "ash_config_wo", "ash_brok", "ash_brok_marks", "ash_repl", "ash_repl_history". The "File name:" field is empty. At the bottom right of the dialog are "Open" and "Cancel" buttons.

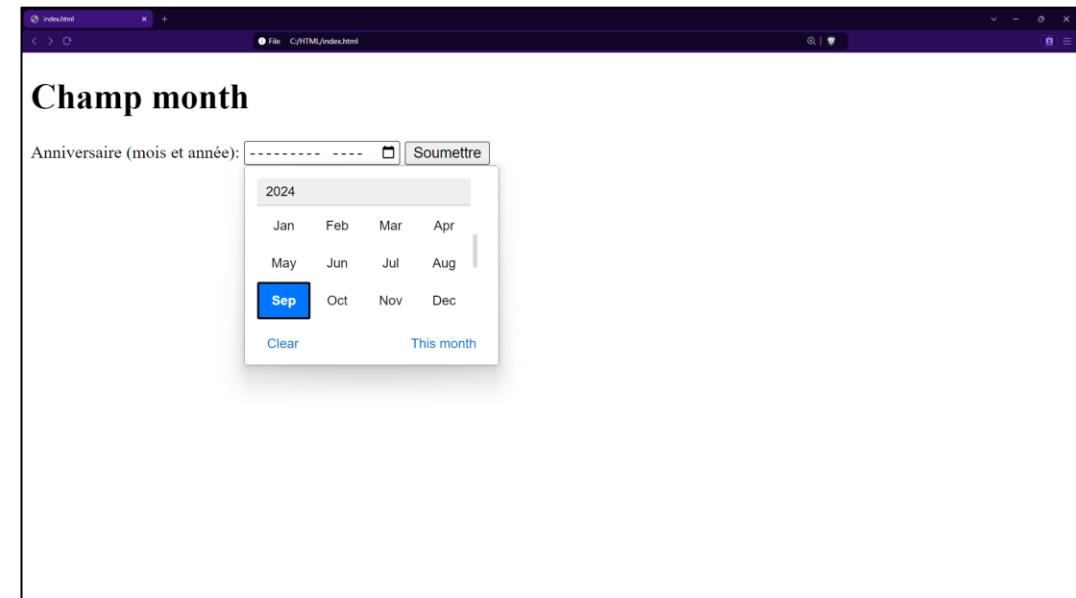
Formulaires HTML

Input type month:

`<input type="month">` permet à l'utilisateur de sélectionner un mois et une année. Selon la prise en charge du navigateur, un sélecteur de date peut apparaître dans le champ de saisie.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Champ month</h1>
5
6     <form action="action_page.php">
7       <label for="anniversaire-mois">Anniversaire (mois et année):</label>
8       <input type="month" id="anniversaire-mois" name="anniversaire-mois" />
9       <input type="submit" value="Soumettre" />
10    </form>
11  </body>
```



The screenshot shows a browser window titled "index.html" with the URL "C:/HTML/index.html". The page content is "Champ month". Below it is a form with a label "Anniversaire (mois et année)". The input field is a date picker. The year "2024" is selected. The month "Sep" is highlighted in blue. Other months visible are Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Oct, Nov, and Dec. At the bottom of the date picker are buttons for "Clear" and "This month".

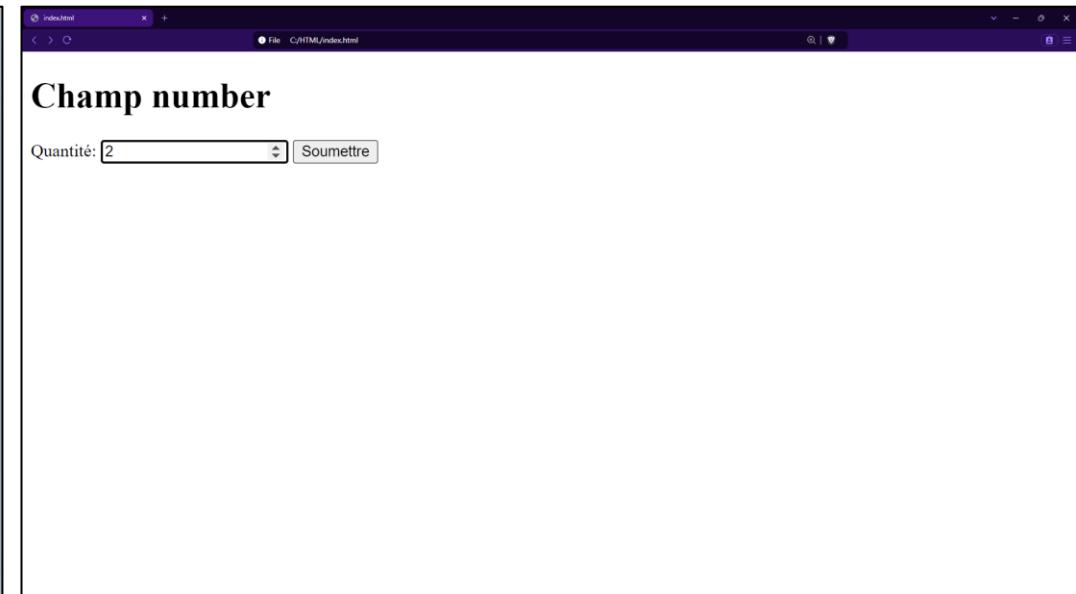
Formulaires HTML

Input type number:

<input type="number"> définit un champ de saisie numérique.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Champ number</h1>
5
6     <form action="action_page.php">
7       <label for="quantite">Quantité:</label>
8       <input type="number" id="quantite" name="quantite" />
9       <input type="submit" value="Soumettre" />
10    </form>
11  </body>
12 </html>
```



Formulaires HTML

Restrictions des inputs:

Voici une liste de certaines restrictions des inputs courantes:

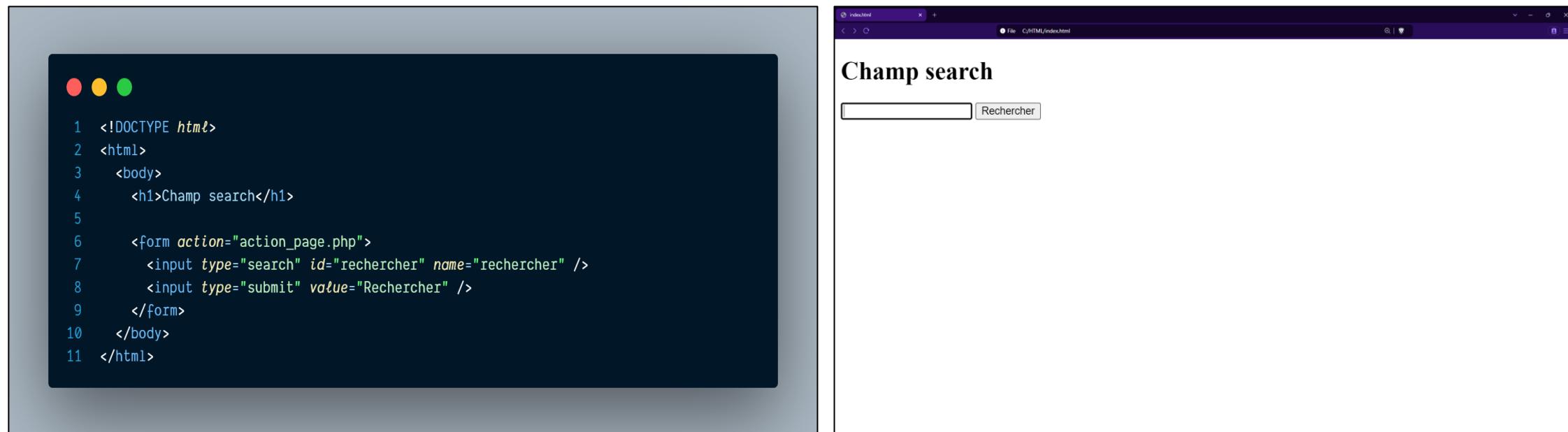
attribut	Description
checked	Spécifie qu'un champ de saisie doit être présélectionné lors du chargement de la page (pour type="checkbox" ou type="radio")
disabled	Spécifie qu'un champ de saisie doit être désactivé
max	Spécifie la valeur maximale d'un champ de saisie
maxlength	Spécifie le nombre maximal de caractères pour un champ de saisie
min	Spécifie la valeur minimale d'un champ de saisie
pattern	Spécifie une expression régulière pour vérifier la valeur d'entrée
readonly	Spécifie qu'un champ de saisie est en lecture seule (ne peut pas être modifié)
required	Indique qu'un champ de saisie est obligatoire (doit être rempli)
size	Spécifie la largeur (en caractères) d'un champ de saisie
step	Spécifie les intervalles de nombres légaux pour un champ de saisie
value	Spécifie la valeur par défaut d'un champ de saisie

Formulaires HTML

Input type search:

`<input type="search">` est utilisé pour les champs de recherche (un champ de recherche se comporte comme un champ de texte normal).

Exemple:



The image shows a code editor window on the left and a browser window on the right. The code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Champ search</h1>
5
6     <form action="action_page.php">
7       <input type="search" id="rechercher" name="rechercher" />
8       <input type="submit" value="Rechercher" />
9     </form>
10   </body>
11 </html>
```

The browser window shows the rendered page with the title "Champ search". It features a single input field labeled "Rechercher" and a "Rechercher" button.

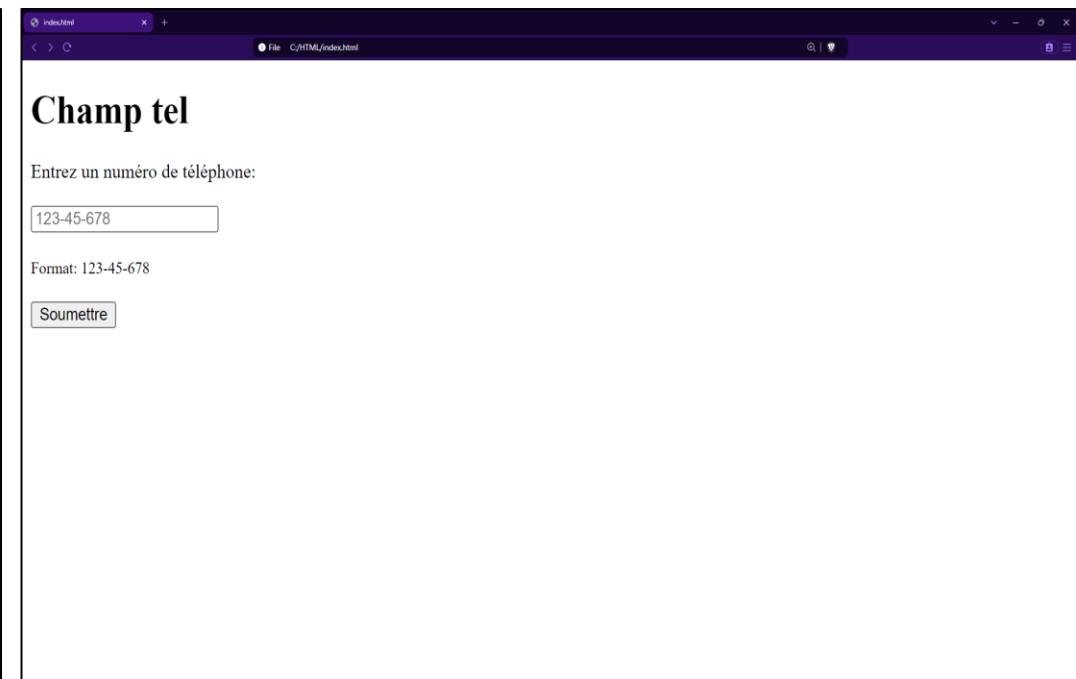
Formulaires HTML

Input type tel:

`<input type="tel">` est utilisé pour les champs de saisie qui doivent contenir un numéro de téléphone.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Champ tel</h1>
5
6     <form action="action_page.php">
7       <label for="telephone">Entrez un numéro de téléphone:</label><br /><br />
8       <input type="tel" id="telephone" name="telephone" placeholder="123-45-678"
9         pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}" required /><br /><br />
10      <small>Format: 123-45-678</small><br /><br />
11      <input type="submit" value="Soumettre" />
12    </form>
13  </body>
14 </html>
```



Formulaires HTML

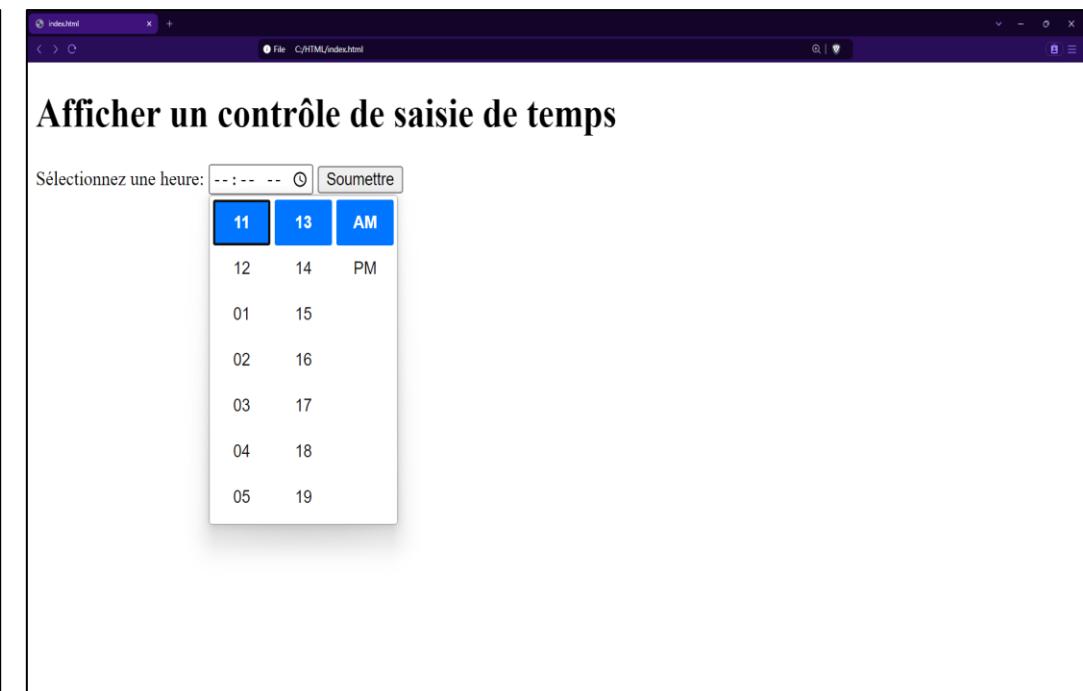
Input type time:

`<input type="time">` permet à l'utilisateur de sélectionner une heure.

Selon la prise en charge du navigateur, un sélecteur de temps peut apparaître dans le champ de saisie.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Afficher un contrôle de saisie de temps</h1>
5
6     <form action="action_page.php">
7       <label for="heure">Sélectionnez une heure:</label>
8       <input type="time" id="heure" name="heure" />
9       <input type="submit" value="Soumettre" />
10    </form>
11  </body>
12 </html>
```



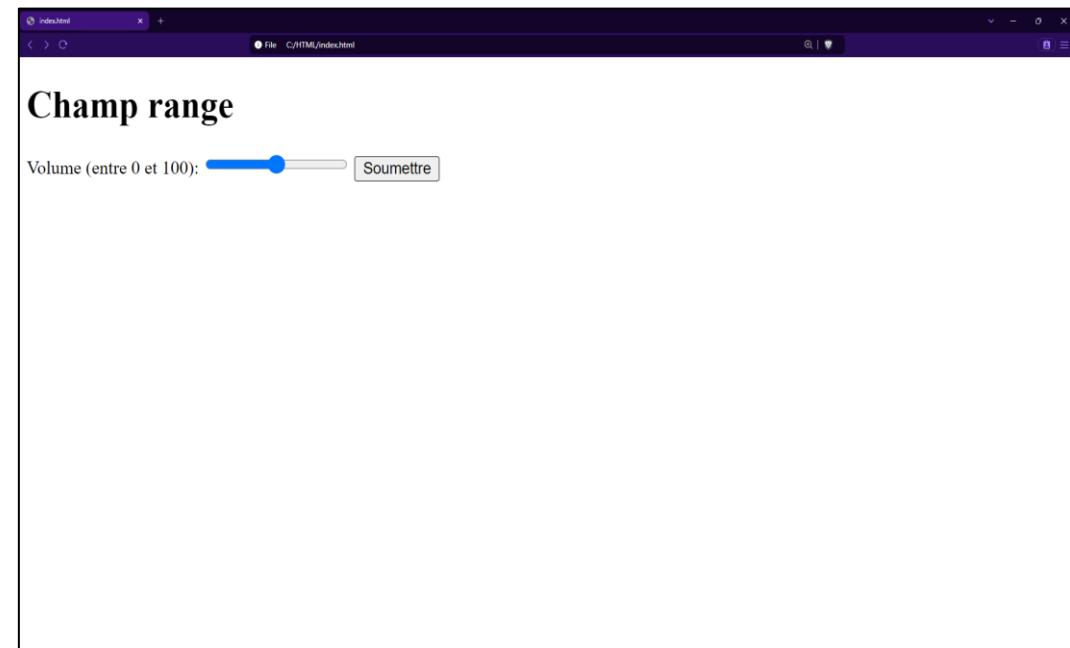
Formulaires HTML

Input type range:

`<input type="range">` définit un contrôle permettant de saisir un nombre dont la valeur exacte n'est pas importante (comme un contrôle de curseur). La plage par défaut est comprise entre 0 et 100. Cependant, vous pouvez définir des restrictions sur les nombres acceptés avec les attributs `min`, `max` et `step`:

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Champ range</h1>
5
6     <form action="action_page.php">
7       <label for="volume">Volume (entre 0 et 100):</label>
8       <input type="range" id="volume" name="volume" min="0" max="100" step="10" />
9       <input type="submit" value="Soumettre" />
10    </form>
11  </body>
12 </html>
```



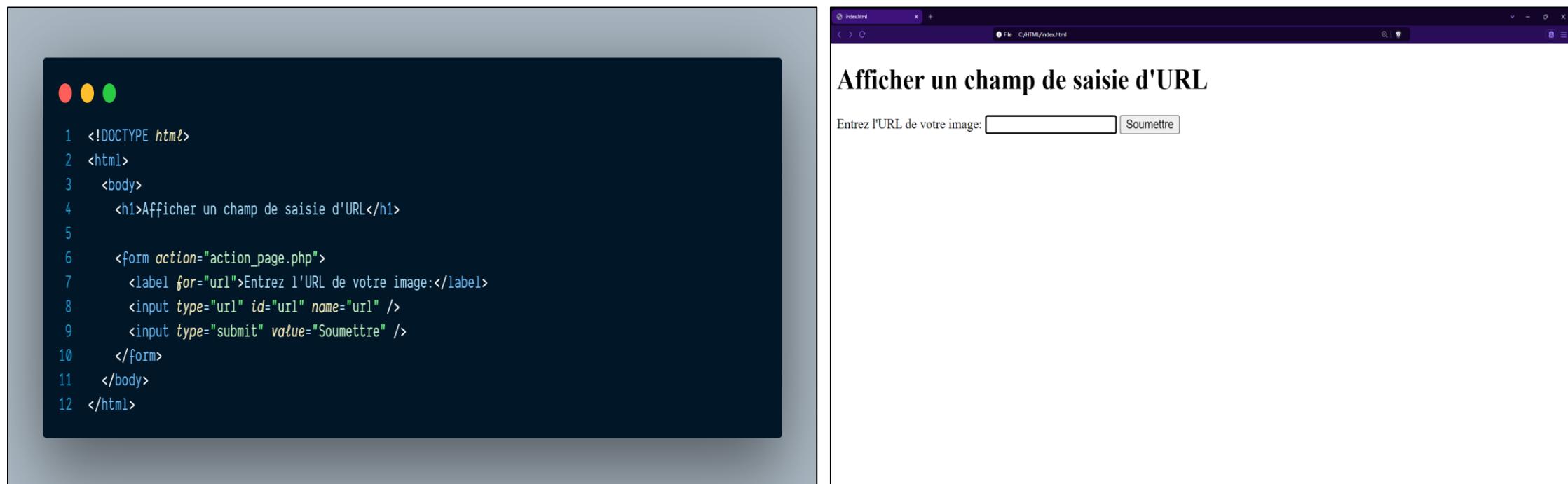
Formulaires HTML

Input type url:

`<input type="url">` est utilisé pour les champs de saisie qui doivent contenir une adresse URL.

Selon la prise en charge du navigateur, le champ URL peut être automatiquement validé lors de sa soumission.

Exemple:



The image shows a split-screen view. On the left, a code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Afficher un champ de saisie d'URL</h1>
5
6     <form action="action_page.php">
7       <label for="url">Entrez l'URL de votre image:</label>
8       <input type="url" id="url" name="url" />
9       <input type="submit" value="Soumettre" />
10    </form>
11  </body>
12 </html>
```

On the right, a web browser window titled "index.html" shows the rendered page. The title is "Afficher un champ de saisie d'URL". The page contains the text "Entrez l'URL de votre image:" followed by an empty input field and a "Soumettre" button.

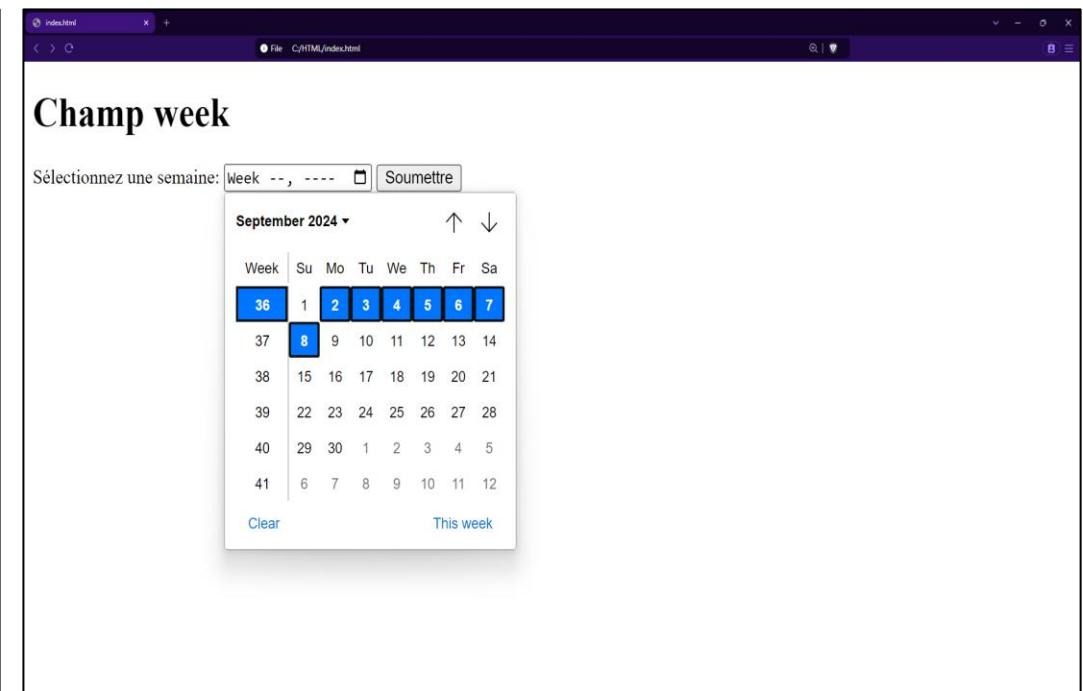
Formulaires HTML

Input type week:

`<input type="week">` permet à l'utilisateur de sélectionner une semaine et une année. Selon la prise en charge du navigateur, un sélecteur de date peut apparaître dans le champ de saisie.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Champ week</h1>
5
6     <form action="action_page.php">
7       <label for="semaine">Sélectionnez une semaine:</label>
8       <input type="week" id="semaine" name="semaine" />
9       <input type="submit" value="Soumettre" />
10    </form>
11  </body>
12 </html>
```



Formulaires HTML

Autres attributes des inputs HTML:

attribut	Description
value	L'attribut value spécifie une valeur initiale pour un champ de saisie
multiple	L'attribut multiple spécifie que l'utilisateur est autorisé à saisir plusieurs valeurs dans un champ de saisie.
placeholder	L'attribut placeholder spécifie une brève indication qui décrit la valeur attendue d'un champ saisie (un exemple de valeur ou une brève description du format attendu).
autofocus	L'attribut autofocus spécifie qu'un champ de saisie doit automatiquement obtenir le focus lors du chargement de la page.
list	L'attribut liste fait référence à un élément <datalist> qui contient des options prédéfinies pour un élément <input>.
autocomplete	L'attribut autocomplete spécifie si la saisie semi-automatique doit être activée ou désactivée pour un formulaire ou un champ de saisie.

Médias HTML

Multimédia HTML:

Le multimédia sur le Web comprend le son, la musique, les vidéos, les films et les animations.

Qu'est-ce que le multimédia?

Le multimédia existe sous de nombreux formats différents. Il peut s'agir de presque tout ce que vous pouvez entendre ou voir, comme des images, de la musique, du son, des vidéos, des disques, des films, des animations, etc.

Les pages Web contiennent souvent des éléments multimédias de différents types et formats.

Prise en charge du navigateur:

Les premiers navigateurs Web ne prenaient en charge que le texte, limité à une seule police dans une seule couleur.

Plus tard, des navigateurs ont pris en charge les couleurs, les polices, les images et le multimédia.

Médias HTML

Formats multimédias:

Les éléments multimédias (comme l'audio ou la vidéo) sont stockés dans des fichiers multimédias.

La façon la plus courante de découvrir le type d'un fichier est de regarder son extension.

Les fichiers multimédias ont des formats et des extensions différentes comme : .wav, .mp3, .mp4, .mpg, .wmv et .avi.

Médias HTML

Formats vidéo courants:



Il existe de nombreux formats vidéo.

Les formats MP4, WebM et Ogg sont pris en charge par HTML.

Le format MP4 est recommandé par YouTube.

Format	File	Description
MPEG	.mpg .mpeg	MPEG. Développé par le Moving Pictures Expert Group. Le premier format vidéo populaire sur le Web. N'est plus pris en charge en HTML.
AVI	.avi	AVI (Audio Video Interleave). Développé par Microsoft. Couramment utilisé dans les caméras vidéo et les téléviseurs. Fonctionne bien sur les ordinateurs Windows, mais pas dans les navigateurs Web.
WMV	.wmv	WMV (Windows Media Video). Développé par Microsoft. Couramment utilisé dans les caméras vidéo et les téléviseurs. Fonctionne bien sur les ordinateurs Windows, mais pas dans les navigateurs Web.
QuickTime	.mov	QuickTime. Développé par Apple. Couramment utilisé dans les caméras vidéo et les téléviseurs. Fonctionne bien sur les ordinateurs Apple, mais pas dans les navigateurs Web.

Médias HTML

RealVideo	.rm .ram	RealVideo. Développé par Real Media pour permettre la diffusion de vidéos avec des bandes passantes faibles. Ne fonctionne pas dans les navigateurs Web.
Flash	.swf .flv	Flash. Développé par Macromedia. Nécessite souvent un composant supplémentaire (plug-in) pour fonctionner dans les navigateurs Web.
Ogg	.ogg	Theora Ogg. Développé par la Fondation Xiph.Org. Supporté par HTML.
WebM	.webm	WebM. Développé par Mozilla, Opera, Adobe et Google. Pris en charge par HTML.
MPEG-4 or MP4	.mp4	MP4. Développé par le groupe d'experts en images animées. Couramment utilisé dans les caméras vidéo et le matériel TV. Pris en charge par tous les navigateurs et recommandé par YouTube.

Remarque: seules les vidéos MP4, WebM et Ogg sont prises en charge par la norme HTML.

Médias HTML

Formats audio courants:

MP3 est le meilleur format pour la musique enregistrée compressée. Le terme MP3 est devenu synonyme de musique numérique. Si votre site Web porte sur la musique enregistrée, le MP3 est le choix idéal.

Format	File	Description
MIDI	.mid	MIDI (Musical Instrument Digital Interface). Format principal pour tous les
	.midi	appareils de musique électronique comme les synthétiseurs et les cartes son pour PC. Les fichiers MIDI ne contiennent pas de son, mais des notes numériques qui peuvent être jouées par des appareils électroniques. Fonctionne bien sur tous les ordinateurs et matériels de musique, mais pas dans les navigateurs Web.
RealAudio	.rm	RealAudio. Développé par Real Media pour permettre la diffusion audio avec de
	.ram	faibles bandes passantes. Ne fonctionne pas dans les navigateurs Web.
WMA	.wma	WMA (Windows Media Audio). Développé par Microsoft. Fonctionne bien sur les ordinateurs Windows, mais pas dans les navigateurs Web.
AAC	.aac	AAC (Advanced Audio Coding). Développé par Apple comme format par défaut pour iTunes. Fonctionne bien sur les ordinateurs Apple, mais pas dans les navigateurs Web.

Médias HTML

WAV	.wav	WAV. Développé par IBM et Microsoft. Compatible avec les systèmes d'exploitation Windows, Macintosh et Linux. Compatible avec HTML.
Ogg	.ogg	Ogg. Développé par la Fondation Xiph.Org. Supporté par HTML.
MP3	.mp3	Les fichiers MP3 sont en fait la partie sonore des fichiers MPEG. Le MP3 est le format le plus populaire pour les lecteurs de musique. Il combine une bonne compression (petits fichiers) avec une haute qualité. Pris en charge par tous les navigateurs.
MP4	.mp4	MP4 est un format vidéo, mais peut également être utilisé pour l'audio. Pris en charge par tous les navigateurs.

Remarque: seuls les fichiers audio MP3, WAV et Ogg sont pris en charge par la norme HTML.

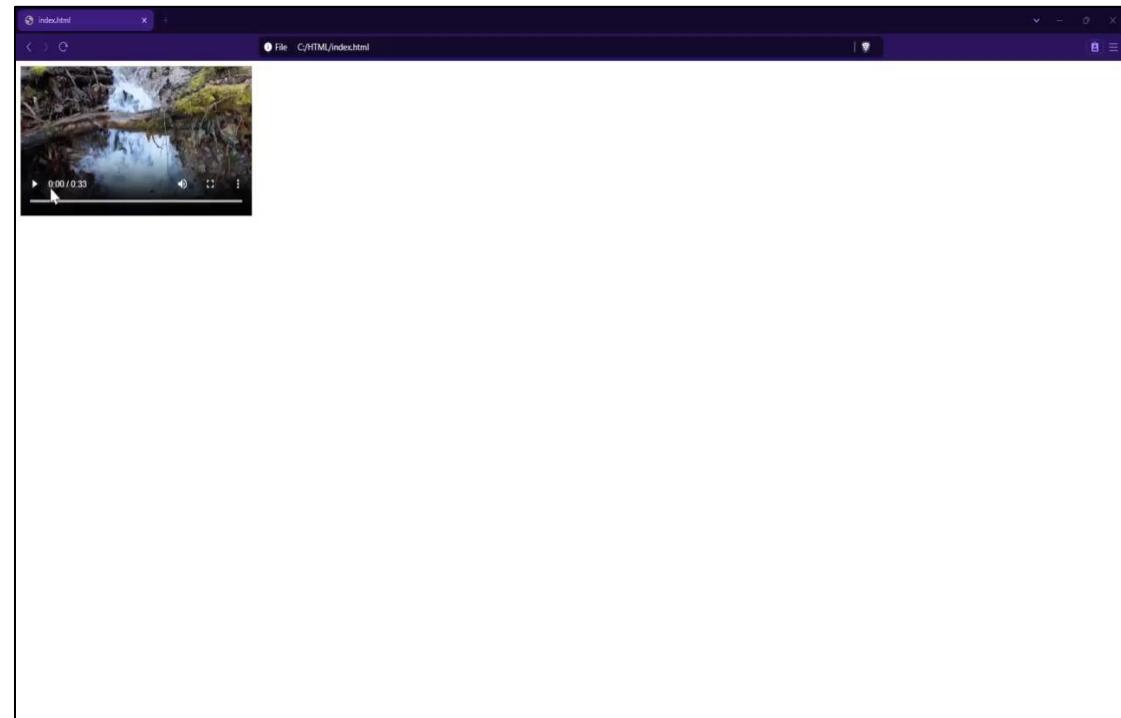
Médias HTML

Vidéo HTML:

L'élément HTML `<video>` est utilisé pour afficher une vidéo sur une page Web.

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <video width="400" controls>
5       <source src="video.mp4" type="video/mp4" />
6       <source src="video.ogg" type="video/ogg" />
7       Votre navigateur ne prend pas en charge la vidéo HTML.
8     </video>
9   </body>
10 </html>
```



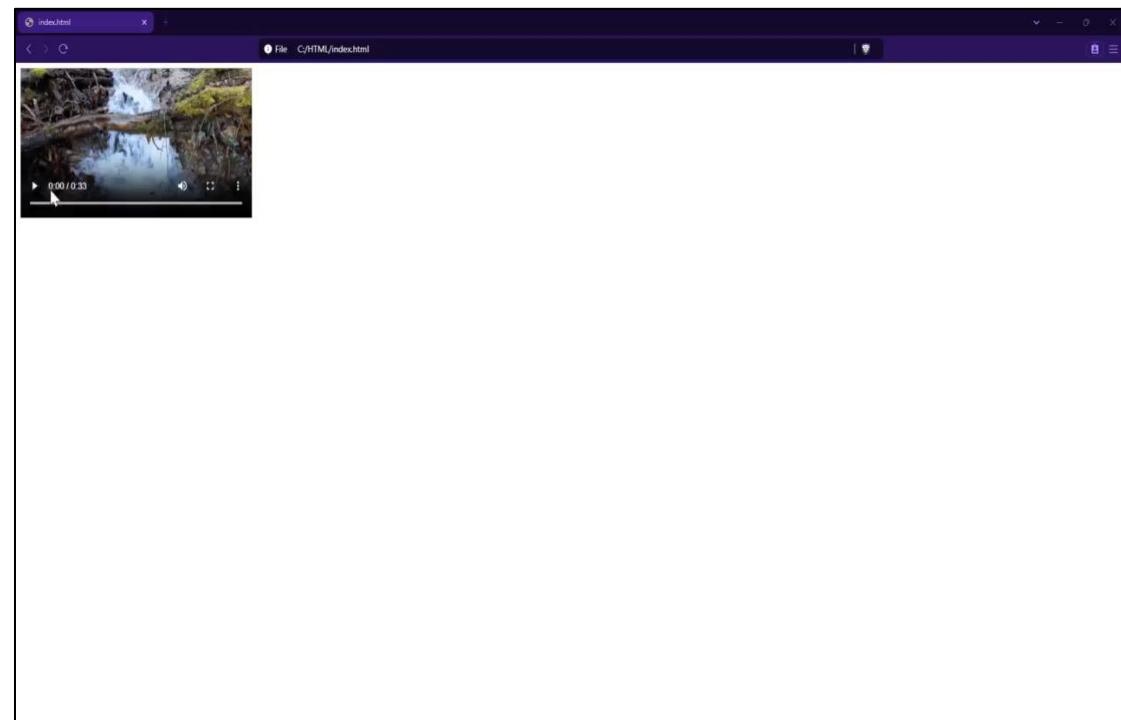
Médias HTML

L'élément HTML <video>:

Pour afficher une vidéo au format HTML, utilisez l'élément <video>:

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <video width="400" controls>
5       <source src="video.mp4" type="video/mp4" />
6       <source src="video.ogg" type="video/ogg" />
7       Votre navigateur ne prend pas en charge la vidéo HTML.
8     </video>
9   </body>
10 </html>
```



Médias HTML

Comment ça marche?

L'attribut `controls` ajoute des commandes vidéo, comme la lecture, la pause et le volume.

Il est conseillé d'inclure systématiquement les attributs `width` et `height`. Si la hauteur et la largeur ne sont pas définies, la page risque de scintiller pendant le chargement de la vidéo.

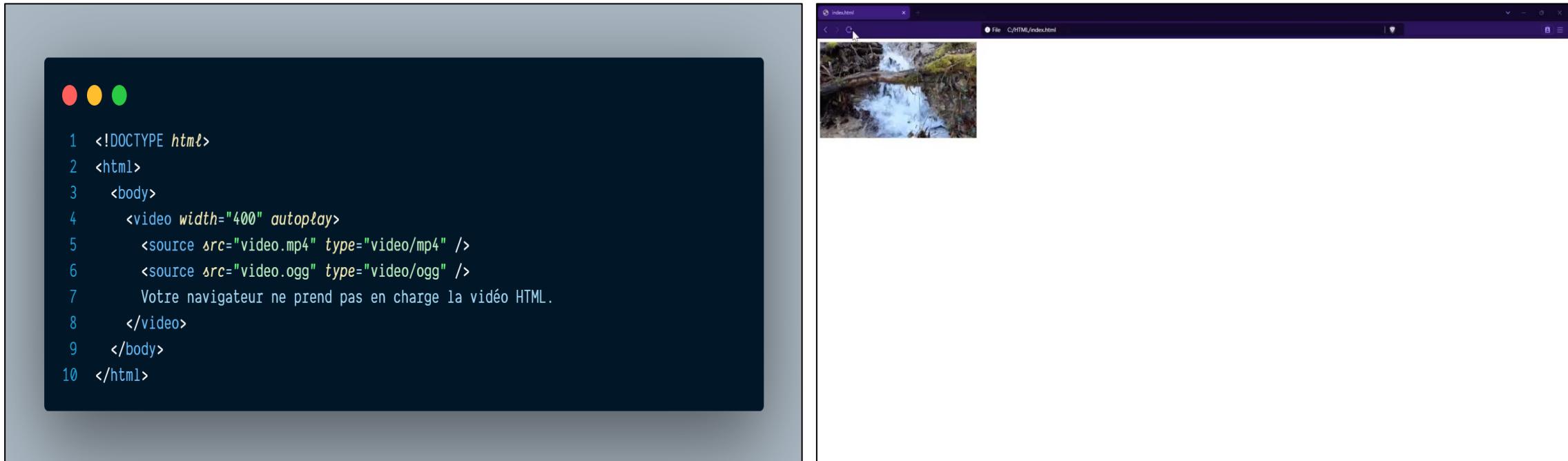
L'élément `<source>` vous permet de spécifier des fichiers vidéo alternatifs parmi lesquels le navigateur peut choisir. Le navigateur utilisera le premier format reconnu.

Le texte entre les balises `<video>` et `</video>` ne s'affichera que dans les navigateurs qui ne prennent pas en charge l'élément `<video>`.

HTML `<video>` Lecture automatique:

Pour démarrer automatiquement une vidéo, utilisez l'attribut `autoplay`:

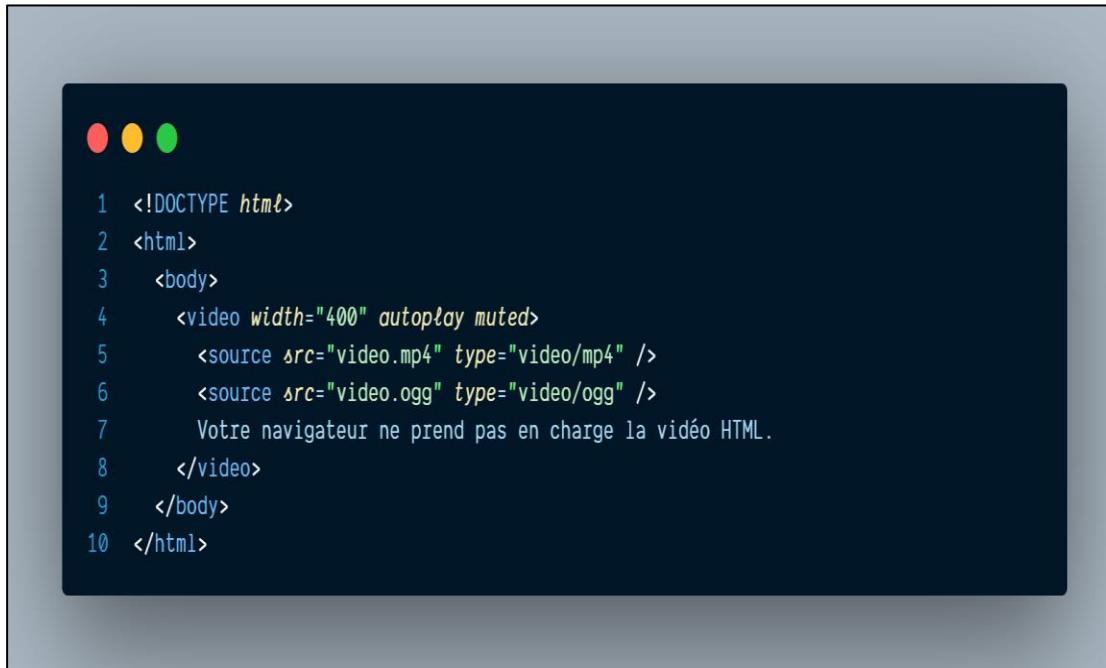
Médias HTML



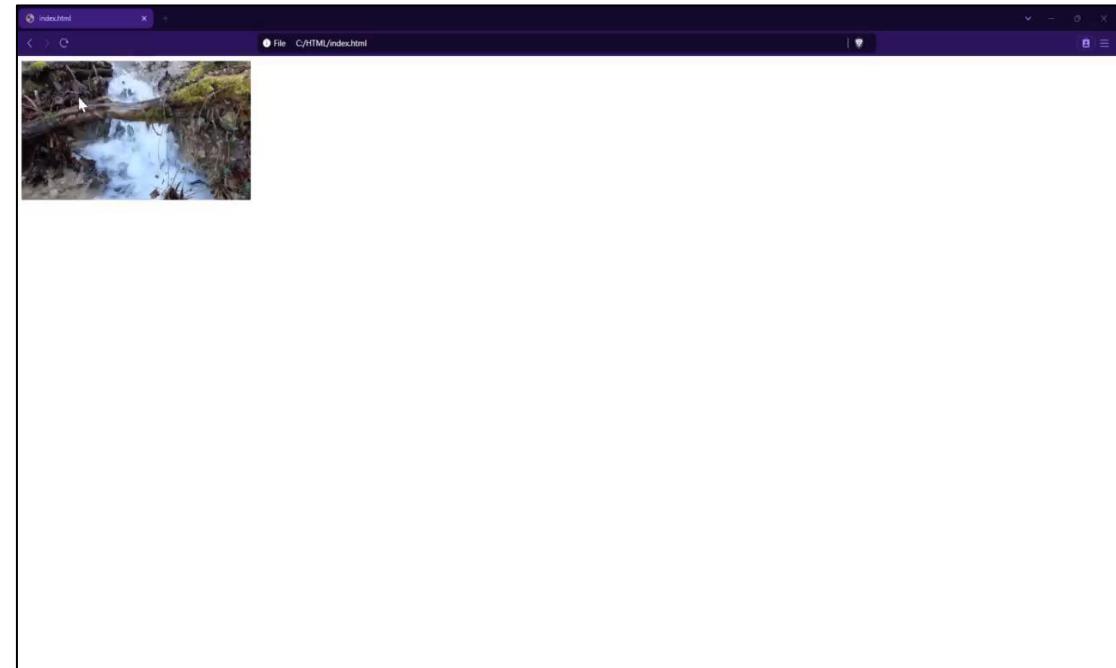
Remarque: les navigateurs Chromium n'autorisent pas la lecture automatique dans la plupart des cas. Cependant, la lecture automatique en mode silencieux est toujours autorisée.

Ajoutez **muet** après la **autoplay** pour permettre à votre vidéo de démarrer automatiquement (mais en mode muet) :

Médias HTML



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <video width="400" autoplay muted>
5       <source src="video.mp4" type="video/mp4" />
6       <source src="video.ogg" type="video/ogg" />
7       Votre navigateur ne prend pas en charge la vidéo HTML.
8     </video>
9   </body>
10 </html>
```



Formats vidéo HTML:

Trois formats vidéo sont pris en charge : MP4, WebM et OGG.

Remarque: Chrome, Opera, Safari, Edge, Firefox prennent en charge ces 3 formats, mais Safari ne prend pas en charge le format Ogg.

Médias HTML

Vidéo HTML - Types de médias

Format de Fichier	Type de média
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

Vidéo HTML attributs:

attribut	Valeur	Description
autoplay	autoplay	Spécifie que la lecture de la vidéo commencera dès qu'elle sera prête
controls	controls	Spécifie que les commandes vidéo doivent être affichées (comme un bouton de lecture/pause, etc.).
height	pixels	Définit la hauteur du lecteur vidéo
loop	loop	Spécifie que la vidéo recommencera à chaque fois qu'elle sera terminée
muted	muted	Spécifie que la sortie audio de la vidéo doit être coupée

Médias HTML

poster	<i>URL</i>	Spécifie une image à afficher pendant le téléchargement de la vidéo ou jusqu'à ce que l'utilisateur appuie sur le bouton de lecture
preload	auto metadata none	Spécifie si et comment l'auteur pense que la vidéo doit être chargée lors du chargement de la page
src	<i>URL</i>	Spécifie l'URL du fichier vidéo
width	<i>pixels</i>	Définit la largeur du lecteur vidéo

Médias HTML

Audio HTML:

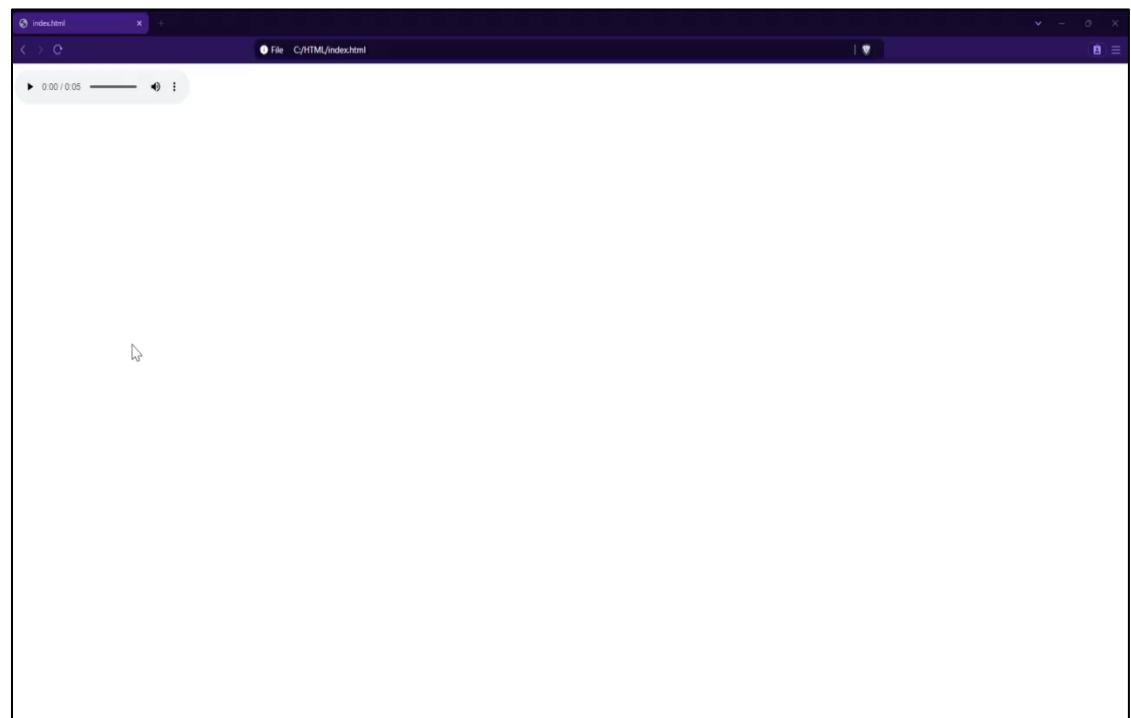
L'élément HTML `<audio>` est utilisé pour lire un fichier audio sur une page Web.

L'élément HTML `<audio>`

Pour lire un fichier audio en HTML, utilisez l'élément `<audio>`:

Exemple:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <audio controls>
5       <source src="audio.mp3" type="audio/ogg" />
6       <source src="video.mp3" type="audio/mpg" />
7       Votre navigateur ne prend pas en charge l'élément audio.
8     </audio>
9   </body>
10 </html>
```



Médias HTML

Comment ça marche?

L'attribut `controls` ajoute des commandes audio, comme la lecture, la pause et le volume.

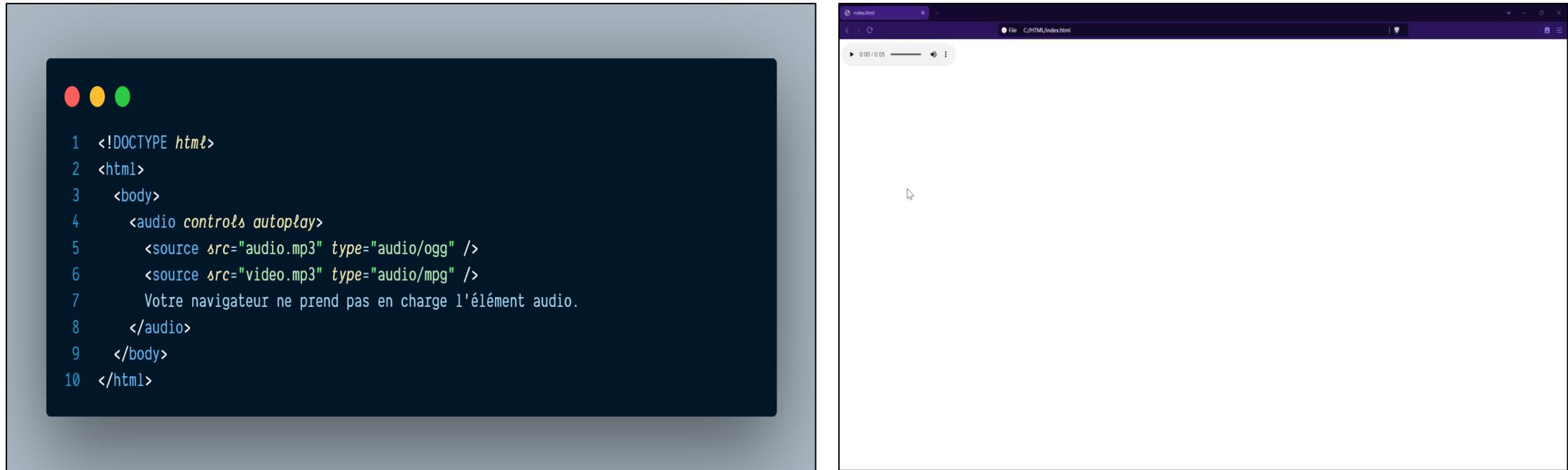
L'élément `<source>` vous permet de spécifier des fichiers audio alternatifs parmi lesquels le navigateur peut choisir. Le navigateur utilisera le premier format reconnu.

Le texte entre les balises `<audio>` et `</audio>` ne s'affichera que dans les navigateurs qui ne prennent pas en charge l'élément `<audio>`.

HTML `<audio>` Lecture automatique:

Pour démarrer automatiquement un fichier audio, utilisez l'attribut `autoplay`:

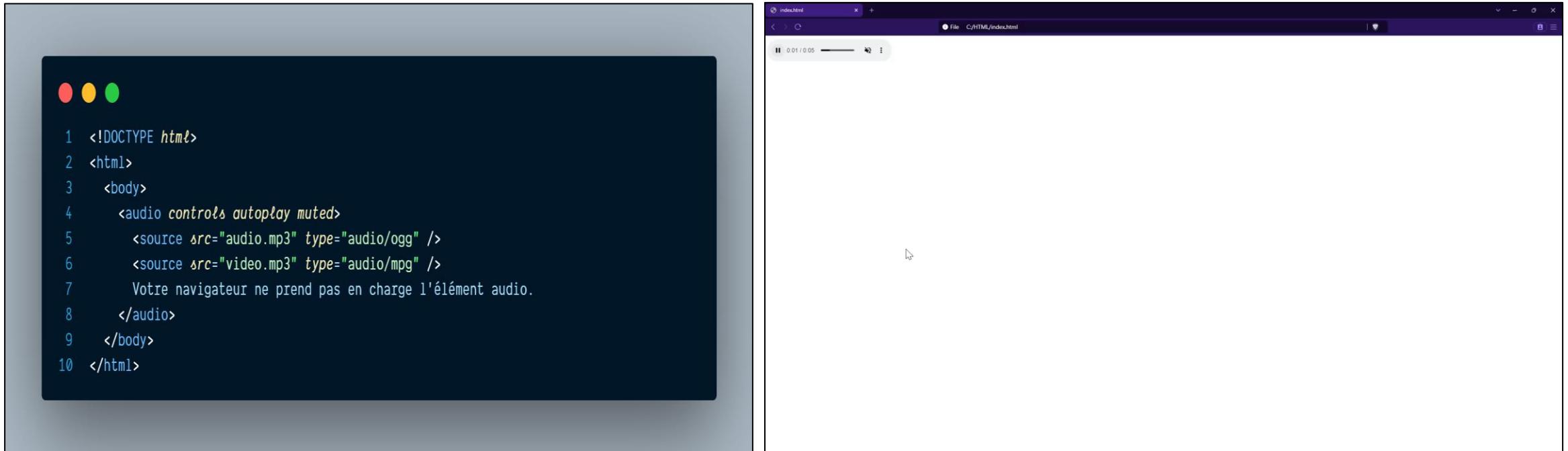
Médias HTML



Remarque: les navigateurs Chromium n'autorisent pas la lecture automatique dans la plupart des cas. Cependant, la lecture automatique en mode silencieux est toujours autorisée.

Ajoutez **muet** après la **autoplay** pour permettre à votre fichier audio de démarrer automatiquement la lecture (mais en mode muet):

Médias HTML



Formats audio HTML:

Trois formats vidéo sont pris en charge : MP3, WAV et OGG.

Remarque: Chrome, Opera, Safari, Edge, Firefox prennent en charge ces 3 formats, mais Safari ne prend pas en charge le format OGG.

Médias HTML

Audio HTML - Types de médias

Format de Fichier	Type de média
MP43	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

Audio HTML attributs:

attribut	Valeur	Description
autoplay	autoplay	Spécifie que l'audio commencera à jouer dès qu'il sera prêt
controls	controls	Spécifie que les commandes audio doivent être affichées (comme un bouton de lecture/pause, etc.)
height	pixels	Spécifie que l'audio recommencera à chaque fois qu'il sera terminé
loop	loop	Spécifie que la vidéo recommencera à chaque fois qu'elle sera terminée
muted	muted	Spécifie que la sortie audio doit être coupée

Médias HTML

preload	auto metadata none	Spécifie si et comment l'auteur pense que l'audio doit être chargé lors du chargement de la page
src	URL	Spécifie l'URL du fichier audio