

Introduction

Introduction

HTML, acronyme de HyperText Markup Language (Langage de Balisage Hypertexte) est la base de tout site web. C'est une forme de langage informatique mais il ne s'agit pas d'un langage de programmation.

HTML est un langage de balisage standard qui définit la structure et le contenu des pages web en utilisant des balises qui précise le type de l'élément intégrer dans la page et comment il doit être reconnu par le navigateur : en tant que paragraphe, liste, en-tête, lien, image, lecteur multimédia, formulaire ou plusieurs autres formes. En plus, de la définition de la structure et la hiérarchie de la page web l'utilisation des balises permet de définir la sémantique du contenu d'une page web ce qui facilite l'accessibilité et l'indexation des données par les moteurs de recherche.

Au fur et à mesure que nous plongeons dans le monde de HTML, nous découvrirons comment ces balises fonctionnent et comment elles façonnent le contenu d'une page web.

Bien que le HTML soit la base de la structure et du contenu, il peut être encore plus ergonomique et interactif lorsqu'il est combiné avec d'autres langages tel que CSS (Cascading Style Sheets) pour styliser les pages HTML en contrôlant les couleurs, les polices, les marges et d'autres aspects visuels.

L'association du HTML avec CSS découle d'une idée fondamentale : la séparation du contenu et de la présentation. Cette approche de dissociation permet aux concepteurs et aux développeurs de travailler de manière plus efficace et organisée. En modifiant simplement les règles CSS, vous pouvez donner un aspect complètement différent à une page HTML sans altérer son contenu.

En plus du CSS, on peut ajouter de l'interactivité aux pages HTML en associant le langage javascript au langage HTML qui injecte de la vie dans les éléments HTML. Cette association tripartite crée une synergie puissante qui transforme les pages web basiques en des pages flexibles, ergonomiques, attrayantes et interactives.

Pour bien manier ces trois langages, HTML, CSS et JavaScript, l'utilisation d'un éditeur de texte spécifique s'avère indispensable pour garantir une structure soignée du code de vos pages HTML.

Il existe une variété d'éditeurs de texte (gratuits et payants) disponibles pour coder en HTML, CSS et JavaScript, chacun offrant des fonctionnalités et des avantages spécifiques. Nous citons par exemple : Visual Studio Code (VS Code), Notepad++, Sublime Text, Atom, et plusieurs autres éditeurs.

Vous avez 3 points sur un total de 3

3/3

1 / 11

Page 1 sur 11.

Introduction

2 / 11

Page 2 sur 11.

Concepts de base

Dans cette section, nous développons les concepts qui servent de base à la création de pages Web. Comprendre ces concepts vous permettra de développer une solide compréhension du langage HTML et de commencer à créer des pages Web simples et structurées.

HTML est constitué d'une large gamme d'éléments ou de balises qui permettent d'englober ou de baliser différentes sections du contenu, afin de les présenter ou de les faire agir d'une manière spécifique. Un navigateur n'affiche pas ces balises HTML, mais les utilise pour déterminer comment afficher le document.

Les balises

Une balise est une suite de mots sans distinction entre majuscules et minuscules qui s'écrivent entre chevrons < et >.

Les balises dénotent la nature du texte qu'elles englobent, et elles offrent au navigateur la compréhension nécessaire pour l'affichage adéquat aux visiteurs d'un site web.

La casse des éléments en HTML n'a pas d'incidence. Ils peuvent être rédigés en majuscules ou en minuscules.

Par exemple, un élément <title> peut être saisi comme <title>, <TITLE>, <Title>, <TiTlE>, etc., et il opérera de manière adéquate.

Toutefois, il est préconisé de consigner tous les éléments en minuscules, dans un souci de cohérence et de lisibilité.

Il existe deux types de balises :

Les **balises en paire** (balises avec contenu) : composées d'une balise ouvrante et d'une balise fermante. Une balise ouvrante est associée à une balise fermante. Par exemple :

```
<p> Mon premier paragraphe</p>
```

On distingue une **balise ouvrante** (<p>) et une **balise fermante** (</p>).

Tout ce qui se trouve en dehors de ces balises n'appartient pas au paragraphe.

Les **balises orpheline** (balise sans contenu) : constituées d'une seule balise. Il n'y a qu'une balise ouvrante. Il n'est pas nécessaire de fermer cette balise. On n'utilise pas de slash (/) à la fin d'une balise orpheline, comme .

Par exemple :

```

```

Les attributs

Les éléments peuvent également être dotés d'attributs, tels que :

```
<p id="section2">Le Madagascar est un pays African. </p>
```

Les attributs fournissent des informations additionnelles concernant l'élément, sans que ces informations n'apparaissent dans le contenu proprement dit.

Pour instaurer un attribut, il faut :

- Introduire un espace entre cet attribut et le nom de l'élément (ou de l'attribut précédent, si l'élément possède déjà un ou plusieurs attributs).
- Nommer l'attribut, puis ajouter un signe égal (=).
- Assigner une valeur à l'attribut, qui doit être encadrée par des guillemets ouvrants et fermants.

Les attributs sont en quelque sorte considérés comme des options pour les balises. Ils viennent les compléter en fournissant des informations additionnelles.

Un attribut se trouve dans la balise ouvrante d'un élément en paire ou directement dans une balise orpheline. Un exemple en est donné avec la balise ci-dessous :

La balise suivante :

```
<a href="https://www.auf.org/" title="Page d'accueil de l'AUF"
```

```
target="_blank">Site Web de l'Agence universitaire de la
```

```
Francophonie</a>
```

La balise `<a>` représente un lien et permet de transformer l'élément qu'elle englobe en lien cliquable. Cette balise peut accepter plusieurs attributs :

1. L'attribut `href` : Cet attribut indique l'adresse web vers laquelle vous souhaitez que le lien dirige. Autrement dit, il spécifie l'URL que le navigateur ouvrira lorsque le lien est cliqué. Par exemple, `href="https://auf.org/"`.
2. L'attribut `title` : Cet attribut fournit des informations additionnelles concernant le lien, telles que le nom de la page à laquelle le lien renvoie. Par exemple, `title="Page d'Accueil de l'AUF"`. Lorsque le curseur survole le lien, ces informations s'affichent sous forme d'une info-bulle.
3. L'attribut `target` : Cet attribut détermine le contexte de navigation utilisé pour afficher le lien. Par exemple, `target="_blank"` ouvrira le lien dans un nouvel onglet du navigateur, permettant ainsi à l'utilisateur de conserver la page actuelle ouverte.

Ces attributs donnent la possibilité de personnaliser le comportement et l'apparence des liens créés avec la balise `<a>`.

Les attributs booléens

Les attributs booléens sont des attributs dépourvus de valeur définie. Ils ne possèdent qu'une seule valeur.

```
<input type="text" disabled="disabled" />
```

Il est tout à fait possible d'exprimer cette même ligne comme suit (nous avons également inclus un élément `input` non désactivé à titre de référence, pour une meilleure compréhension de ce qui se produit) :

```
<input type="text" disabled />
```

Les guillemets

Les valeurs des attributs sont toutes encloses entre des guillemets doubles (") ou des guillemets simples (').

Le choix entre les deux est essentiellement une question de style.

```
<a href="http://www.exemple.com">Un lien vers mon exemple.</a>
```

```
<a href='http://www.exemple.com'>Un lien vers mon exemple</a>
```

Vous pouvez tester et modifier ces codes en utilisant [CodePen](#).

Vous avez 4 points sur un total de 4

4/4

2 / 11

Page 2 sur 11.

Concepts de base

3 / 11

Page 3 sur 11.

Créer Votre Premier Exemple HTML

Premiers Pas dans le Code

Pour commencer le voyage passionnant dans le monde du HTML, nous allons explorer les étapes initiales pour créer votre tout premier exemple HTML.

Étape 1 : Préparation de l'Environnement

Avant de nous lancer, assurez-vous d'avoir un éditeur de texte prêt à être utilisé. Ouvrez votre éditeur et préparez-vous à créer votre première page HTML.

Étape 2 : Rédaction du Code HTML de Base

Commencez par créer un nouveau fichier HTML dans votre éditeur de texte. Ce fichier comporte l'extension ".html", (par exemple, mon-premier-exemple.html) . Ensuite, saisissez les balises de base nécessaires pour structurer votre page.

Étape 3 : Visualisation de la page

Ouvrez le fichier créé dans votre navigateur Web. Vous verrez votre première page HTML avec le contenu ajouté.

Dans les sections à venir, nous plongerons plus profondément dans les balises, les attributs et les concepts qui forment la base du HTML. Préparez-vous à explorer davantage et à créer des pages Web de plus en plus dynamiques et captivantes.

3 / 11

Page 3 sur 11.

Créer Votre Premier Exemple HTML

4 / 11

Page 4 sur 11.

Anatomie d'un document HTML

Structure d'un document HTML

Commentaires

En HTML, tout comme dans la plupart des langages de programmation, il existe un mécanisme permettant d'insérer des commentaires dans le code. Les commentaires sont ignorés par le navigateur et restent invisibles pour l'utilisateur.

L'objectif des commentaires est de fournir des annotations dans le code pour expliquer son fonctionnement, décrire les diverses parties du code, etc. Cela peut être extrêmement utile pour se rappeler de l'intention du code après un certain temps ou lorsque le code est partagé avec quelqu'un d'autre pour collaboration.

Pour convertir une portion de contenu dans votre fichier HTML en commentaire, vous devez l'entourer des marqueurs spéciaux `<!--` et `-->`, comme ceci

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

Vous avez 5 points sur un total de 5

4 / 11

Page 4 sur 11.

Anatomie d'un document HTML

5 / 11

Page 5 sur 11.

Métadonnées en HTML

En-tête

L'en-tête `<head>` dans un document HTML représente une section du document qui n'est pas visible dans le navigateur lors du chargement de la page. Situé entre les balises `<html>` et `<body>`, l'élément `<head>` agit comme un panneau de contrôle, fournissant des instructions précieuses au navigateur et aux moteurs de recherche.

Cet élément contient :

- Le titre de la page `<title>` : apparaît dans l'onglet du navigateur, offrant aux utilisateurs un aperçu de ce qu'ils peuvent attendre de votre page. Il est également utilisé par les moteurs de recherche pour afficher des résultats pertinents.
- Des liens vers des feuilles de styles CSS (si vous souhaitez appliquer des styles au contenu HTML).
- Des liens externes : Vous pouvez également insérer des liens vers d'autres ressources importantes
- Balises `<meta>` : Les balises `<meta>` offrent des informations supplémentaires sur la page, comme une description concise ou des mots-clés pertinents. Ces informations peuvent être utilisées par les moteurs de recherche pour indexer et classer votre page.

Exemple :

```
<head>
```

```
<!-- Auteur de la page: D-CLIC -->
```

```
<meta name="author" content="D-CLIC">
```

```
<title>Ma page de test</title>
```

```
<!-- Lien vers une feuille de styles CSS -->
```

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

```
</head>
```

La balise `<meta>`

L'en-tête peut englober un assortiment varié de balises `<meta>`. Cette balise a pour rôle de définir des métadonnées, c'est-à-dire des informations destinées au navigateur et aux moteurs de recherche. Ces informations ne seront pas affichées à l'écran.

Vous pouvez spécifier l'encodage de caractères. Cela assure que les caractères spéciaux et les symboles seront affichés correctement et évitant tout problème d'affichage indésirable.

La balise `<meta>` peut contenir des indications telles que :

- Mots-clés : liste de mots-clés pertinents pour l'indexation la page.
- Description de la page : courte description de la page, souvent utilisée par les moteurs de recherche pour afficher un extrait dans les résultats de recherche.
- Auteur de la page : nom de l'auteur de la page, ce qui peut être utile pour l'identification.
- Encodage de Caractères : vous pouvez spécifier l'encodage des caractères pour vous assurer que les caractères spéciaux et internationaux sont affichés correctement. L'encodage UTF-8 est un standard crucial qui assure que les caractères et symboles de toutes les langues puissent être correctement représentés et interprétés dans les applications, les sites web et autres.
- Date de mise à jour : vous pouvez spécifier le délai en secondes après lequel la page sera actualisée ou redirigée.
- Mode d'affichage : pour configurer la manière dont un site web est affiché sur les appareils mobiles (smartphone et tablette)
- Et bien plus encore.

Exemple :

```
<!DOCTYPE html>
```



```
<HTML>
```

```
<HEAD>
```

```
<!-- Mots-clés: HTML, CSS, JavaScript -->
```

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

```
<!-- Descriptions de la page Web : Initiation en HTML -->
```

```
<meta name="description" content="Initiation en HTML">
```

```
<!-- Auteur de la page: D-CLIC -->
```

```
<meta name="author" content="D-CLIC">
```

```
<!-- Encodage de Caractères : UTF-8 -->
```

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

```
<!-- Date d'actualisation de la page : 60 secondes -->
```

```
<meta http-equiv="refresh" content="60">
```

```
<!-- Mode d'affichage: Le site Web a belle apparence sur tous les  
appareils: -->
```

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

```
scale=1.0">
```

```
</HEAD>
```

```
<BODY>
```

```
</BODY>
```

```
</HTML>
```

Vous avez 4 points sur un total de 4

4/4

5 / 11

Page 5 sur 11.

Métadonnées en HTML

6 / 11

Page 6 sur 11.

Organiser votre texte

Les bases : titres et paragraphes

La plupart des contenus textuels incluent des titres et des paragraphes, qu'il s'agisse de romans, de journaux, de manuels scolaires, de magazines, et autres.

En HTML, les paragraphes doivent être encapsulés dans un élément `<p>`, de la manière suivante :

```
<p>Je suis un paragraphe, Ah oui je le suis.</p>
```

Chaque titre doit être inclus dans un élément de titre :

```
<h1>Je suis un titre</h1>
```

Il existe 6 éléments de titre disponibles : `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` et `<h6>`.

Chaque élément représente un niveau de titre différent ; `<h1>` indique le titre principal, `<h2>` représente un sous-titre, `<h3>` un sous-sous-titre, et ainsi de suite jusqu'au niveau de titre le plus bas, qui est `<h6>`.

Exemple :

```
<h1>Initiation au développement Web</h1>
```

```
<p>Bienvenue dans le cours de développement web !!!!</p>
```

```
<p>Vous allez apprendre à façonner des sites web interactifs et à  
donner vie à vos idées de création. </p>
```

```
<h2>Introduction au web</h2>
```

```
<h2>Historique du web</h2>
```

```
<h2>Architecture du web </h2>
```

```
<h3>Cas du web statique</h3>
```

```
<h3>Cas du web dynamique</h3>
```

```
<h2>Le langage HTML</h2>
```

<p> Préparez-vous à explorer le langage HTML.

Cette section est conçue pour vous fournir les bases solides

nécessaires

pour maîtriser le langage HTML.

Alors, préparez-vous à coder, à concevoir et à innover dans le monde

du développement web ! </p>

<h3>Concepts de base</h3>

<h3>Anatomie d'un document HTML</h3>

.....

Vous pouvez tester et modifier ce code en utilisant [CodePen](#).

C'est en effet vous qui déterminez le sens que revêtent les éléments utilisés, du moment que la structure hiérarchique reste cohérente. Cependant, il est recommandé de suivre quelques bonnes pratiques lors de la création de telles structures :

1. Privilégiez l'usage d'un seul <h1> par page, car c'est le niveau principal, et les autres devraient être d'un niveau inférieur.
2. Assurez-vous d'appliquer les balises de titre dans un ordre logique : <h1>, puis <h2>, suivi de <h3>, et ainsi de suite.
3. Même s'il existe 6 niveaux de titre (<h1> à <h6>), il est déconseillé d'utiliser plus de trois niveaux sur une seule page. Les documents comportant de nombreux niveaux deviennent complexes et difficiles à naviguer. Dans ce cas, il vaut mieux découper le contenu en plusieurs pages distinctes.

Vous avez 4 points sur un total de 4

4/4

6 / 11

Page 6 sur 11.

Organiser votre texte

7 / 11

Page 7 sur 11.

Les hyperliens

Les hyperliens

Les hyperliens constituent l'une des innovations les plus captivantes offertes par le Web. Ils permettent de relier nos documents à d'autres documents de notre choix.

Nous pouvons établir des liens vers des parties spécifiques de documents et rendre des applications accessibles via une simple adresse web (contrairement aux applications natives nécessitant une installation).

Pratiquement tout contenu en ligne peut être transformé en un lien, de telle manière que lorsqu'il est cliqué (ou activé autrement), il dirige le navigateur vers une autre adresse web.

Par exemple, la page d'accueil suivante comporte de nombreux liens qui pointent non seulement vers divers articles d'actualité, mais aussi vers d'autres sections du site (fonctionnalité de navigation), des pages d'inscription/connexionfrance, et bien plus encore.

Anatomie d'un lien

Créer un lien simple implique d'insérer le texte (ou tout autre contenu, selon les liens de niveau bloc) que vous souhaitez transformer en lien à l'intérieur d'une balise `<a>`, puis de lui attribuer un attribut `href` (également appelé Hypertext Reference) contenant l'adresse web à laquelle le lien doit pointer.

Exemple :

```
<p>
```

```
Je suis en train de créer un lien à
```

```
<a href="https://www.auf.org">la page d'accueil de l'AUF</a>.
```

```
</p>
```

Cela produit le résultat suivant : "Je suis en train de créer un lien vers [la page d'accueil de l'AUF](#)."

Un autre attribut pouvant être ajouté à un lien est "title". Cet attribut vise à contenir des informations supplémentaires utiles concernant le lien, telles que le contenu de la page ou des détails importants.

Exemple :

```
<p>
```

```
Je suis en train de créer un lien à
```

```
<a
```

```
href="https://www.auf.org"
```

```
title="Les événements, les actualités et les appels à
```

```
candidatures destinées au communauté francophone">
```

```
la page d'accueil de l'AUF</a>.
```

```
</p>
```

Le résultat affiche le contenu de "title" dans une info-bulle lorsque le pointeur de la souris survole le lien : "Je suis en train de créer un lien vers [la page d'accueil de l'AUF](#)".

Un lien peut être ajouté à l'intérieur d'un paragraphe, d'une liste ou d'un autre élément de texte.

Exemple :

```
<p>Visitez
```

```
<a href="https://www.francophonie.org/dcllic">notre site Web</a>
```

```
pour en savoir plus.</p>
```

Pratiquement tout type de contenu peut être converti en lien. Par exemple on peut transformer une image en lien, il vous suffit d'entourer l'image avec les balises <a>. Exemple :

Exemple :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Deuxième exemple de lien </title>
```

```
</head>
```

```
<body>
```

```
<h1>Bienvenue sur notre site Web</h1>
```

```
<p>Nous vous invitons à explorer plus de type de liens</p>
```

```
<a href="https://www.auf.org">  </a>
```

```
</body>
```

```
</html>
```

Les Liens Hypertextes et les Chemins d'Accès

Lorsque vous souhaitez créer un lien hypertexte vers une autre page ou un autre fichier sur votre site web, il est important de spécifier le chemin d'accès correct pour que le navigateur puisse trouver le contenu souhaité.

○

Même Dossier : Si le fichier que vous voulez lier est dans le même dossier que votre page HTML actuelle, vous pouvez simplement utiliser le nom du fichier, par exemple : `< a href="page2.html"> Lien vers une autre page< /a>`. Le navigateur sait que le fichier se trouve dans le même répertoire.

○

○

Dossier Parent : Si le fichier que vous voulez lier est dans un dossier parent (un niveau au-dessus) de votre page HTML actuelle, vous pouvez utiliser `..` pour indiquer le dossier parent, par exemple : `< a href="../page2.html"> Lien vers une autre page< /a>`.

○

○

Chemin Relatif : Si le fichier que vous voulez lier est dans un dossier différent mais dans la même arborescence que votre page HTML actuelle (par exemple il se trouve dans un sous-dossier qui s'appelle "contenu"), vous pouvez spécifier le chemin relatif depuis le répertoire actuel, par exemple : `< a href="contenu/page2.html"> Lien vers une autre page< /a>`.

○

Un lien vers une ancre

Une ancre est un moyen de créer des références et des liens internes à l'intérieur d'une page HTML. Elle agit comme un point de repère que vous pouvez utiliser pour cibler des sections spécifiques du contenu de la page. Les ancres permettent aux utilisateurs de naviguer facilement, dans de longues pages, directement vers les informations pertinentes, sans avoir à faire défiler manuellement.

Pour définir une ancre, suivez ces étapes :

1.

Choisissez l'Endroit de l'Ancre : Identifiez l'endroit dans votre page où vous souhaitez créer l'ancre. Cela pourrait être un titre, un paragraphe ou n'importe quelle section que vous souhaitez cibler.

2.

3.

Ajouter l'Attribut id : Une fois que vous avez identifié l'endroit de l'ancree, ajoutez l'attribut id à la balise qui englobe cette section. L'attribut id doit être unique sur la page, c'est-à-dire qu'aucun autre élément ne doit avoir le même id. Par exemple :

4.

```
<h2 id="section1">Section 1</h2>
```

```
<p>Contenu de la section 1...</p>
```

5.

6.

Créer un Lien vers l'Ancree : Maintenant que vous avez défini l'ancree avec l'attribut id, vous pouvez créer un lien qui pointe vers cette ancree. Utilisez la balise <a> avec l'attribut href pour préciser l'URL de la page, suivi d'un symbole dièse (#) suivi de l'identifiant de l'ancree. Par exemple :

7.

```
<a href="#section1">Aller à la Section 1</a>
```

8.

Lorsque les utilisateurs cliquent sur le lien, ils seront automatiquement déplacés vers la section correspondante de la page.

Ancree située dans une autre page : On peut créer un lien vers une ancree située dans une autre page. Il suffit d'ajouter le nom de la page avant l'id de l'ancree.

Voici un exemple explicatif : Supposons que vous ayez deux fichiers HTML : page1.html et page2.html sur le même chemin. Dans page2.html, vous avez créé une ancree avec l'attribut id défini comme section2. Voici comment créer un lien dans page1.html qui pointe vers l'ancree section2 dans page2.html :

Dans le fichier page2.html :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Page 2</title>
```

```
</head>
```

```
<body>
```

```
<h1>Page 2</h1>
```

```
<p>Contenu de la Section 1...</p>
```

```
<h2 id="section2">Section 2</h2>
```

```
<p>Contenu de la Section 2...</p>
```

```
</body>
```

```
</html>
```

Dans le fichier `pagel.html` :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Lien vers Ancre</title>
```

```
</head>
```

```
<body>
```

```
<h1>Page 1</h1>
```

```
<p>Voici un <a href="page2.html#section2">lien vers la Section 2
```

```
de la Page 2</a>.</p>
```

```
</body>
```

```
</html>
```

```
Da
```

Dans cet exemple, le lien `<a>` dans `page1.html` pointe vers `page2.html` avec l'ancree `#section2` attachée à la fin de l'URL. Lorsque les utilisateurs cliquent sur ce lien, ils seront dirigés vers la `page2.html` et l'ancree `section2`, ce qui fera défiler la page jusqu'à la section correspondante.

Liens vers des ressources non-HTML

Lorsque vous créez un lien vers une ressource téléchargeable (comme un fichier PDF ou un document Word), une ressource à diffuser en continu (comme une vidéo ou de l'audio), ou toute autre ressource ayant un effet potentiellement inattendu (comme l'ouverture d'une fenêtre pop-up ou le chargement d'une vidéo Flash), il est essentiel de le signaler clairement afin de prévenir toute confusion. Cette précaution peut éviter des situations gênantes, telles que :

- Initier un téléchargement de plusieurs mégaoctets de manière inattendue lorsque vous cliquez sur un lien, surtout si vous êtes connecté à une faible bande passante.
- Être redirigé vers une page nécessitant Flash Player après avoir cliqué sur un lien, en cas d'absence de Flash Player installé.

Pour illustrer ce point, voici quelques exemples de formulations que vous pourriez utiliser :

```
<p>
```

```
<a href="https://www.auf.org/wp-content/uploads/2019/04/Plaque-
```

```
AUF.pdf">
```

```
Télécharger ce rapport (PDF, 1563Ko)
```

```
</a>
```

```
</p>
```

```
<p>
```

```
<a href="https://www.youtube.com/watch?v=5Q4HJZit-Dg"
```

```
target="_blank">
```

```
  Regarder la vidéo (le flux s'ouvre dans un nouvel onglet,
```

```
  qualité HD)
```

```
</a>
```

```
</p>
```

Attribut download :

Lorsque vous créez un lien vers une ressource destinée au téléchargement plutôt qu'à l'ouverture dans le navigateur, vous avez la possibilité d'utiliser l'attribut "download" pour spécifier un nom de fichier par défaut pour l'enregistrement. Voici un exemple concret illustrant l'utilisation de cet attribut avec un lien permettant de télécharger la dernière version d'un l'éditeur VS code pour Windows :

```
<a
```

```
  href="https://az764295.vo.msecnd.net/stable/6c3e3dba23e8fadc360aed75
```

```
ce363ba185c49794/VSCodeUserSetup-x64-1.81.1.exe">
```

```
  Télécharger la version de VS code pour Windows (64-bit)>
```

```
</a>
```

Liens de courriel

Il est possible de créer des liens ou des boutons qui permettent aux visiteurs avec un simple clique d'ouvrir leur logiciel de messagerie électronique et d'envoyer un courriel à une adresse électronique spécifique. Cela peut être utile pour encourager l'interaction avec les utilisateurs et pour fournir un moyen de contact direct. C'est fait en utilisant l'élément `<a>` et le schéma d'URL `mailto:`.

Sous sa forme la plus basique et la plus communément utilisée, un lien `mailto:` indique simplement l'adresse du destinataire voulu. Par exemple :

```
<a href="mailto:formation@auf.org"> Envoyer un courriel à
```

```
formation@auf.org</a>
```

Ceci donne un résultat qui ressemble à ceci : [Envoyer un courriel à formation@auf.org](mailto:formation@auf.org).

En fait, l'adresse de courriel est même optionnelle. Si vous l'omettez (c'est-à-dire, si votre `href` est simplement "mailto:"), une nouvelle fenêtre de courriel sortant sera ouverte par le client de courriel de l'utilisateur, sans adresse de destination encore spécifiée. Ceci est souvent utilisé pour les liens "Partager", permettant aux utilisateurs de cliquer pour envoyer un courriel à une adresse de leur choix.

Spécification des détails

En plus de l'adresse électronique, vous avez la possibilité de fournir d'autres informations dans un lien "mailto:". En fait, tous les champs d'en-tête standard peuvent être ajoutés à l'URL "mailto:" que vous spécifiez. Parmi les plus couramment utilisés figurent "subject" (sujet), "cc" (copie carbone) et "body" (corps du message) — bien que ce dernier ne soit pas strictement un champ d'en-tête, il vous permet d'ajouter un court message au contenu du nouveau courriel. Chaque champ est ajouté à l'URL sous forme de requête.

Voici un exemple illustrant l'inclusion des champs "cc" (copie carbone), "bcc" (copie carbone cachée), "subject" (sujet) et "body" (corps du message) dans un lien "mailto:" :

```
<a
```

```
href="mailto:formation@auf.org?cc=communication@auf.org&bcc=directeu
```

```
r@rapid.com&subject=FormationHTML&body=Le%20corps%20du%20courriel">
```

```
 Envoyer un e-mail avec copie, copie cachée, sujet et corps de
```

```
message
```


Vous avez 4 points sur un total de 5

4/5

Correction Voir la solution

URL absolue vs. URL relative

Les termes fréquemment utilisés dans le contexte du web sont "URL absolue" et "URL relative":

- **URL absolue** : Elle pointe vers une adresse précise sur le web, en incluant le protocole et le nom de domaine. Par exemple, si une page `index.html` est placée dans le dossier "projets" sur le site web <http://www.exemple.com>, alors son adresse complète serait <http://www.exemple.com/projets/index.html>. Une URL absolue reste inchangée, peu importe où elle est utilisée.
- **URL relative** : Elle pointe vers une adresse en fonction de la position du fichier à partir duquel le lien est créé. Par exemple, si vous voulez établir un lien depuis le fichier principal `index.html` du site <http://www.exemple.com> vers un fichier PDF dans le même dossier, l'URL relative serait simplement le nom du fichier, tel que `fichier.pdf`, sans nécessité d'informations supplémentaires. Si ce PDF est dans un sous-dossier nommé "documents" dans "projets", l'URL relative serait `documents/fichier.pdf`.
Une URL relative varie en fonction de l'emplacement du fichier source. Si vous déplacez le fichier `index.html` du dossier "projets" vers le répertoire principal du site web (niveau supérieur, sans dossier), alors un lien relatif vers un fichier `documents/fichier.pdf` pointerait désormais vers une adresse différente, dépendamment de la nouvelle position.

Cependant, notez que le déplacement du fichier `index.html` n'altère pas automatiquement la position du fichier PDF ou du dossier "documents". Ainsi, le lien pourrait pointer vers une mauvaise destination et ne pas fonctionner correctement. C'est pourquoi il est important d'être vigilant.

En prenant en compte cette explication, bien que les liens absolus puissent sembler plus stables en évitant les problèmes liés aux déplacements de fichiers, il est recommandé d'utiliser des liens relatifs autant que possible pour faire référence à des emplacements au sein d'un même site web.

7 / 11

Page 7 sur 11.

Les hyperliens

Formatage avancé du texte

Les listes

HTML prend en charge plusieurs types de listes pour organiser et structurer le contenu d'une page web. Voici les trois types principaux de listes de HTML :

- Liste ordonnée
- Liste non ordonnée
- Liste descriptive

Listes ordonnées

Les listes ordonnées sont comme une liste de tâches numérotées. Chaque tâche a un numéro pour montrer son ordre.

Exemple :

```
<ol>
```

```
<li>HTML</li>
```

```
<li>CSS</li>
```

```
<li>JavaScript</li>
```

```
</ol>
```

Résultat :

1. HTML
2. CSS
3. JavaScript

Listes non ordonnées

Pensez à une liste de courses que vous rédigez avant d'aller faire les courses. Les listes non ordonnées sont comme des listes d'articles sans ordre particulier. Vous voulez acheter des pommes, des bananes et des oranges, mais il n'y a pas d'ordre spécifique dans lequel vous devez les acheter.

Exemple :

```
<ul>
```

```
<li>Informatique</li>
```

```
<li>Mécanique</li>
```

```
<li>Électrique</li>
```

```
</ul>
```

Résultat :

- Informatique
- Mécanique
- Électrique

Listes descriptives

Les listes descriptives sont utilisées pour associer des éléments avec leurs descriptions, tels que des termes et leurs définitions, ou encore des questions et leurs réponses.

Les listes descriptives sont balisées différemment par rapport aux autres types de listes. Elles utilisent l'élément `<dl>` comme conteneur global ; chaque terme est enveloppé dans un élément `<dt>` (signifie "description term"), et chaque description est enveloppée dans un élément `<dd>` (signifie "description definition").

Exemple :

```
<dl>
```

```
<dt>HTML</dt>
```



```
<dd>HyperText Markup Language. Le langage de balisage utilisé pour  
structurer le contenu des pages web.</dd>  
  
<dt>CSS</dt>  
  
<dd>Cascading Style Sheets. Une technologie utilisée pour définir  
la présentation et le style des pages web.</dd>  
  
<dt>JavaScript</dt>  
  
<dd>Un langage de programmation utilisé pour rendre les pages web  
interactives et dynamiques.</dd>  
  
</dl>
```

Résultat :

HTML

HyperText Markup Language. Utilisé pour créer des pages web.

CSS

Cascading Style Sheets. Utilisé pour rendre les pages belles.

JavaScript

Un langage utilisé pour rendre les pages interactives.

Chaque type de liste a sa propre façon de montrer l'information. Les listes ordonnées utilisent des numéros, les listes non ordonnées utilisent des points, et les listes descriptives associent des mots à leurs significations.

Vous pouvez tester et modifier ces codes en utilisant [CodePen](#).

Citations

Le langage HTML propose également des fonctionnalités permettant de mettre en évidence les citations. Le choix de la balise à utiliser dépend du type de citation : qu'il s'agisse d'une citation en bloc ou d'une citation en ligne

Citations en bloc :

Lorsqu'une section ou un contenu de niveau bloc (comme un paragraphe, plusieurs paragraphes, une liste, etc.) est tiré d'une autre source, il est possible de l'indiquer en l'encadrant avec l'élément `<blockquote>` et en fournissant une URL pointant vers la source de la citation via l'attribut "cite".

Voici un **exemple** de l'élément `<blockquote>`.

```
<p>
```

```
L'<strong>Élément HTML <code>&lt;blockquote&gt;</code></strong>
```

```
(ou <em>Élément HTML bloc de citation</em>) indique que le bloc de
```

```
texte inclus
```

```
est une citation étendue.
```

```
</p>
```

Exemple de bloc de citation. Pour le changer en bloc de citation, on ferait simplement ceci :

```
<blockquote cite="/fr/docs/Web/HTML/Element/blockquote">
```

```
<p>
```

```
L'<strong>Élément HTML <code>&lt;blockquote&gt;</code></strong>
```

```
(ou
```

```
<em>Élément HTML bloc de citation</em>) indique que le bloc de
```

```
texte inclus
```

```
est une citation étendue.
```

```
</p>
```

```
</blockquote>
```

Par défaut, le navigateur affichera ce contenu comme un paragraphe indenté, signalant ainsi qu'il s'agit d'une citation.

Vous pouvez tester et modifier ces codes en utilisant [CodePen](#).

Citations en ligne

Les citations en ligne suivent exactement le même principe, à la différence près que l'on emploie l'élément `<q>`.

À titre d'exemple, le balisage ci-dessous présente une citation en ligne :

```
<p>
```

```
L'élément citation — <code>&lt;q&gt;</code> — est
```

```
<q cite="/fr/docs/Web/HTML/Element/q">
```

```
prévu pour de courtes citations ne nécessitant pas un nouvel
```

```
alinéa
```

```
</q>
```

```
</p>
```

Le navigateur affichera par défaut ce texte entre guillemets pour signaler qu'il s'agit d'une citation, comme illustré ci-dessous :

Abréviations

Les abréviations en HTML servent à représenter des expressions ou des termes abrégés au sein du contenu d'une page web.

Ces abréviations sont définies à l'aide de l'élément `<abbr>`, auquel on ajoute un attribut `title` pour fournir une description complète de l'abréviation.

Lorsque les utilisateurs survolent l'abréviation avec leur curseur, le navigateur affiche une info-bulle contenant la signification complète de l'abréviation, améliorant ainsi la compréhension du contenu.

```
<p>
```

```
HTML est un langage de <abbr title="Hypertext Markup
```

```
Language">marquage hypertexte</abbr>
```

```
utilisé pour structurer le contenu des pages web.
```

```
</p>
```

Dans cet exemple, si un utilisateur survole "HTML", une info-bulle apparaîtra avec la description "Hypertext Markup Language".

Cela permet d'enrichir l'expérience de lecture en fournissant des informations explicites sur les abréviations utilisées.

Vous pouvez tester et modifier ce code en utilisant [CodePen](#).

Balisage des détails de contact

Le balisage des détails de contact en HTML permet d'afficher des informations de contact de manière structurée sur une page web.

Cela peut inclure des éléments tels que l'adresse, le numéro de téléphone, l'adresse e-mail, etc. Voici comment cela peut être fait avec des exemples :

- **Adresse** : L'élément `<address>` est utilisé pour marquer une section contenant des informations de contact, telles qu'une adresse postale.

```
<address>
```

```
123, Rue de l'Exemple <br> Ville Imaginaire <br> Pays Fictif
```

```
<br>
```

```
</address>
```

- **Numéro de téléphone** : L'élément `<a>` peut être utilisé avec le protocole `tel:` pour créer un lien de numérotation directe lorsque l'utilisateur clique sur le numéro.

```
<p>Numéro de téléphone : <a href="tel:+1234567890">123-456-
```

```
7890</a></p>
```

- **Adresse e-mail** : De manière similaire, l'élément `<a>` avec le protocole `mailto:` peut être utilisé pour créer un lien pour envoyer un e-mail.

```
<p>Adresse e-mail : <a
```

```
href="mailto:contact@example.com">contact@example.com</a></p>
```

- **Localisation sur la carte** : Si vous souhaitez afficher une carte pour indiquer une localisation, des services externes comme Google Maps peuvent être intégrés.

Vous pouvez également utiliser l'élément `<iframe>` pour incorporer une carte.

```
<iframe
```

```
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d12345.678
```

```
9!2d
```

```
-
```

```
1.2345678!3d12.3456789!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1
```

```
m2!1s0x0%3A
```

```
0x0!2zMTPCsDM5JzM0LjkiTiAxwrAzMicOMS43Ilc!5e0!3m2!1sen!2sus!4v123456
```

```
7890123!5m2!
```

```
1sen!2sus"
```

```
width="600" height="450" style="border:0;" allowfullscreen=""
```

```
loading="lazy">
```

```
</iframe>
```

L'utilisation de ces éléments HTML pour marquer les détails de contact rend non seulement votre contenu plus structuré, mais offre également des avantages en termes d'accessibilité et d'expérience utilisateur.

Vous pouvez tester et modifier ces code en utilisant [CodePen](#).

Balises des heures et dates

Les balises d'heure et de date en HTML sont utilisées pour marquer les informations temporelles dans le contenu d'une page web. Cela permet non seulement d'afficher les dates et heures de manière structurée, mais également d'améliorer l'accessibilité et l'interopérabilité des données. Voici comment utiliser ces balises avec des exemples :

- **Balise <time>** : L'élément <time> est utilisé pour représenter une date, une heure ou une plage horaire. Il peut inclure un attribut datetime pour indiquer la valeur temporelle en format ISO 8601.

```
<p>Mon anniversaire est le <time datetime="2000-08-21">21 août
```

```
2000</time>.</p>
```

- **Balise <date>** (expérimental) : L'élément <date> est une balise expérimentale qui peut être utilisée pour afficher une date en utilisant le format de l'interface utilisateur locale. Notez que sa prise en charge peut varier.

```
<p>Rendez-vous le <date>2023-08-21</date> à 15h00.</p>
```

- **Attribut datetime dans les balises <time>** : L'attribut datetime dans la balise <time> permet d'indiquer la valeur temporelle de manière explicite. Il peut être utilisé pour les dates, les heures ou les deux, selon le contexte.

```
<p>Le film commence à <time datetime="18:30">18h30</time> ce
```

```
soir.</p>
```

L'utilisation de ces balises temporelles permet non seulement d'afficher les informations de manière compréhensible pour les utilisateurs, mais elles sont également interprétables par les moteurs de recherche, les lecteurs d'écran et d'autres technologies, améliorant ainsi l'accessibilité et la sémantique de votre contenu.

Vous pouvez tester et modifier ces codes en utilisant [CodePen](#).

8 / 11

Page 8 sur 11.

Formatage avancé du texte

9 / 11

Page 9 sur 11.

Images en HTML

Les images en HTML

Insérer des images dans une page HTML est essentiel pour ajouter des éléments visuels à votre contenu. Voici comment procéder en utilisant des exemples :

L'élément `` est utilisé pour insérer des images dans une page HTML. Il ne nécessite pas de balise de fermeture et utilise l'attribut `src` pour spécifier l'URL de l'image.

```

```

Chemin d'accès à l'image :

L'attribut `src` doit contenir le chemin d'accès relatif ou absolu vers l'image.

Si l'image se trouve dans le même répertoire que la page HTML, vous pouvez simplement spécifier le nom du fichier. Sinon, fournissez le chemin complet ou relatif.

Si votre image est nommée "francophonie.png" et qu'elle réside dans le même répertoire que votre page HTML, voici comment vous pouvez l'intégrer en utilisant une URL relative :

```

```

Si cette image est située dans un sous-répertoire nommé "images" dans le même dossier que votre page HTML (ce que Google recommande pour des raisons de référencement et d'indexation), vous pouvez l'intégrer comme suit :

```

```

Une autre option est d'utiliser l'URL absolue de l'image, par exemple :

```

```

Il est préférable de garder vos images sur le même serveur que la page HTML pour une performance optimale.

Le code ci-dessous produira un résultat similaire à ce qui suit:

Texte alternatif

L'attribut `alt` dans la balise `` est utilisé pour fournir une description textuelle de l'image.

Cela est essentiel pour l'accessibilité, car il permet aux lecteurs d'écran de décrire l'image aux utilisateurs malvoyants.

Exemple : le code au-dessus pourrait être modifié de cette manière :

```

```

La manière la plus simple de tester votre texte `alt` est de mal épeler votre nom de fichier intentionnellement.

Si dans l'exemple, la photo est `francoffonie.jpg`, le navigateur ne l'afficherait pas mais afficherait le texte `alt` à la place :

L'attribut "alt" au sein de la balise `` sert à fournir une description textuelle de l'image en question. Cet aspect revêt une importance cruciale pour l'accessibilité, car il permet aux lecteurs d'écran de décrire l'image aux personnes souffrant de déficiences visuelles.

Une description précise et pertinente dans l'attribut "alt" permet aux utilisateurs de comprendre le contenu de l'image même en l'absence visuelle, contribuant ainsi à une expérience en ligne équitable et respectueuse de la diversité des utilisateurs.

Attributs additionnels :

D'autres attributs, tels que `width` (largeur) et `height` (hauteur), peuvent être utilisés pour spécifier les dimensions de l'image en pixels.

Cependant, il est recommandé d'utiliser ces attributs avec précaution pour éviter de déformer l'image.

```

```

Cela ne fait pas grande différence à l'affichage dans des circonstances normales.

Titre d'images

Vous pouvez ajouter un attribut `title` aux images, pour fournir un supplément d'information si nécessaire.

Dans notre exemple, nous pourrions faire ceci :

```

```

Cela donne une info-bulle avec le texte entré dans l'attribut `title`.

Légender des images avec `figure` et `figcaption`

Légender des images avec les balises `<figure>` et `<figcaption>`, plusieurs approches sont possibles pour les assortir à vos images.

Par exemple, vous pouvez opter pour la méthode suivante :

```
<div class="figure">
```

```

```

```
<p> La francophonie. </p>
```

```
</div>
```

Cependant, une solution plus élégante existe en utilisant les balises HTML5 `<figure>` et `<figcaption>`.

Ces balises ont été spécialement conçues à cet effet : fournir un conteneur sémantique pour les objets et lier clairement cet objet à sa légende.

Notre exemple précédent pourrait être réécrit de la manière suivante :

```
<figure>
```

```

```

```
<figcaption> La francophonie. </figcaption>
```

```
</figure>
```

L'élément `<figcaption>` indique au navigateur et aux technologies d'assistance que la légende décrit le contenu de l'élément `<figure>`.

Un élément `<figure>` n'est pas nécessairement lié à une image. Il constitue une unité de contenu indépendante qui :

- Manifeste votre préoccupation pour l'accessibilité et la facilité de compréhension.
- Peut être positionné à divers endroits dans un flux linéaire de page.
- Fournit des informations essentielles en soutien au contenu principal.

Cet élément peut contenir un ensemble d'images, des extraits de code, des fichiers audio, des vidéos, des équations, des tableaux ou encore bien d'autres éléments.

Vous avez 6 points sur un total de 6

6/6

9 / 11

Page 9 sur 11.

Images en HTML

10 / 11

Tableaux en HTML

Les tableaux en HTML

HTML met à disposition différents éléments accompagnés d'attributs spécifiques.

Cette fonctionnalité se révèle particulièrement adaptée pour exposer des informations sous forme de tableaux sur le web, tels que des emplois de temps, des données financières ou des états financiers.

Les tableaux offrent l'avantage d'une grande clarté. Les informations peuvent être aisément appréhendées grâce aux correspondances visuelles entre les titres des lignes et des colonnes.

Dans l'exemple ci-dessous, vous avez la possibilité de rechercher une planète géante gazeuse au sein du système jovien qui compte 62 lunes. Cette quête peut être accomplie en croisant les titres pertinents des lignes et des colonnes.

```
<table>
```

```
<tr>
```

```
<th>Spécialité</th>
```

```
<th>Nom du cours</th>
```

```
<th>Nombre d'apprenants</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Informatique</td>
```

```
<td>Python</td>
```

```
<td>22</td>
```

```
</tr>
```

Informatique
Java
112
Electrique
Système logique
27
Biologie
Génétique
13

Si mis en œuvre de manière appropriée, même les personnes ayant des problèmes de vision peuvent interpréter les informations tabulaires affichées dans un tableau HTML.

Un tableau HTML bien élaboré doit rendre les informations compréhensibles pour les utilisateurs ayant des limitations visuelles ou une vision restreinte.

Vous pouvez tester et modifier ce code en utilisant [CodePen](#).

Créer un premier tableau

Chaque tableau est délimité par les balises `<table>` `</table>`.

La plus petite unité d'un tableau est la cellule, qui est créée avec la balise `<td>` (signifiant "tableau de données"). Insérez ceci entre les balises du tableau :

```
<td>Bonjour, je suis votre première cellule.</td>
```

Pour obtenir quatre cellules, il suffit de copier la première trois fois.

```
<td>Bonjour, je suis votre première cellule</td>
```

```
<td>je suis votre deuxième cellule</td>
```

```
<td>je suis votre troisième cellule</td>
```

```
<td>je suis votre quatrième cellule</td>
```

Les cellules sont intrinsèquement disposées dans une ligne contiguë. Chaque balise `<td>` génère une cellule individuelle et, collectivement, elles constituent la première rangée.

En insérant plus de cellules, la rangée s'étend en longueur.

Afin d'empêcher cette extension horizontale et d'amorcer le positionnement des cellules suivantes sur une nouvelle ligne, il est nécessaire d'utiliser la balise `<tr>` (représentant "tableau rangée").

Plaçons les quatre cellules entre les balises `<tr>` :

```
<tr>
```

```
<td>Bonjour, je suis votre première cellule</td>
```

```
<td>je suis votre deuxième cellule</td>
```

```
<td>je suis votre troisième cellule</td>
```

```
<td>je suis votre quatrième cellule</td>
```

```
</tr>
```

Voici un tableau avec deux lignes et quatres colonnes

```
<table>
```

```
<tr>
```

```
<td>Bonjour, je suis votre première cellule</td>
```

```
<td>je suis votre deuxième cellule</td>
```

```
<td>je suis votre troisième cellule</td>
```

```
<td>je suis votre quatrième cellule</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Deuxième ligne, première cellule</td>
```

```
<td>Cellule 2.</td>
```

```
<td>Cellule 3.</td>
```

```
<td>Cellule 4.</td>
```

```
</tr>
```

```
</table>
```

Ajouter des en-têtes avec <th>

Les en-têtes de tableau sont des cellules particulières qui signalent le commencement d'une ligne ou d'une colonne.

Elles définissent la nature des données présentes dans cette ligne ou colonne.

```
<table>
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
<th>Knocky</th>
```

```
<th>Flor</th>
```

```
<th>Ella</th>
```

```
<th>Juan</th>
```

```
</tr>
```

```
<tr>
```

```
<th>Race</th>
```

```
<td>Jack Russell</td>
```

```
<td>Poodle</td>
```

```
<td>Streetdog</td>
```


<td>Cocker Spaniel</td>

</tr>

<tr>

<th>Age</th>

<td>16</td>

<td>9</td>

<td>10</td>

<td>5</td>

</tr>

<tr>

<th>Propriétaire</th>

<td>Belle-mère</td>

<td>Moi</td>

<td>Moi</td>

<td>Belle-sœur</td>

</tr>

<tr>

```
<th>Habitudes alimentaires</th>
```

```
<td>Mange tous les restes</td>
```

```
<td>Grignote la nourriture</td>
```

```
<td>Mange copieusement</td>
```

```
<td>Mange jusqu'à ce qu'il éclate</td>
```

```
</tr>
```

```
</table>
```

Pour attribuer une identité visuelle et sémantique aux en-têtes de tableau, nous utilisons la balise `<th>` ("table header").

Cette balise opère de manière analogue à `<td>`, mais elle marque spécifiquement un en-tête plutôt qu'une cellule ordinaire.

L'incorporation d'en-têtes dans un tableau offre divers avantages.

Ils simplifient la repérabilité des données recherchées, car les en-têtes sont clairement différenciés.

L'ensemble esthétique du tableau en bénéficie.

Les en-têtes de tableau possèdent un style par défaut : ils sont affichés en caractères gras et centrés, même si vous n'appliquez pas de style personnalisé pour les distinguer.

Un atout supplémentaire des en-têtes de tableau réside dans l'attribut `scope` (dont nous discuterons dans la section suivante).

Cet attribut accroît l'accessibilité des tableaux en reliant chaque en-tête à toutes les données des cellules d'une ligne ou d'une colonne.

Étendre des cellules sur plusieurs lignes ou colonnes

Au sein du langage HTML, l'extension des cellules sur plusieurs lignes ou colonnes au sein d'un tableau revêt une importance fondamentale pour la conception de mises en page sophistiquées et la structuration optimale des données.

Afin d'atteindre cet objectif, nous faisons appel aux attributs `rowspan` (fusion de lignes) et `colspan` (fusion

de colonnes), à insérer dans les balises <td> ou <th>.

Ces attributs autorisent la spécification du nombre de lignes ou de colonnes qu'une cellule doit englober.

Attribut rowspan pour la fusion de lignes :

L'attribut rowspan détermine la quantité de lignes que la cellule devra englober verticalement.

Par exemple, si vous souhaitez combiner une cellule sur deux lignes, vous insérez rowspan="2" dans la balise.

```
<tr>
```

```
<td rowspan="2">Cellule fusionnée</td>
```

```
<td>Ligne 1, Colonne 2</td>
```

```
<td>Ligne 1, Colonne 3</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Ligne 2, Colonne 2</td>
```

```
<td>Ligne 2, Colonne 3</td>
```

```
</tr>
```

Attribut colspan pour la fusion de colonnes :

L'attribut colspan détermine le nombre de colonnes que la cellule doit couvrir horizontalement.

Par illustration, pour regrouper une cellule sur trois colonnes, vous insérez colspan="3" dans la balise.

```
<tr>
```

```
<td colspan="3">Colonne 1</td>
```

```
<td colspan="3">Cellule fusionnée sur trois colonnes</td>
```

```
</tr>
```

Il convient de souligner que l'emploi de la fusion de lignes ou de colonnes doit être en accord avec la structure de votre tableau.

Un excès de fusions pourrait compliquer la compréhension du tableau et affecter sa clarté.

En maîtrisant habilement l'utilisation des attributs `rowspan` et `colspan`, vous pouvez élaborer des tableaux plus élaborés et mieux organisés, améliorant ainsi la mise en forme de vos données et l'expérience de navigation sur votre site web.

Illustrons ceci avec l'exemple suivant :

```
<table border=1>
```

```
<tr>
```

```
<th rowspan="2">Sites</th>
```

```
<th colspan="3">Performances</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Nombre visiteurs</td>
```

```
<td>Nombre de mises à jour</td>
```

```
</tr>
```

```
<tr>
```

```
<td>www.example.com</td>
```

```
<td>1320</td>
```

```
<td>5</td>
```

```
</tr>
```

```
<tr>
```

```
<td>www.test.com</td>
```

```
<td>876</td>
```

```
<td>16</td>
```

```
</tr>
```

```
</table>
```

Vous pouvez tester et modifier ce code en utilisant [CodePen](#).

Structuration d'un tableau

Les éléments `<thead>` et `<tbody>` sont employés afin de structurer les tableaux HTML en dissociant l'entête du tableau de son contenu principal.

Cette approche concourt à rehausser l'accessibilité et la clarté du contenu tabulaire, particulièrement lorsque le tableau est élaboré ou contient une grande quantité de données. Ci-dessous, vous trouverez des éclaircissements plus approfondis concernant chacun de ces éléments :

`<thead>` (Table Header) :

L'élément `<thead>` est déployé pour établir la section correspondant à l'en-tête du tableau. C'est à cet endroit que vous devriez positionner les lignes de table qui englobent les titres des colonnes. Ces titres sont généralement conçus pour décrire les données contenues dans chaque colonne du tableau. Par exemple, le tableau suivant présente les détails des produits, les en-têtes de chaque colonne pourraient être "Référence", "Nom", "disponibilité", etc.

L'élément `<thead>` doit être inséré à l'intérieur de la balise `<table>`. Voici un exemple illustrant l'utilisation de l'élément `<thead>` :

```
<table>

<thead>

<tr>

<th>Référence</th>

<th>Nom</th>

<th>Disponibilité</th>

</tr>

</thead>

<tbody>

<!-- Les données du tableau vont ici -->

</tbody>

</table>
```

`<tbody>` (Table Body) :

L'élément `<tbody>` est employé pour délimiter la partie correspondant au contenu central du tableau. C'est là que vous devriez insérer les lignes renfermant les données effectives du tableau. Ces informations peuvent différer d'une ligne à l'autre, mais elles s'alignent généralement sur le modèle instauré par les entêtes de colonnes situés dans la section `<thead>`.

Inclure les données dans un élément `<tbody>` distinct facilite la gestion et l'analyse des informations, tout en rendant la structure du tableau plus intelligible pour les utilisateurs de lecteurs d'écran.

Voici un exemple qui illustre l'usage de l'élément `<tbody>` :

```
<table>
```

```
<thead>
```

```
<!-- En-têtes de colonne -->
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td>A1-521</td>
```

```
<td>Produit A1</td>
```

```
<td>Oui</td>
```

```
</tr>
```

```
<tr>
```

```
<td>A2-224</td>
```

```
<td>Produit A2</td>
```

```
<td>Non</td>
```

```
</tr>
```

```
<!-- Autres lignes de données -->
```

```
</tbody>
```

```
</table>
```

En employant les balises `<thead>` et `<tbody>`, vous érigez une architecture limpide pour vos tableaux HTML. Cette approche simplifie la gestion, améliore l'accessibilité et favorise la compréhension du contenu, que ce soit pour les utilisateurs ou les moteurs de recherche.

Vous pouvez tester et modifier ces code en utilisant [CodePen](#).

Vous avez 6 points sur un total de 7

6/7

[Correction](#)[Voir la solution](#)

10 / 11

Page 10 sur 11.

Tableaux en HTML