

<HTML>

Balises	Description	Exemple
<!DOCTYPE>	Tout document HTML doit commencer avec une déclaration <!DOCTYPE>. Cette déclaration n'est pas une balise HTML, c'est une information pour le navigateur qui indique le type du document.	<!DOCTYPE html>
<html> </html>	<html> est la balise qui contient toutes les autres, à l'exception de la balise <!DOCTYPE>.	<html lang="fr-FR"> </html>
<head> </head>	La balise <head> fournit des informations générales (métadonnées) sur le document, incluant son titre, scripts, et feuilles de style. Il ne peut y avoir qu'un seul élément <head> dans un document HTML.	
<meta>	La balise <meta> représente toute information de métadonnées qui ne peut pas être représentée par un des autres éléments de métadonnées, tels que <base>, <link>, <script>, <style> ou <title>.	<meta charset="UTF-8">
<link>	La balise <link> définit la relation entre le document courant et une ressource externe. Cet élément peut être utilisé pour définir un lien vers une feuille de style, vers les icônes utilisées en barre de titre ou comme icône d'application sur les appareils mobiles.	<link rel="stylesheet" href="apple.css" >
<title> </title>	La balise <title> définit le titre du document (qui est affiché dans la barre de titre du navigateur ou dans l'onglet de la page). Il ne contient que du texte.	<title>Apple</title>
<body> </body>	La balise <body> représente le contenu principal du document HTML. Il ne peut y avoir qu'un élément <body> par document.	
<header> </header>	La balise <header> représente du contenu introductif, généralement un groupe d'éléments d'introduction ou d'aides à la navigation. Il peut contenir des éléments de titre mais aussi un logo, un formulaire de recherche et d'autres éléments.	
<main> </main>	La balise <main> représente le contenu principal du corps (<body>) d'un document. La zone de contenu principal se compose de contenu directement lié ou qui développe le sujet central d'un document, ou la fonctionnalité centrale d'une application.	
<footer> </footer>	La balise <footer> représente un pied de page pour sa section de contenu ou racine de sectionnement la plus proche. Un <footer> contient généralement des informations sur l'auteur-ice de la section, des données de droit d'auteur ou des liens vers des documents associés.	
<div> </div>	La balise <div> est le conteneur générique du contenu de flux. Il n'a aucun effet sur le contenu ou la mise en page tant qu'il n'est pas mis en forme d'une manière quelconque à l'aide de CSS.	<div class="classeName"> </div>
<a> 	La balise <a> crée un hyperlien vers des pages web, des fichiers, des adresses e-mail, des emplacements dans la même page ou toute autre ressource accessible par une URL.	 Tous les produits

	La balise permet d'intégrer une image dans un document.	
<menu> </menu>	La balise <menu> est une alternative sémantique à , mais il est traité par les navigateurs (et exposé dans l'arbre d'accessibilité) comme identique à . Il représente une liste non ordonnée d'éléments (représentés par des éléments).	
 	La balise représente une liste d'éléments sans ordre particulier. Il est souvent représenté par une liste à puces.	
 	La balise représente une liste ordonnée, généralement affichée sous forme de liste numérotée.	
 	La balise est utilisé pour représenter un élément dans une liste. Il doit être contenu dans un élément parent : une liste ordonnée (), une liste non ordonnée () ou un menu (<menu>).	
<h1> </h1>	Les balises <h1> à <h6> représentent les six niveaux de titre de section. <h1> correspond au niveau de section le plus haut et <h6> correspond au niveau le plus faible. Par défaut, tous les éléments de titre créent une boîte de niveau bloc dans la mise en page, commençant sur une nouvelle ligne et prenant toute la largeur disponible dans leur bloc conteneur.	<h1>Bienvenue !</h1>
<p> </p>	La balise <p> représente un paragraphe. Il est utilisé pour insérer du texte dans la page html.	<p>Apple est une entreprise ...</p>
<button> </button>	La balise <button> est un élément interactif qui peut être activé avec une souris, un clavier, un doigt, une commande vocale ou tout autre technologie d'assistance. Une fois activé, il peut déclencher une action tel qu'envoyer un formulaire ou ouvrir une boîte de dialogue.	<button id="btnPlay" onclick="main()">Play</button>
<form> </form>	La balise <form> représente une section du document contenant des contrôles interactifs permettant de soumettre des informations. Il permet par exemple la création de formulaire de connexion ou de création de compte.	<form class="formSign" id="formRegister" method="post">
<label> </label>	La balise <label> représente une légende pour un élément d'une interface utilisateur. Il est principalement utilisé dans les formulaires (balise <form>) pour donner un nom à un champ d'entrée (<input>).	<label for="firstName">Prénom</label>
<input> </input>	La balise <input> permet de créer des contrôles interactifs pour les formulaires web afin d'accepter des données de la part de l'utilisateur ; une grande variété de types de données d'entrée et de widgets de contrôle est disponible, selon l'appareil et agent utilisateur. L'élément <input> est l'un des éléments HTML les plus puissants et les plus complexes en raison du grand nombre de combinaisons possibles de types d'entrée et d'attributs.	<input type="text" id="inptFirstName" name="firstName" required="">
<table>	La balise <table> représente des données tabulaires, c'est-à-dire des informations	<table>

</table>	présentées dans un tableau à deux dimensions composé de lignes et de colonnes de cellules contenant des données.	<pre> <caption></caption> <thead> <tr> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> <tfoot> <tr> <th></th> <td></td> </tr> </tfoot> </table> </pre>
<caption> </caption>	La balise <caption> définit la légende (ou le titre) d'un tableau, fournissant à ce dernier un nom accessible ou une description accessible.	<caption> Titre du tableau </caption>
<thead> </thead>	La balise <thead> encapsule un ensemble de lignes de tableau (balise <tr>), indiquant qu'elles constituent l'en-tête d'un tableau contenant des informations sur les colonnes du tableau. Il s'agit généralement d'en-têtes de colonnes (éléments <th>).	
<tbody> </tbody>	La balise <tbody> permet de regrouper une ou plusieurs lignes du tableau (balise <tr>), indiquant qu'elles constituent le corps (principal) des données d'un tableau.	
<tfoot> </tfoot>	<tfoot> encapsule un ensemble de lignes de tableau (balise <tr>), indiquant qu'elles constituent le pied d'un tableau avec des informations sur les colonnes du tableau.	
<tr> </tr>	<tr> définit une ligne de cellules dans un tableau. Les cellules de la ligne peuvent ensuite être créées à l'aide d'une combinaison d'éléments <td> (cellule de données) et <th> (cellule d'en-tête).	
<th></th>	<th> définit une cellule comme l'en-tête d'un groupe de cellules de tableau et peut être utilisé comme enfant de l'élément <tr>.	<th> Titre de ligne ou de colonne </th>
<td></td>	<td> définit une cellule d'un tableau qui contient des données et peut être utilisé comme enfant de l'élément <tr>.	<td> Contenu de la cellule </td>
<script>	La balise <script> est utilisé pour intégrer du code ou des données exécutables : il sert	<script

<code></script></code>	généralement à intégrer ou référencer du code JavaScript. L'élément <code><script></code> peut aussi être utilisé avec d'autres langages, comme le langage de programmation GLSL de WebGL ou JSON.	<code>type="text/javascript" src="jsFileName.js" </script></code>
<code><style> </style></code>	La balise <code><style></code> contient des informations de mise en forme pour un document ou une partie d'un document. Il contient du CSS, qui est appliqué au contenu du document contenant l'élément <code><style></code> . Cette balise existe mais est à éviter autant que possible.	

<!DOCTYPE>

Principaux attributs	Utilité	Exemple
Pas de paramètre		

<html>

Principaux attributs	Utilité	Exemple
<code>lang=""</code>	Il est recommandé de toujours utiliser cet attribut. A destination des moteurs de recherches et navigateurs, il permet de déclarer la langue de la page web.	<code><html lang="fr-FR"> </html></code>

`<head> <body> <header> <main> <footer> <div> <menu> <h1> <p> <table> <thead> <tbody> <tfoot> <th> <td>`

Principaux attributs	Utilité	Exemple
<code>id</code>	Définit un identifiant (ID) qui doit être unique dans tout le document. Il sert à identifier l'élément lors de la création de liens, de scripts ou de mise en forme (avec le CSS).	<code><head id="idName"> </head></code>
<code>class</code>	Les classes permettent au CSS et à JavaScript de sélectionner et d'accéder à des éléments spécifiques. Il est possible d'associer une balise à plusieurs classes en les séparant par un espace.	<code><head class="className"> </head></code>
<code>style</code>	Contient les déclarations CSS à appliquer à l'élément. Cet attribut et l'élément HTML <code><style></code> ont principalement pour but de permettre une mise en forme rapide, par exemple pour des tests. Il est recommandé de définir les styles dans un ou plusieurs fichiers séparés.	<code><head style="color:red"> </head></code>

<meta>

Principaux attributs	Utilité	Exemple
charset	Si l'attribut charset est défini, l'élément <meta> est une déclaration d'encodage, indiquant l'encodage des caractères utilisé dans le document.	<meta charset="UTF-8">
name	Si l'attribut name est défini, l'élément <meta> fournit des métadonnées au niveau du document qui s'appliquent à toute la page.	<meta name="viewport" content="width=device-width, initial-scale=1.0">
content	Les attributs name et content peuvent être utilisés ensemble pour fournir des métadonnées au document sous forme de paires nom-valeur, l'attribut name donnant le nom de la métadonnée et l'attribut content sa valeur.	<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link>

Principaux attributs	Utilité	Exemple
href	Cet attribut définit l'URL de la ressource liée. L'URL utilisée peut être absolue ou relative.	<link rel="stylesheet" href="apple.css">
rel	Cet attribut indique la relation qui existe entre le document lié et le document courant.	<link rel="stylesheet" href="apple.css">

<title>

Principaux attributs	Utilité	Exemple
id	Définit un identifiant (ID) qui doit être unique dans tout le document. Il sert à identifier l'élément lors de la création de liens, de scripts ou de mise en forme (avec le CSS).	<title id="idName"> Apple </title>
class	Les classes permettent au CSS et à JavaScript de sélectionner et d'accéder à des éléments spécifiques. Il est possible d'associer une balise à plusieurs classes en les séparant par un espace.	<title class="className"> Apple </title>
style	Contient les déclarations CSS à appliquer à l'élément. Cet attribut et l'élément HTML <style> ont principalement pour but de permettre une mise en forme rapide, par exemple pour des tests. Il est recommandé de définir les styles dans un ou plusieurs fichiers séparés.	<title style="color : red">Apple </title>

<a>

Principaux attributs	Utilité	Exemple
href	Cet attribut définit l'URL de la ressource liée. L'URL utilisée peut être absolue ou relative.	<code> Tous les produits </code>
id	Définit un identifiant (ID) qui doit être unique dans tout le document. Il sert à identifier l'élément lors de la création de liens, de scripts ou de mise en forme (avec le CSS).	<code> Tous les produits </code>
class	Les classes permettent au CSS et à JavaScript de sélectionner et d'accéder à des éléments spécifiques. Il est possible d'associer une balise à plusieurs classes en les séparant par un espace.	<code> Tous les produits </code>
style	Contient les déclarations CSS à appliquer à l'élément. Cet attribut et l'élément HTML <code><style></code> ont principalement pour but de permettre une mise en forme rapide, par exemple pour des tests. Il est recommandé de définir les styles dans un ou plusieurs fichiers séparés.	<code> Tous les produits </code>

Principaux attributs	Utilité	Exemple
src	L'URL de l'image. Cet attribut est obligatoire.	<code></code>
alt	Définit une description textuelle alternative pour l'image.	<code></code>
id	Définit un identifiant (ID) qui doit être unique dans tout le document. Il sert à identifier l'élément lors de la création de liens, de scripts ou de mise en forme (avec le CSS).	<code></code>
class	Les classes permettent au CSS et à JavaScript de sélectionner et d'accéder à des éléments spécifiques. Il est possible d'associer une balise à plusieurs classes en les séparant par un espace.	<code></code>
style	Contient les déclarations CSS à appliquer à l'élément. Cet attribut et l'élément HTML <code><style></code> ont principalement pour but de permettre une mise en forme rapide, par exemple pour des tests. Il est recommandé de définir les styles dans un ou plusieurs fichiers séparés.	<code></code>

<button>

Principaux attributs	Utilité	Exemple
onclick	Indique l'action à faire lorsque l'on clique sur le bouton.	<pre><button onclick="functionName()"> Clique moi !!</button></pre>
name	Le nom du bouton, soumis en tant que paire avec la valeur (value) du bouton comme partie des données du formulaire.	<pre><button name="btnName"> Clique moi !!</button></pre>
value	Définit la valeur associée au name du bouton lorsqu'il est soumis avec les données du formulaire. Cette valeur est transmise au serveur en paramètres lorsque le formulaire est soumis.	<pre><button name="btnName" value="btnValue" > Clique moi !! </button></pre>
type	Le comportement par défaut du bouton. Les valeurs possibles sont : submit, reset, button.	<pre><button type="submit"> Clique moi !!</button></pre>
id	Définit un identifiant (ID) qui doit être unique dans tout le document. Il sert à identifier l'élément lors de la création de liens, de scripts ou de mise en forme (avec le CSS).	<pre><button id="idName"> Clique moi !!</button></pre>
class	Les classes permettent au CSS et à JavaScript de sélectionner et d'accéder à des éléments spécifiques. Il est possible d'associer une balise à plusieurs classes en les séparant par un espace.	<pre><button class="className"> Clique moi !!</button></pre>
style	Contient les déclarations CSS à appliquer à l'élément. Cet attribut et l'élément HTML <style> ont principalement pour but de permettre une mise en forme rapide, par exemple pour des tests. Il est recommandé de définir les styles dans un ou plusieurs fichiers séparés.	<pre><button style="color:red"> Clique moi !! </button></pre>

<form>

Principaux attributs	Utilité	Exemple
method	Cet attribut définit la méthode HTTP qui sera utilisée pour envoyer les données au serveur. C'est un attribut énuméré qui peut prendre les valeurs suivantes : post, get, dialog.	<code><form method="post"></form></code>
action	L'URL qui traite l'envoi du formulaire.	<code><form action="/action_page.php" method="post"> </form></code>
id	Définit un identifiant (ID) qui doit être unique dans tout le document. Il sert à identifier l'élément lors de la création de liens, de scripts ou de mise en forme (avec le CSS).	<code><form id="formRegister" method="post"></form></code>
class	Les classes permettent au CSS et à JavaScript de sélectionner et d'accéder à des éléments spécifiques. Il est possible d'associer une balise à plusieurs classes en les séparant par un espace.	<code><form class="formSign" method="post"></form></code>
style	Contient les déclarations CSS à appliquer à l'élément. Cet attribut et l'élément HTML <code><style></code> ont principalement pour but de permettre une mise en forme rapide, par exemple pour des tests. Il est recommandé de définir les styles dans un ou plusieurs fichiers séparés.	<code><form style="color:blue"></form></code>

<label>

Principaux attributs	Utilité	Exemple
for	La valeur correspond à l'identifiant du contrôle de formulaire étiquetable dans le même document, associant le <code><label></code> à ce contrôle de formulaire.	<code><label for="firstName"> Prénom </label></code>
id	Définit un identifiant (ID) qui doit être unique dans tout le document. Il sert à identifier l'élément lors de la création de liens, de scripts ou de mise en forme (avec le CSS).	<code><label for="firstName" id="idName"> Prénom </label></code>
class	Les classes permettent au CSS et à JavaScript de sélectionner et d'accéder à des éléments spécifiques. Il est possible d'associer une balise à plusieurs classes en les séparant par un espace.	<code><label for="firstName" class="className"> Prénom </label></code>
style	Contient les déclarations CSS à appliquer à l'élément. Cet attribut et l'élément HTML <code><style></code> ont principalement pour but de permettre une mise en forme rapide, par exemple pour des tests. Il est recommandé de définir les styles dans un ou plusieurs fichiers séparés.	<code><label style="color:blue"> Prénom </label></code>

<input>

Principaux attributs	Utilité	Exemple
type	Une chaîne de caractères qui indique le type de contrôle à afficher. On utilisera par exemple la valeur checkbox pour afficher une case à cocher. Si cet attribut est absent (ou qu'une valeur inconnue est utilisée), ce sera un champ de type text qui sera utilisé, permettant de saisir un texte dans le contrôle de formulaire.	<code><input type="text" name="firstName"></code>
name	Une chaîne de caractères qui fournit le nom associé au contrôle. Le nom est envoyé avec la valeur du contrôle lors de l'envoi du formulaire.	<code><input type="text" name="firstName"></code>
required	Un attribut booléen qui, lorsqu'il est présent, indique qu'une valeur doit être saisie avant que le formulaire puisse être envoyé. Cet attribut est pris en charge pour les types de contrôle text, search, url, tel, email, date, month, week, time, datetime-local, number, password, checkbox, radio, et file.	<code><input type="text" name="firstName" required></code>
id	Définit un identifiant (ID) qui doit être unique dans tout le document. Il sert à identifier l'élément lors de la création de liens, de scripts ou de mise en forme (avec le CSS).	<code><input type="text" id="idName" name="firstName"></code>
class	Les classes permettent au CSS et à JavaScript de sélectionner et d'accéder à des éléments spécifiques. Il est possible d'associer une balise à plusieurs classes en les séparant par un espace.	<code><input type="text" class="className" name="firstName"></code>
style	Contient les déclarations CSS à appliquer à l'élément. Cet attribut et l'élément HTML <code><style></code> ont principalement pour but de permettre une mise en forme rapide, par exemple pour des tests. Il est recommandé de définir les styles dans un ou plusieurs fichiers séparés.	<code><input style="color:blue"></code>

<script>

Principaux attributs	Utilité	Exemple
type	Cet attribut indique le type de script représenté.	<code><script type="text/javascript" src="game.js"> </script></code>
src	Cet attribut définit l'URI d'un script externe. Cela peut être utilisé pour insérer des scripts autrement qu'en les insérant à même le document.	<code><script type="text/javascript" src="game.js"> </script></code>

<style>

Principaux attributs	Utilité	Exemple
type	Cet attribut définit le langage de la feuille de style sous la forme d'un type MIME (le jeu de caractères ne doit pas être indiqué). Cet attribut est optionnel, la valeur par défaut est text/css.	<code><style type="text/css"></code>