



# HTML

DATE DE CRÉATION : 18 MARS 2024

AUTHOR: ABDELLAH MOUNAJAIK

EDITOR: M. YASIN OZTURK

AUTHOR: BENOIT LEPAGE

# Sommaire

---

Sommaire .....	1
1. HTML ? .....	2
2. Les Métadonnées dans la tête .....	2
3. Les balises.....	3
4. Une page html .....	5
5. Les <form>ulaires .....	6
6. Les <table>aux .....	16
Créer un tableau .....	16
7. La sémantiques .....	19

# 1. HTML ?

**HTML** signifie « *HyperText Markup Language* » qu'on peut traduire par « *langage de balises* pour l'hypertexte ».

Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. D'autres technologies sont utilisées avec HTML pour décrire la présentation d'une page ([CSS](#)) et/ou ses fonctionnalités interactives ([JavaScript](#)).

Hypertexte : Fonction permettant d'établir des liaisons directes entre éléments (texte, image...) de documents différents.

Le langage HTML tout entier repose sur l'utilisation d'éléments.

Si vous comprenez bien ce qu'est un élément, vous comprenez le HTML.

Aujourd'hui, il existe plus de 120 éléments HTML différents aux rôles très variés et qui font la richesse de ce langage.

## 2. LES MÉTADONNÉES DANS LA TÊTE

Les métadonnées sont implémentées seulement dans la balise `<head>`.

Elles contiennent des informations à propos de la page, elles sont essentielles à la bonne compréhension de la page par le navigateur.

Élément	Description
<u><code>&lt;base&gt;</code></u>	<p>L'élément <code>&lt;base&gt;</code> définit l'URL de base à utiliser pour recomposer toutes les URL relatives contenues dans un document. Il ne peut y avoir qu'un seul élément <code>&lt;base&gt;</code> au sein d'un document.</p> <p>Par ex : <code>&lt;base href="https://www.w3schools.com/" target="_blank"&gt;</code></p>
<u><code>&lt;head&gt;</code></u>	<p>Fournit des informations générales (métadonnées) sur le document, incluant <u>son titre</u> et des liens ou des définitions vers des <u>scripts</u> et <u>feuilles de style et toutes les ressources</u>.</p>
<u><code>&lt;link&gt;</code></u>	<p>Définit la relation entre le document courant et une ressource externe. Cet élément peut être utilisé pour définir un lien vers <u>une feuille de style</u>, vers les icônes utilisées en barre de titre ou comme icône d'application sur les appareils mobiles.</p>

**<meta>**

Représente toute information de métadonnées qui ne peut pas être représentée par un des éléments (<base>, <link>, <script>, <style> ou <title>)

**<style>**

Contient des informations de mise en forme pour un document ou une partie d'un document. Par défaut, les instructions de mise en forme écrites dans cet élément sont écrites en CSS.

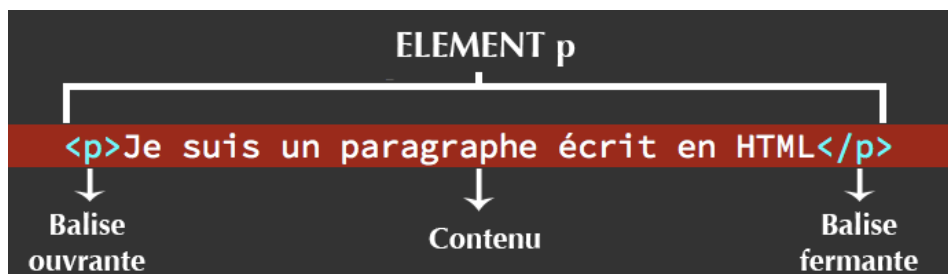
**<title>**

L'élément **<title>** définit le titre du document (qui est affiché dans la barre de titre du navigateur ou dans l'onglet de la page). Cet élément ne peut contenir que du texte, les balises qu'il contiendrait seraient ignorées.

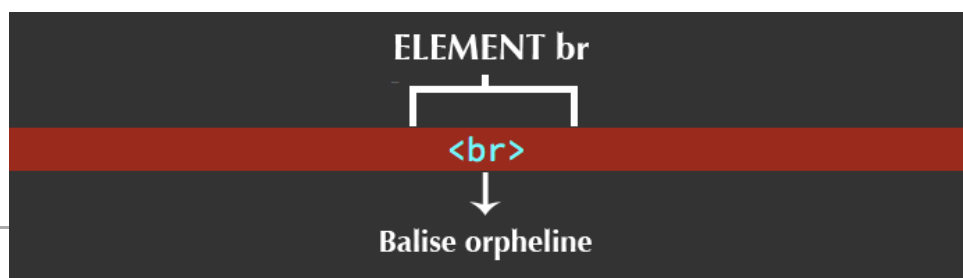
### 3. LES BALISES

Il y a trois termes dont vous devez absolument comprendre le sens en HTML.

Ce sont les termes élément, balise et attribut. Les éléments, tout d'abord, vont nous servir à définir des objets dans notre page. Grâce aux éléments, nous allons pouvoir définir un paragraphe (élément p), des titres d'importances diverses (éléments h1, h2, h3, h4, h5 et h6) ou un lien (élément a). Les éléments sont constitués de balises. Dans la majorité des cas, un élément est constitué d'une paire de balises : une balise **ouvrante** et une balise **fermante**.



Les balises reprennent le nom de l'élément et sont entourées de chevrons. La balise fermante possède en plus un slash qui précède le nom de l'élément.





Cependant, certains éléments ne sont constitués que d'une balise qu'on appelle alors balise auto-fermante. C'est par exemple le cas de l'élément `< br >` qui va nous servir à créer un retour à la ligne (Breaking Line)

### Quelques balises

<code>&lt;h1&gt;</code> à <code>&lt;h6&gt;</code>	structurez votre contenu avec des titres et des sous-titres. ... Les titres sont créés à l'aide des différents tags h. Plus le chiffre est petit, plus le titre est grand. On commence par le H1
<code>&lt;p&gt;</code>	Définis un paragraphe contenant une ou plusieurs phrases.
<code>&lt;hr/&gt;</code>	Trait horizontal
<code>&lt;br/&gt;</code>	Saut de ligne.
<code>&lt;ol&gt;</code>	Avec <code>&lt;ol&gt;</code> , une liste ordonnée est créée.
<code>&lt;ul&gt;</code>	Avec <code>&lt;ul&gt;</code> , on crée une liste non ordonnée mais avec des puces.
<code>&lt;li&gt;</code>	<code>&lt;li&gt;</code> est utilisé pour ajouter une entrée à la liste., list item
<code>&lt;b&gt;</code> , <code>&lt;strong&gt;</code>	<code>&lt;b&gt;</code> permet d'écrire un mot, une phrase ou même un paragraphe en gras.
<code>&lt;img /&gt;</code>	Le tag HTML <code>&lt;img&gt;</code> permet d'insérer une image.

---

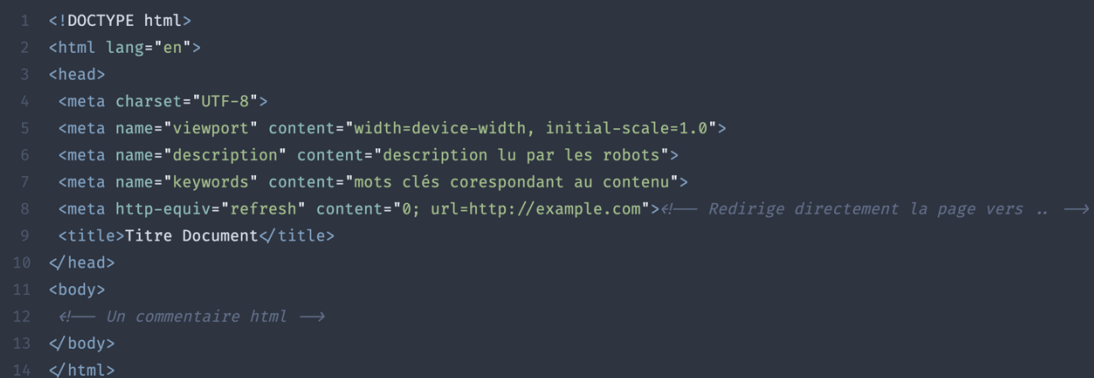
`<u>` La partie entourée de `<u>` est soulignée.

---

`<a>` Le lien hypertexte est déterminé avec `<a>`

---

## 4. ANATOMIE D'UNE PAGE HTML



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <meta name="description" content="description lu par les robots">
7 <meta name="keywords" content="mots clés corespondant au contenu">
8 <meta http-equiv="refresh" content="0; url=http://example.com"><!-- Redirige directement la page vers .. -->
9 <title>Titre Document</title>
10 </head>
11 <body>
12 <!-- Un commentaire html -->
13 </body>
14 </html>
```

L'élément `<html lang='fr'>` et `</html>` va contenir toute la page `<head>` et `<body>`.

L'élément `head` est un conteneur pour les balises `meta`, `title`, etc ... .

`<meta name="description" content="Free Web tutorials">`

Vital pour les moteurs de recherche, c'est une description souvent affichée sous le résultat de la requête .

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

Mots clés, représentant la page HTML

`<meta name="author" content="John Doe">`

Auteur.rice de la page

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

La balise `meta name="viewport"` va permettre de donner des instructions relatives à la taille et à l'échelle du viewport aux navigateurs mobiles afin que les différents éléments d'une page s'affichent au mieux.

Nous allons pouvoir passer plusieurs propriétés à l'attribut `content`.

Les propriétés `width` et `height` vont nous permettre de contrôler la taille du viewport dans lequel notre page doit s'afficher. On peut leur passer un nombre ou le mot clef `device-width` qui correspond à la taille de l'écran en pixels CSS à l'échelle 100%.

Ici, il me semble intéressant de définir ce qu'est « **un pixel CSS** ». Les pixels CSS correspondent à la surface utilisable de l'écran. Ce sont des pixels virtuels que l'appareil « pense » avoir. L'idée importante ici est qu'un pixel CSS n'est pas toujours égal à un pixel physique.

Les pixels physiques correspondent aux pixels réels qui composent un écran. C'est également ce qu'on appelle la définition d'un écran. Les écrans retina et haute définition possèdent généralement 4 fois plus de pixels réels que de pixels CSS.

La propriété `user-scalable` permet à l'utilisateur de zoomer dans la page (avec la valeur `yes`) ou, au contraire, lui interdit de la faire (avec la valeur `no`).

Cette propriété est souvent utilisée avec les propriétés **minimum-scale** et **maximum-scale** auxquelles on va pouvoir passer un nombre entre **0 et 10** et qui va représenter le niveau de dé-zoom ou de zoom que l'utilisateur est autorisé à faire.

Finalement, la propriété `initial-scale` permet de définir de niveau de zoom initial du viewport, c'est-à-dire son échelle. Nous allons également pouvoir lui passer un nombre entre 0 et 10.

L'encodage utilisé `<meta charset="utf-8">` indique l'encodage utilisé dans la page \*.html, cela détermine comment les caractères spéciaux s'affichent (accents, ligatures, idéogrammes chinois et japonais, etc.).

L'élément `<body>` contiendra tout le contenu de notre page, la partie 'visible' de la page HTML.

## 5. LES `<FORM>`ULAIRES

*L'élément HTML `<form>` représente un formulaire, c'est-à-dire une section d'un document qui contient des contrôles interactifs permettant à un utilisateur de fournir des informations.*

Pour créer un formulaire, vous aurez besoin de la balise HTML `<form>` `</form>`, elle permet d'indiquer le début et la fin du formulaire.

Un formulaire HTML avec deux champs de saisie et un bouton de soumission.

```
<form action="/action_page.php" method="get">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
```

```
<input type="submit" value="Submit">
</form>
```

Insérez des champs de texte avec la balise HTML `<input>`

La balise HTML `<input>` est une balise orpheline.

On la retrouvera plusieurs fois dans ce chapitre. Souvent, ce sera la valeur de son attribut, `type`, qui changera selon le type de champ que l'on veut ajouter : `text`, `email`, `date`, `submit`, ...

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

Ajoutez un champ de libellé avec la balise HTML `<label>` `</label>`

Un libellé est un champ de texte pré-rempli avec une indication pour guider le visiteur du site sur ce qu'il doit écrire. On crée un champ de libellé avec la balise HTML `<label>` `</label>` :

```
<label>Votre prénom</label> : <input type="text" name="prenom">
```

ou bien

```
<label>Votre prénom: <input type="text" name="prenom"></label>
```

Mais cela ne suffit pas : il faut lier le `label` à la zone de texte.

Pour ce faire, on doit donner un nom à la zone de texte, non pas avec l'attribut `name` mais avec l'attribut `id` (que l'on peut utiliser sur toutes les balises).

Pour lier le `label` au champ, il faut lui donner un attribut `for` qui a la même valeur que l' `id` du champ :

```
<form method="" action="">
  <label for="prenom">Votre prénom:
    <input type="text" name="prenom" id="prenom">
  </label>
</form>
```

On peut ajouter un certain nombre d'autres attributs à la balise `<input>` pour personnaliser son fonctionnement :

- agrandir le champ avec `size` ;
- limiter le nombre de caractères que l'on peut saisir avec `maxlength` ;
- pré-remplir le champ avec une valeur par défaut à l'aide de `value` ;
- donner une indication sur le contenu du champ avec `placeholder` . Cette indication disparaîtra dès que le visiteur aura cliqué à l'intérieur du champ.



Dans l'exemple suivant, la zone de texte contient une indication permettant de comprendre ce qu'il faut saisir ; le champ fait 30 caractères de long, mais on ne peut écrire que 20 caractères maximum à l'intérieur :

```
<form method="get" action="">
    <label for="prenom">Votre prénom :</label>
    <input type="text" name="prenom" id="prenom" placeholder="texte s'effaçant
au focus" size="30" maxlength="20">
</form>
```

Testez vous-même le résultat pour observer le comportement du champ. En attendant, voici le rendu du champ dans son état initial (avant que le visiteur ne renseigne quoi que ce soit) :

Ajoutez une zone de texte multi-ligne avec `<textarea>` `</textarea>`

Une zone de texte multi ligne est une zone de texte de plusieurs lignes. C'est utilisé dans le cas où le visiteur d'un site aurait besoin d'écrire un paragraphe de texte dans un formulaire.

Pour créer une zone de texte multiligne, on utilise `<textarea>` `</textarea>` .

Comme pour tout autre élément du formulaire, il faut lui donner un nom avec `name` et utiliser un `label` qui explique de quoi il s'agit :

```
<form method="get" action="">
<label for="prenom">Votre prénom :</label>
<input
  type="text"
  name="prenom"
  id="prenom"
  placeholder="Ex. : Mathieu"
  size="30"
  maxlength="20"
/>
</form>
```

Et voici le résultat en image :

Votre prénom :

Mais on peut modifier la taille par défaut du `<textarea>`, et ce, de deux façons différentes :

1. En appliquant au `<textarea>` les propriétés CSS `width` et `height` .
2. En ajoutant à la balise `<textarea>` les attributs :
  1. `rows` (indique le nombre de lignes qui peuvent être affichées) ;
  2. et `cols` (indique le nombre de colonnes qui peuvent être affichées).

Utilisez `<input>` pour des formats de saisie particuliers

Type de champs    Code

un e-mail            `<input type="email">`

une URL `<input type="url">`

On peut demander à saisir une adresse absolue, commençant généralement par `http://`

un numéro de téléphone `<input type="tel">`

un nombre entier        `<input type="number">`

Le champ s'affiche en général avec des petites flèches pour changer la valeur.

Pour les types de champs nombre, date et curseur, vous pouvez personnaliser le fonctionnement du champ avec les attributs suivants :

- `min` : valeur minimale autorisée ;
- `max` : valeur maximale autorisée ;
- `step` : c'est un "pas" de déplacement. Si vous indiquez un pas de 2, le champ n'acceptera que des valeurs de 2 en 2 (par exemple 0, 2, 4, 6...).

Laissez le visiteur choisir une option

HTML offre plusieurs manières de choisir une ou des option(s) dans un formulaire. Ce sont des éléments qui demandent au visiteur de faire un choix parmi une liste de possibilités, comme :

1. Les cases à cocher (plusieurs choix possibles).
2. Les boutons radio (un seul choix possible).
3. Les listes déroulantes (un seul choix possible).

Insérez des cases à cocher avec `<input type="checkbox">`

On crée une case à cocher avec la balise `<input>` en spécifiant le type `checkbox` qui signifie "case à cocher" en français :

`<input type="checkbox" name="choix">`

```

1  <form method="" action="">
2      Cochez les aliments que vous aimez manger :<br />
3      <input type="checkbox" name="frites" id="frites" />
4      <label for="frites">Frites</label><br />
5      <input type="checkbox" name="steak" id="steak" />
6      <label for="steak">Steak</label><br />
7      <input type="checkbox" name="epinards" id="epinards" />
8      <label for="epinards">Épinards</label><br />
9      <input type="checkbox" name="huitres" id="huitres" />
10     <label for="huitres">Huitres</label>
11 </form>

```

Ce qui donne, une liste d'aliments (le visiteur a la possibilité d'en cocher plusieurs) :

Vous pouvez faire en sorte qu'une case soit cochée par défaut, avec l'attribut `checked` :

```
<input type="checkbox" name="choix" checked>.
```

Insérez des boutons radio avec `<input type="radio">`

Les boutons radio permettent au visiteur de faire un choix (et un seul) parmi une liste de possibilités. Il y a une petite difficulté supplémentaire par rapport aux cases à cocher : il faut organiser les boutons radio en groupes.

Les options d'un même groupe possèdent le même nom, `name`, mais chaque option doit avoir une valeur (`value`) différente.

La balise à utiliser est toujours `<input>`, avec cette fois la valeur `radio` pour l'attribut `type` :

```
<form method="" action="">
```

```
    Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :<br>
```

```
    <input type="radio" name="age" value="moins15" id="moins15"> <label for="moins15">Moins de 15
ans</label><br>
```

```
    <input type="radio" name="age" value="medium15-25" id="medium15-25"> <label for="medium15-25">15-25 ans</label><br>
```

```
<input type="radio" name="age" value="medium25-40" id="medium25-40"> <label for="medium25-40">25-40 ans</label><br>
<input type="radio" name="age" value="plus40" id="plus40"> <label for="plus40">Plus de 40 ans</label>
</form>
```

Ce qui donne :

Pourquoi avoir mis le même nom pour chaque option ?

Tout simplement pour que le navigateur sache de quel groupe votre sélection fait partie. Essayez d'enlever les attributs name, vous verrez qu'il devient possible de sélectionner tous les éléments d'options. Or, ce n'est pas ce que l'on veut ; c'est pour cela qu'on les lie entre eux en leur donnant un nom identique.

Insérez une liste déroulante avec les balises <select> et <option>

Le fonctionnement est un peu différent :

1. On utilise la balise <select> </select> pour indiquer le début et la fin de la liste déroulante.
2. On ajoute l'attribut name à la balise pour donner un nom à la liste.
3. Puis, à l'intérieur du <select> </select>, on place plusieurs balises <option> </option> (une par choix possible), pour donner à chacune d'elles un attribut value afin d'identifier ce que le visiteur a choisi :

```
<form method="" action="">
  <label for="pays">Dans quel pays habitez-vous ?</label><br>
  <select name="pays" id="pays">
    <option value="espagne">Espagne</option>
    <option value="royaume-uni">Royaume-Uni</option>
    <option value="canada">Canada</option>
    <option value="japon">Japon</option>
  </select>
</form>
```

Si vous voulez qu'une option soit sélectionnée par défaut, utilisez l'attribut `selected` :

```
<option value="canada" selected>Canada</option>
```

Dans quel pays habitez-vous ?

Créez le bouton d'envoi du formulaire avec `<input>`

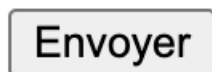
Pour créer le bouton d'envoi, c'est encore la balise `<input>` qu'on utilise.

`type="submit"` : c'est celui que vous utiliserez le plus souvent. Le visiteur sera conduit à la page indiquée dans l'attribut `action` du formulaire

On peut changer le texte affiché à l'intérieur des boutons avec l'attribut `value` :

```
<input type="submit" value="Envoyer">
```

Ce qui nous donne un input qui a l'apparence d'un bouton :



### action

L'URL qui traite l'envoi du formulaire. Cette valeur peut être remplacée par un attribut `formaction` sur un `<button>`, `<input type="submit">`, ou `<input type="image">`.

### enctype

Lorsque la valeur de l'attribut `method` est `post`, cet attribut définit le type MIME qui sera utilisé pour encoder les données envoyées au serveur. C'est un attribut énuméré qui peut prendre les valeurs suivantes :

- `application/x-www-form-urlencoded` : la valeur par défaut si l'attribut n'est pas défini
- `multipart/form-data` : la valeur utilisée par un élément `<input>` avec l'attribut `type="file"`.
- `text/plain`, correspondant au [type MIME](#) éponyme et utilisé à des fins de débogage.

Cette valeur peut être remplacée par un attribut [formenctype](#) sur un `<button>`, `<input type="submit">`, ou `<input type="image">`.

#### [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` : La méthode [POST](#) ; données du formulaire envoyées comme [corps de la requête](#).
- `get` : La méthode [GET](#) ; données du formulaire annexées à l'URL `action` avec un séparateur `?`. Utilisez cette méthode lorsque le formulaire [n'a pas d'effets secondaires](#).
- `dialog` : Lorsque le formulaire se trouve à l'intérieur d'un `<dialog>`, ferme la boîte de dialogue à la soumission.

Si cet attribut n'est pas défini, la valeur par défaut utilisée est `get`. Cette valeur peut-être remplacée par un attribut [formmethod](#) sur un `<button>`, `<input type="submit">`, ou `<input type="image">`.

#### [novalidate](#)

Cet attribut booléen indique si le formulaire doit être validé au moment de sa soumission. S'il n'est pas défini, le formulaire sera validé lors de sa soumission. Il peut être surchargé par l'attribut [formnovalidate](#) des éléments `<button>` ou `<input>` appartenant au formulaire.

#### [target](#)

Un nom ou un mot-clé indiquant où afficher la réponse après avoir envoyé le formulaire. Dans HTML 4, c'est le nom, ou le mot-clé, d'une frame. Dans HTML5, c'est le nom, ou le mot-clé, d'un *contexte de navigation* (onglet, fenêtre, frame). Les mots-clés suivants ont un sens particulier :

- `_self` (par défaut) : Charger dans le même contexte de navigation que le contexte actuel.
- `_blank`: Chargement dans un nouveau contexte de navigation sans nom.
- `_parent`: Charge dans le contexte de navigation parent de celui en cours. S'il n'y a pas de parent, se comporte de la même manière que `_self`.
- `_top`: Charger dans le contexte de navigation de niveau supérieur (c'est-à-dire le contexte de navigation qui est un ancêtre du contexte actuel et qui n'a pas de parent). S'il n'y a pas de parent, se comporte de la même manière que `_self`.

Cette valeur peut être remplacée par un attribut [`formtarget`](#) sur un `<button>`, `<input type="submit">`, ou `<input type="image">`.

Source: <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form>



## 6. LES <TABLE>AUX

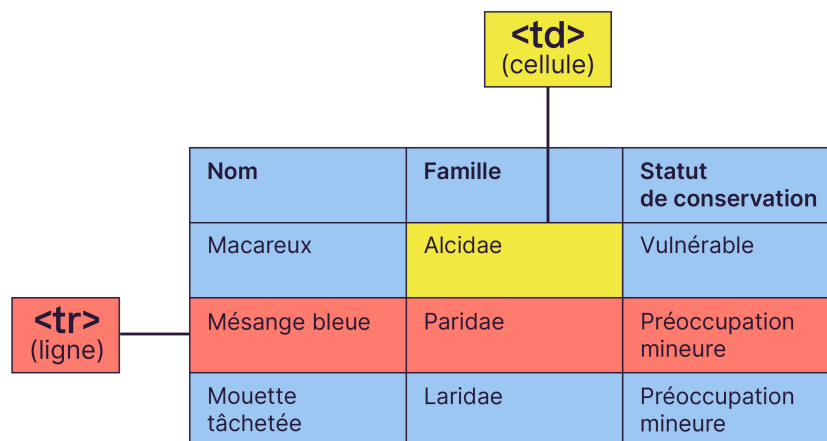
## 7. CRÉER UN TABLEAU

La balise en paire **<table>** **</table>** indique le début et la fin d'un tableau.

Pour construire un tableau ligne par ligne, on indique pour chaque ligne, **<tr>** , le contenu des différentes cellules, **<td>**.

**<tr>** **</tr>** indique le début et la fin d'une ligne du tableau ;

**<td>** **</td>** indique le début et la fin du contenu d'une cellule.



Nom	Famille		Statut de conservation
Macareux	Alcidae		Vulnérable
Mésange bleue	Paridae		Préoccupation mineure
Mouette tâchetée	Laridae		Préoccupation mineure

Ajoutez une ligne d'en-tête au tableau avec la balise HTML `<th>`

```

1 <table>
2   <tr>
3     <th>Nom</th>
4     <th>Âge</th>
5     <th>Pays</th>
6   </tr>
7   <tr>
8     <td>Carmen</td>
9     <td>33 ans</td>
10    <td>Espagne</td>
11  </tr>
12  <tr>
13    <td>Michelle</td>
14    <td>26 ans</td>
15    <td>États-Unis</td>
16  </tr>
17 </table>

```

La ligne d'en-tête se crée avec un `<tr>` mais les cellules qu'elle contient sont, cette fois, encadrées par des balises `<th>` (pour table header ou "en-tête de tableau" en français) et non pas `<td>` !

La ligne d'en-tête est très facile à reconnaître pour deux raisons :

les cellules sont des `<th>` au lieu des `<td>` habituels ;

c'est la première ligne du tableau (c'est idiot, mais encore faut-il le préciser).

Donnez un titre au tableau avec la balise HTML `<caption></caption>`

```

1 <table>
2   <caption>Passagers du vol 377</caption>
3   <tr>
4     <th>Nom</th>
5     <th>Âge</th>
6     <th>Pays</th>
7   </tr>
8   <tr>
9     <td>Carmen</td>
10    <td>33 ans</td>
11    <td>Espagne</td>
12  </tr>
13  <tr>
14    <td>Michelle</td>
15    <td>26 ans</td>
16    <td>États-Unis</td>
17  </tr>
18 </table>

```

Normalement, tout tableau doit avoir un titre. Le titre permet de renseigner rapidement le visiteur sur le contenu du tableau. Notre exemple est constitué d'une liste de personnes... oui, mais alors ? Qu'est-ce que cela représente ? Sans titre de tableau, vous le voyez, on est un peu perdu.

Heureusement, il y a `<caption>` pour ajouter un titre à notre tableau !

Cette balise se place tout au début du tableau, juste avant l'en-tête. C'est elle qui contient le titre du tableau .

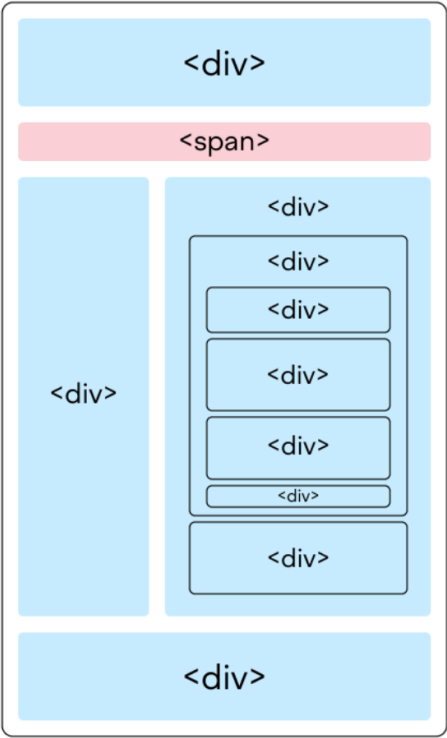
### Les balises TABLE

Element	What it is
<code>&lt;table&gt;</code>	The table itself
<code>&lt;caption&gt;</code>	The caption for the table. Like a figcaption to a figure.
<code>&lt;thead&gt;</code>	The table header
<code>&lt;tbody&gt;</code>	The table body
<code>&lt;tfoot&gt;</code>	The table footer
<code>&lt;tr&gt;</code>	A table row
<code>&lt;th&gt;</code>	A table cell that is a header
<code>&lt;td&gt;</code>	A table cell that is data
<code>&lt;col&gt;</code>	A column (a no-content element)
<code>&lt;colgroup&gt;</code>	A group of columns

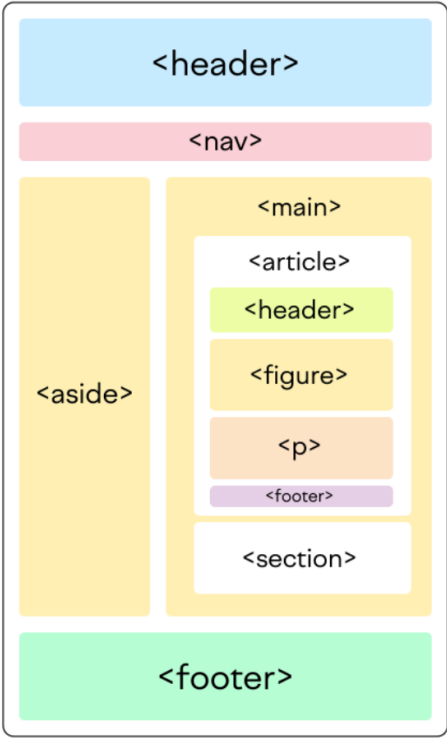
## 8. LA SÉMANTIQUES HTML5

Nom	Détails
<main>	Section principale du document HTML, contient la plus part des balises présentées ci-dessous.
<header>	Section d'introduction du document entier (en tête de page)
<nav>	Section possédant des liens de navigation principaux (au sein du document ou vers d'autres pages), elle est généralement situé dans le header.
<article>	Section déclarant que le contenu dans cette balise est un ou des articles de la page
<aside>	Barre latérale
<section>	Section générique regroupant un même sujet, une même fonctionnalité.
<footer>	Section de conclusion du document entier (pied de page).

HTML non sémantique



HTML sémantique



Add image format png, jpg ,...

Source : <https://css-tricks.com/complete-guide-table-element/>