

# TD Base HTML

Tags: HTML TD

## Utilisation de la documentation

La documentation dans le domaine du Web existe, nous n'allons donc pas ajouter une nouvelle documentation qui sera de toute façon moins complète que celle qui existe déjà.

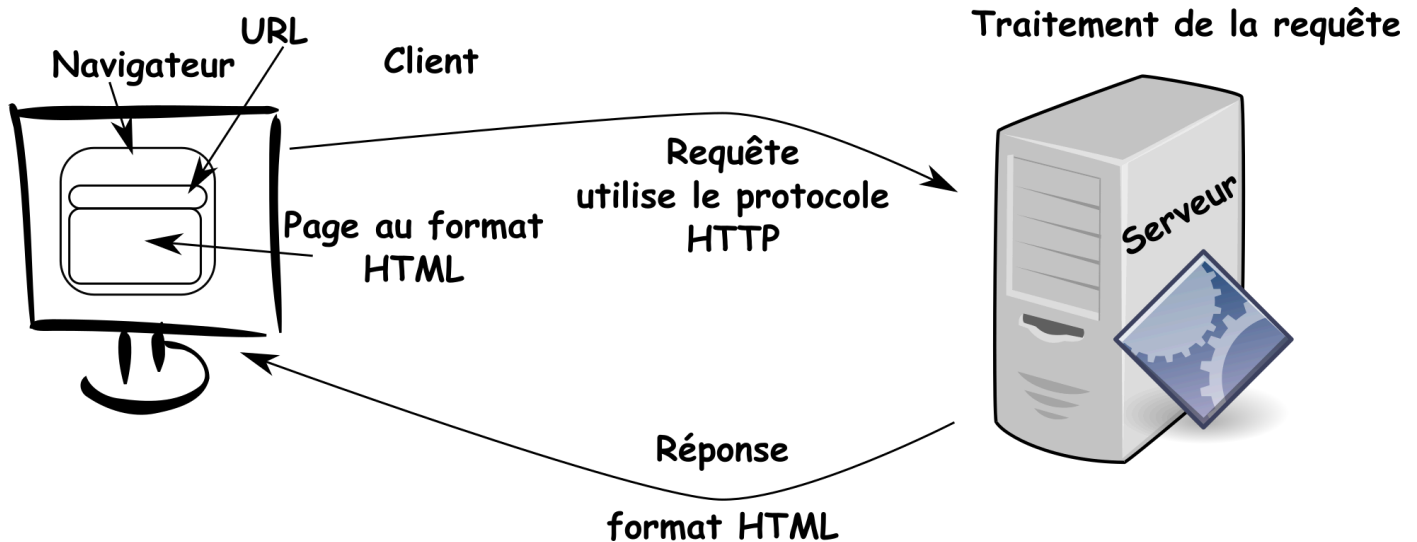
### A la fin de ce TD

- Vous saurez utiliser la documentation du Web côté client.
- Vous saurez ce qu'est une page web.

## Le WEB

Le WEB est une application réseau qui utilise le protocole [HTTP](#) (*Hypertext Transfer Protocol*) (couple application) au-dessus des protocoles TCP/IP. Le Web a été inventé par [Tim Berners-Lee](#) et [Robert Caillau](#) en mars 1989. Cette application WEB utilise un ensemble de pages de texte qui sont structurées à l'aide d'un langage de balises, le langage **HTML** (*Hypertext Markup Language*), des objets médias (image, son, vidéo) et des liens.

Une application Web est une application client-serveur, c'est-à-dire qu'un client, à partir d'un navigateur va faire une requête vers un serveur qui est en attente. A la réception de la requête, le serveur va l'analyser, la traiter et construire une réponse qu'il va renvoyer vers le client. Le protocole HTTP est utilisé pour le dialogue entre le client et le serveur. Le contenu de la réponse est formaté en utilisant le plus souvent le langage de balises HTML.



Un organisme, le [W3C](#) (World Wide Web Consortium) est créé en 1994 pour donner les règles, normes syntaxiques, sémantiques et évolution du langage HTML. Le W3C utilisera finalement les travaux proposés par un groupe de travail le [WhatWG](#) (*Web Hypertext Application Technology Working Group*) pour introduire le langage HTML 5.

Après de multiples évolutions ([histoire du WEB](#)), détours et querelles entre constructeurs et développeurs de navigateurs, le WEB utilise aujourd'hui le langage [HTML 5](#) pour exprimer la structure et la sémantique de la page. On associe avec le langage HTML 5, des Feuilles de style en cascade ou [CSS 3](#) (*Cascading Style Sheets*) pour définir le style, design du contenu de la page et le langage [javascript](#) pour ajouter de l'interactivité à la page.

## La documentation proposée sur le Web

- [Technologies web pour développeurs](#).
- Une [documentation](#) complète.

## Les outils de validation du contenu

- [Valideur officiel du W3C](#)
- [Valideur CSS du W3C](#)
- [Valideur HTML 5 expérimental](#)

## Constituant d'une page Web

Une page Web est écrite à l'aide du langage de balises HTML. Le langage de balises HTML5 tente de simplifier au maximum l'écriture d'une page.

La page minimale qui est acceptée par un vérificateur sera :

```
<!DOCTYPE html>
<title>Titre de la page</title>
```

Sans balise `html` , `head` et `body` .

## Une page de base

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Titre de la page</title>
</head>
<body>
<h1>Bonjour Monde!</h1>
</body>
</html>
```

## La notion d'encodage

Comme indiqué dans la section précédente, La page HTML est le résultat de la construction d'une réponse par un serveur. Il est important que cette réponse soit comprise par tous les clients quel que soit le système d'exploitation et quel que soit le navigateur. Pour cela il est nécessaire de préciser l'encodage des caractères utilisé.

L'[UTF-8](#) (*Universal Character Set Transformation Format - 8 bits*) est l'encodage le plus souvent utilisé. La balise meta suivante dans l'entête de la page HTML permet de préciser la valeur de l'encodage :

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

## Le format des fichiers échangés entre le client et le serveur : le type MIME

Les types [MIME](#) (*Multipurpose Internet Mail Extensions*) sont utilisés en HTTP pour affiner le dialogue entre le navigateur et le serveur web, et préciser quels sont les types de fichiers échangés. Une déclaration MIME est constituée de deux parties (normalisées) : un type et un sous-type, séparés par un caractère slash. Ceux débutant par `x-` ne sont pas standardisés, et les préfixes `vnd.` font référence à une propriété du vendeur.

Par exemple :

- text/html : pour un document HTML
- text/css : pour une feuille de style CSS

Dans le tableau suivant on trouvera quelques exemples de type MIME :

Type MIME	Usage
application/javascript	Script JavaScript (anciennement text/javascript)
application/octet-stream	Flux de données (type inconnu ou fichiers exécutables)
application/xhtml+xml	XHTML
application/x-shockwave-flash	Animation Adobe Flash
application/vnd.ms-excel	Fichier Excel (Microsoft)
application/msword	Fichier Word (Microsoft)
application/vnd.openxmlformats-officedocument.wordprocessingml.document	Fichier Word (Microsoft) de type Open XML
application/vnd.ms-powerpoint	Fichier Powerpoint (Microsoft)
application/vnd.oasis.opendocument.text	Fichier texte OpenDocument
application/vnd.oasis.opendocument.spreadsheet	Fichier de feuille de calcul OpenDocument
application/vnd.oasis.opendocument.presentation	Fichier présentation OpenDocument
audio/mpeg	Audio MPEG ou MP3
image/gif	Image GIF
image/jpeg	Image JPEG
image/png	Image PNG
image/svg+xml	Image SVG
text/css	Feuille de style CSS
text/html	HTML
text/plain	Texte simple

Type MIME	Usage
text/xml	XML
video/mpeg	Vidéo MPEG
video/mp4	Vidéo MP4
video/x-flv	Vidéo FLV Flash Video (Adobe)

Tableau tiré de [HTML 5 Une référence pour le développeur web](#) Rodolphe Rimelé

Cette information sera ajoutée et transmise à l'aide du protocole HTTP. Le client adaptera son comportement en fonction du type MIME indiqué par le serveur.

## Notion de balise

Une page en langage HTML se compose d'une suite de balises ou éléments. Voici un exemple de balise (tiré de la [documentation web](#)).

Il existe plusieurs types de balises :

- Les balises avec contenu par exemple `p` qui commence avec `<p>` , on parle de balise entrante et qui se termine avec `</p>` , on parle de balise fermante. La balise peut avoir un contenu comme dans la figure 2.
- Les balises sans contenu, on parle de balises auto-fermantes. Elles n'ont pas de contenu par exemple `hr` qui ajoute une ligne horizontale et que l'on note `<hr>` . La liste des balises qui répondent à cette syntaxe sont :

`area, base, br, col, command, embed, hr, img, input, keygen, link, meta, param,`

Avec HTML 5, chaque balise a une signification sémantique qui ajoute de l'information au document dans sa globalité. Par exemple la balise `strong` a pour contenu une donnée importante, la balise `small` a pour contenu des mentions légales. De nouvelles balises comme `section` , `article` , `nav` sont d'autres exemples de balises qui permettent d'ajouter de la sémantique au contenu de la page Web.



Figure tirée de [Les bases du HTML](#)

Certains éléments peuvent se succéder, une suite de paragraphes par exemple, d'autres éléments peuvent s'imbriquer et former une hiérarchie d'éléments, une suite de sections dans un article par exemple. D'autres encore doivent contenir d'autres éléments, les listes `ul`, `ol` par exemples.

Quand un élément se trouve imbriqué dans un autre, on parle d'une relation **parent**. Quand deux éléments se succèdent on parle de relation **frère**.

Dans le tableau qui suit, on donne une liste des éléments reconnus dans un document HTML 5.

Type	Rôle	Exemples
Métadonnées	Présentation ou comportement	<code>base</code> , <code>command</code> , <code>link</code> , <code>meta</code> , <code>noscript</code> , <code>script</code> , <code>style</code> , <code>title</code>
Flux	Contenu structuré	texte simple et éléments <code>a</code> , <code>abbr</code> , <code>address</code> , <code>area</code> , <code>article</code> , <code>aside</code> , <code>audio</code> , <code>b</code> , <code>bdi</code> , <code>bdo</code> , <code>blockquote</code> , <code>br</code> , <code>button</code> , <code>canvas</code> , <code>cite</code> , <code>code</code> , <code>command</code> , <code>datalist</code> , <code>del</code> , <code>details</code> , <code>dfn</code> , <code>div</code> , <code>dl</code> , <code>em</code> , <code>embed</code> , <code>fieldset</code> , <code>figure</code> , <code>footer</code> , <code>form</code> , <code>h1</code> , <code>h2</code> , <code>h3</code> , <code>h4</code> , <code>h5</code> , <code>h6</code> , <code>header</code> , <code>hgroup</code> , <code>hr</code> , <code>i</code> , <code>iframe</code> , <code>img</code> , <code>input</code> , <code>ins</code> , <code>kbd</code> , <code>keygen</code> , <code>label</code> , <code>map</code> , <code>mark</code> , <code>math</code> , <code>menu</code> , <code>meter</code> , <code>nav</code> , <code>noscript</code> , <code>object</code> , <code>ol</code> , <code>output</code> , <code>p</code> , <code>pre</code> , <code>progress</code> , <code>q</code> , <code>ruby</code> , <code>s</code> , <code>samp</code> , <code>script</code> , <code>section</code> , <code>select</code> , <code>small</code> , <code>span</code> , <code>strong</code> , <code>style</code> , <code>sub</code> , <code>sup</code> , <code>svg</code> , <code>table</code> , <code>textarea</code> , <code>time</code> , <code>ul</code> , <code>var</code> , <code>video</code> , <code>wbr</code>
Section	En-tête et pied de page	<code>article</code> , <code>aside</code> , <code>nav</code> , <code>section</code>
Titrage	Titres et sous-titres	<code>h1</code> , <code>h2</code> , <code>h3</code> , <code>h4</code> , <code>h5</code> , <code>h6</code> , <code>hgroup</code>
Texte	Texte du document	texte simple et éléments <code>a</code> , <code>abbr</code> , <code>area</code> , <code>audio</code> , <code>b</code> , <code>bdi</code> , <code>bdo</code> , <code>br</code> , <code>button</code> , <code>canvas</code> , <code>cite</code> , <code>code</code> , <code>command</code> , <code>datalist</code> , <code>del</code> , <code>dfn</code> , <code>em</code> , <code>embed</code> , <code>i</code> , <code>iframe</code> , <code>img</code> , <code>input</code> , <code>ins</code> , <code>kbd</code> , <code>keygen</code> , <code>label</code> , <code>map</code> , <code>mark</code> , <code>math</code> , <code>meter</code> , <code>noscript</code> , <code>object</code> , <code>output</code> , <code>progress</code> , <code>q</code> , <code>ruby</code> , <code>s</code> , <code>samp</code> , <code>script</code> , <code>select</code> , <code>small</code> , <code>span</code> , <code>strong</code> , <code>sub</code> , <code>sup</code> , <code>svg</code> , <code>textarea</code> , <code>time</code> , <code>var</code> , <code>video</code> , <code>wbr</code>
Contenu embarqué	Ressources externes à inclure dans le document	<code>audio</code> , <code>canvas</code> , <code>embed</code> , <code>iframe</code> , <code>img</code> , <code>object</code> , <code>video</code> , <code>svg</code>

Type	Rôle	Exemples
Contenu interactif	Destiné à l'interaction avec l'utilisateur	<code>a</code> , <code>audio</code> , <code>button</code> , <code>details</code> , <code>embed</code> , <code>iframe</code> , <code>img</code> , <code>input</code> , <code>keygen</code> , <code>label</code> , <code>menu</code> , <code>object</code> , <code>select</code> , <code>textarea</code> , <code>video</code>

Tableau tiré de [HTML 5 Une référence pour le développeur web](#) Rodolphe Rimelé

Le site [All the tag's](#) propose une liste des balises sous la forme d'un tableau périodique des éléments, également appelé tableau ou table de Mendeleïev. C'est un bon point de départ pour découvrir dans le détail chaque élément.

## Les attributs d'une balise

Les attributs sont des mots clés qui sont placés à la suite du nom de l'élément dans la balise ouvrante. Les attributs modifient les propriétés de l'élément. Un élément peut contenir 0 ou plusieurs attributs. Pour certains éléments, l'attribut est essentiel, par exemple `img` et son attribut `src`.

Les attributs que nous retrouverons souvent : `id` qui permet de référencer un élément dans un document et `class` qui permet de le lier avec des propriétés de style.

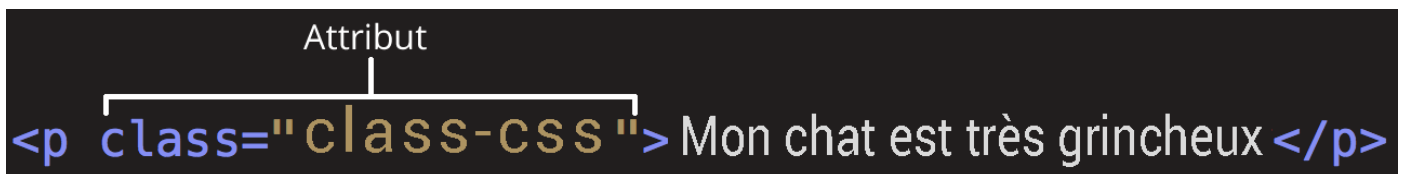


Figure tirée de [Les bases du HTML](#)

L'attribut est associé à une valeur, qui suit le symbole '='. Si la valeur contient un espace, il est obligatoire de protéger la valeur par des guillemets. D'une façon générale, il est recommandé d'utiliser des guillemets cela évite des erreurs et facilite la lecture.

Quand l'attribut est un booléen, sa seule présence indique la valeur vraie pour cet attribut, on pourra aussi écrire :

```
checked, checked=true, checked=1, checked=checked, checked=jevalidelentree
```

## Les commentaires

Il est possible de placer des commentaires dans votre document HTML 5. Ceux-ci ne seront pas affichés par le navigateur. Ils sont délimités par les chaînes de caractères `<!--` et `-->`.

Voici un exemple de commentaire :

```
<!-- Article sur l'utilisation du vélo en ville -->
<article>
```

```
<h1>Un titre</h1>
<section>Introduction</section>
</article>
```

## Les éléments du langage HTML 5

Je reprends ici une tentative de classification des éléments proposée dans le livre [HTML 5 Une référence pour le développeur web](#).

On distingue 3 types de modèles de contenu en plus du texte simple :

- les **éléments de flux**, qui structurent le document;
- les **éléments de phrasé**, qui sont en général imbriqués dans les précédents, et ajoutent une valeur sémantique particulière à leur propre contenu ou qui correspondent à des éléments médias ou de formulaire;
- les **éléments de métadonnées** (ou méta-informations), qui apportent des renseignements complémentaires au contenu ou agissent sur la présentation.

Catégorie	Éléments
Contenu de flux	a, p, hr, pre, ul, ol, dl, div, h1, h2, h3, h4, h5, h6, hgroup, address, blockquote, ins, del, object, map, noscript, section, nav, article, aside, header, footer, video, audio, figure, table, form, fieldset, menu, canvas, details ou éléments de phrasé ou texte
Contenu de phrasé	a, em, strong, small, mark, abbr, dfn, i, b, s, code, var, samp, kbd, sup, sub, q, cite, span, bdo, bdi, br, wbr, ins, del, img, embed, object, iframe, map, area, script, noscript, ruby, video, audio, input, textarea, select, button, label, output, datalist, keygen, progress, command, canvas, time, meter ou texte
Métadonnées	link, style, meta name, meta http-equiv, meta charset, script, noscript, command

Tableau tiré de [HTML 5 Une référence pour le développeur web](#) Rodolphe Rimelé

## La balise html

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <!-- L'en-tête du document -->
</head>
<body>
```

```
    <!-- Le corps du document -->
</body>
</html>
```

La balise `html` permet de spécifier le langage (attribut `lang`) utilisé dans le contenu du document et ce que doit stocker le navigateur pour pouvoir afficher la page en fonctionnement non-connecté (attribut `manifest`). Le contenu de l'élément `html` se résume à l'utilisation de 2 autres éléments : `head` et `body`.

## La balise `head`

L'entête est utilisé pour préciser des informations sur le document lui-même et les liens qu'il utilise.

Élément	Occurrences	Rôle
<code>&lt;title&gt;</code>	1	Titre du document
<code>&lt;meta&gt;</code>	0 ou +	Méta-informations
<code>&lt;link&gt;</code>	0 ou +	Relations vers des ressources externes (par exemple des feuilles de style CSS)
<code>&lt;style&gt;</code>	0 ou +	Styles CSS embarqués dans le document
<code>&lt;script&gt;</code>	0 ou +	Scripts embarqués dans le document
<code>&lt;base&gt;</code>	0 ou 1	Base d'adresses par défaut et de cible pour tous les liens du document

Tableau tiré de [HTML 5 Une référence pour le développeur web](#) Rodolphe Rimelé

Un exemple :

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <!-- Encodage des caractères -->
    <meta charset="utf-8">
    <meta name="author" content="Robert Duchmol">
    <meta name="description" content="Tout savoir sur le langage de balises HTML 5">
    <meta name="keywords" content="Web, HTML, HTML5">
    <meta name="generator" content="Mon éditeur HTML">
    <!-- Base générale des liens -->
    <base href="http://localhost:8000/">
    <!-- Titre du document -->
    <title>Ma page sur les kiwis</title>
    <!-- Une feuille de style -->
```

```
<link rel="stylesheet" href="css/screen.css" type="text/css" media="screen">
<!-- Des styles complémentaires -->
<style type="text/css">
  p{
    font-family:Sans-serif;
    text-align:center;
  }
</style>
<!-- Un fichier JavaScript externe -->
<script src="js/global.js"></script>
</head>
<body>
  <p>Ceci est ma première page HTML5.</p>
</body>
</html>
```

## Exercice 1

### Question 1.1

1. Étudiez en profondeur la balise `meta`.
2. Étudiez en profondeur la balise `link` et faites un inventaire des attributs avec une explication pour chacun.

## Les autres balises

### Question 1.2

1. Donnez la différence essentielle entre les balises de groupement `div` et `span`.
2. Étudiez en profondeur la balise `a` et faites un inventaire des attributs avec une explication pour chacun.
3. Étudiez en profondeur la balise `table` et faites un inventaire des balises qu'elle peut imbriquer.

## Les balises audio, vidéo, dessin

### Question 1.3

Étudiez les balises `audio`, `video` et `canvas`.

