

Hyper Text Markup Language

HTML

SOMMAIRE

• Introduction.....	4 - 6
◦ Qu'est-ce que le "World Wide Web" ?.....	4
◦ Qu'est-ce qu'une architecture Client/Serveur ?.....	4
◦ Qu'est-ce que le HTML ?.....	5
◦ À quelle version HTML somme-nous ?.....	5
◦ Utilise-t-on uniquement HTML pour les pages web ?.....	6
◦ Avec quoi écrit-on du texte HTML ?.....	6
• Architecture HTML.....	7 - 8
◦ Les balises.....	7
◦ Les attributs.....	7
◦ Minuscules ou majuscules ?.....	7
◦ Structure d'un document.....	7
• Les balises de premier niveau.....	9 - 9
◦ La balise <DOCTYPE>.....	9
◦ La balise <html>.....	9
◦ L'en-tête <head>.....	9
◦ Le corps <body>.....	9
◦ Les commentaires.....	9
• Les balises d'en-tête <head>.....	10 - 12
◦ Les balises <meta>.....	10
• Les balises de corps <body>.....	13 - 23
◦ Types de balise.....	13
◦ Les balises de titre.....	13
◦ Les balises de structuration de texte.....	13
◦ Les balises de mise en forme de texte.....	14
◦ Les balises de citation.....	15
◦ Les balises de mise en évidence.....	15
◦ Les balises génériques.....	15
◦ Les balises sémantiques.....	15
◦ La balise script.....	16
◦ Les balises de liste.....	16
◦ Les balises d'images et de médias.....	17
◦ Les balises de tableau.....	19
◦ Les balises de formulaire.....	19
◦ Les balises de dessins et d'interaction.....	23
◦ Les balises obsolètes.....	23

- **Aller plus loin**..... 24 - 25
 - Technologies liées à HTML5..... 24
- **Caractères accentués et spéciaux**..... 25 - 25
 - Les caractères accentués..... 25
 - Les caractères spéciaux..... 25
- **Résumé des balises**..... 26 - 28
- **Résumé des attributs**..... 29 - 33
- **Travaux pratiques**..... 34 - 34

Introduction

Qu'est-ce que le « World Wide Web » ?

Le World Wide Web (**WWW**) est un système de pages web publiques interconnectés à travers l'internet dont l'architecture a été proposé par Tim Berners-Lee. Ce dernier a créé les premiers serveurs web, navigateur web, et page web en 1990. Le web devient public qu'à partir de l'année 1991. Le web et l'internet ne sont pas la même chose, le web est l'une des nombreuses applications bâties au-dessus de l'internet.

Le web repose sur trois mécanismes pour rendre ses données facilement disponibles et le plus largement possible :

- Des protocoles, pour accéder à des ressources nommées dont on possède l'adresse (par exemple, **HTTP** – HyperText Transfer Protocol)
- Un système de nommage uniforme pour leur localisation sur le Web (par exemple : les **URL** – Uniform Ressource Location)
- L'hypertext, pour faciliter la navigation entre ses ressources (par exemple, **HTML** – HyperText Markup Language)

Lier, ou connecter des ressources grâce à des hyperliens, est un concept majeur qui a donné au web le statut de collection de documents connectés.

Toutes les technologies du web s'appuient sur des recommandations du World Wide Web Consortium (**W3C**), fondé également par Tim Berners-Lee, qui est un organisme de standardisation constitué des principaux groupes d'intérêt du web, comme des développeurs de navigateur web, des entités gouvernementales, des chercheurs et des universités. Sa mission inclut aussi l'éducation et la sensibilisation.

Qu'est-ce qu'une architecture Client/Serveur ?

L'architecture client/serveur désigne un mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs postes clients du serveur : chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique.

- **Caractéristiques d'un serveur :**
 - Il est passif (ou maître).
 - Il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients.
 - Dès qu'une requête lui parvient, il la traite et envoie une réponse.
- **Caractéristiques d'un client :**
 - Il est actif (ou esclave).
 - Il envoie des requêtes au serveur.
 - Il attend et reçoit les réponses du serveur.

Qu'est-ce que le « HTML » ?

Publier de l'information en vue d'une distribution mondiale nécessite l'usage d'un langage universellement compréhensible accepté par tous les ordinateurs. Le langage de publication utilisé sur le World Wide Web est le langage de balisage hypertext, le HTML.

HTML donne aux auteurs les moyens de :

- Publier des documents en ligne possédant des titres, du texte, des tableaux, des listes, des photos, etc.
- Ramener des informations en ligne via des liens hypertextes avec un clic.
- Concevoir des formulaires pour conduire des transactions avec des services distants, dans le but de rechercher des informations, d'acheter en ligne, de prendre des commandes, etc.
- Inclure des feuilles de calculs, des séquences vidéos ou sonores et d'autres applications, directement dans les documents.

C'est un langage de balisage, qui a comme objectif de représenter l'information. Ces balises sont interprétables par les différents navigateurs (Chrome, Safari, Opera, Firefox, Edge, ...) et permettent l'affichage de l'information désirée.

À quelle version HTML sommes-nous ?

Le langage HTML a beaucoup évolué depuis sa première version :

- **HTML 1.0** : 1992
 - 1992 : éléments Titres, Hyperliens, Structuration du texte en titre et sous-titres, liste ou texte brut, Mécanisme rudimentaire de recherche par index
 - 1993 : ajout des éléments img (uniquement gif et xbm), form input ... (formulaires)
 - 1994 : ajout de nombreux éléments de présentation
- **HTML 2.0** : 1995-1996
 - support des éléments table (tableaux)
 - 1995 : brouillon du HTML 3.0 finalement abandonné
- **HTML 3.2** : 1997
 - standardisation des éléments table et de nombreux éléments de présentation
 - préparation au support des éléments style et script
- **HTML 4.0** : 1999
 - standardisation de nombreuses extensions de style et script
 - ajout des éléments frames remplacé plus tard par élément object
 - amélioration de l'accessibilité du contenu (attributs ARIA)
- **XHTML 1.0** : 2000-2006
 - 2000 : Abandon du développement HTML au profit du XHTML
 - 2004 : Des éditeurs de navigateur web redonne une vie au HTML pour répondre aux nouveaux besoins, faisant concurrence au XHTML 2.0 en cours de développement
- **HTML 5** : 2007 à nos jours
 - 2007 : face aux réticences des développeurs pour le XHTML 2.0, le W3C relance le développement du HTML, premiers pas vers l'HTML 5.
 - 2009 : Abandon définitif du XHTML 2.0
 - 2014 : HTML 5 est officialisé, ajout des éléments sémantiques, abandon de certains éléments devenus inutiles

Utilise-t-on uniquement HTML pour les pages web ?

L'HTML est un langage de balisage, de structure, c'est un peu le squelette d'une page web. Il est actuellement dans sa cinquième version (HTML 5). Pour améliorer nos pages web d'autres langages ont été créés pour apporter plus de vie et de style :

- **CSS** : Cascading StyleSheet ou Feuilles de style en cascade, c'est le langage qui s'occupera de modifier l'apparence de la page. Actuellement le CSS est à sa troisième version (CSS 3).
- **JavaScript** : JavaScript (ou JS), à ne pas confondre avec Java, c'est un langage de script léger côté client, orienté objet, principalement connu comme le langage script des pages web. Ce langage apportera une meilleure interaction avec l'utilisateur. Il a une tendance à évoluer chaque année, sa dernière version à l'heure actuelle datant de juin 2022.
- **PHP** : Pre Hypertext Processor, c'est un langage de script côté serveur conçu pour le développement web mais aussi utilisé comme langage de programmation généraliste. C'est lui qui permettra de faire le pont entre vos pages et vos bases de données grâce au SQL. Tout comme le JavaScript il évolue régulièrement, sa prochaine version stable qui est actuellement encore en test est la version PHP 8.2.0 (Juillet 2022).
- **SQL** : Structured Query Language, c'est un langage normalisé pour créer, lire, mettre à jour ou supprimer des données (CRUD – Create Read Update Delete), issues d'une base de données via un système de requête (query). Il est également possible d'y faire quelques calculs. Cela fait un moment qu'il n'a pas subi de modification, en effet sa version actuelle date de 2011.

Avec quoi écrit-on du texte HTML ?

On utilise un **éditeur de texte** (Sublime Text, Atom, Brackets, NotePad++, jEdit, Visual Studio Code) pour créer un fichier ayant l'extension .html (par exemple : index.html).

Ce fichier peut être ouvert dans un **navigateur web** en faisant un double clic dessus.

À l'intérieur du fichier nous écrirons le contenu de notre page, accompagné de **balises HTML**

On peut afficher le **code source** de n'importe quelle page web de différentes manières :

- Soit en faisant un clic droit, puis en sélectionnant inspecter
- Soit en appuyant sur la touche F12
- La fenêtre apparaissant s'appelle un **outil de développement**, il est présent sur la plupart des navigateurs.

Architecture HTML

Les Balises

HTML est un langage de balises. Toutes les instructions sont exprimées dans des balises (ou tags en anglais), qui sont encadrées par des délimiteurs (signe inférieur < et signe supérieur >). Chaque balise doit être ouverte ou fermée (sauf exceptions). Les balises de fermeture sont les mêmes que celles d'ouverture, mais sont précédées d'un slash (/).

Exemple : `<balise> contenu de la balise </balise>`

Les attributs

Une balise peut accepter un ou plusieurs attributs qui se placent à l'intérieur de la balise d'ouverture. Ils sont séparés les uns par les autres par un espace ainsi que du nom de la balise. Un attribut comporte un nom et une valeur séparés par le signe égale, la valeur est quand à elle entouré par des doubles guillemets (attribut=«valeur»)

Exemple : `<balise attribut="valeur"> contenu de la balise </balise>`

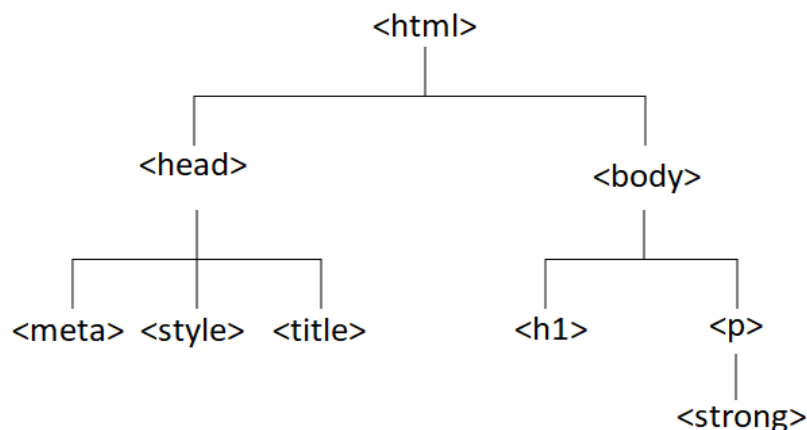
Minuscules ou majuscules ?

En HTML la casse (minuscules ou majuscules) n'a aucune importance, il est cependant **conseillé d'écrire les balises en minuscules**. De même, lorsque vous écrivez un document HTML, pensez à respecter l'indentation (espace ou tabulation) et veillez aux balises d'ouverture et de fermeture afin d'obtenir des documents lisibles.

Structure d'un document

Un document HTML se compose de plusieurs parties :

- Une ligne DOCTYPE (`<!DOCTYPE>`)
- Un élément html (`<html>`), qui comprend :
 - un élément head (`<head>`), une section entête déclarative qui comporte les éléments meta (`<meta>`), title (`<title>`) et link (`<link>`)
 - un élément body (`<body>`), qui comporte le contenu du document, ainsi que les script à la fin.



Voici un exemple de document HTML :

```
<!DOCTYPE html>
  <html lang="fr">
    <head>
      <meta charset="utf-8">
      <meta name="description" content="mots-clés">
      <link rel="stylesheet" content="monStyle.css">
      <title>Mon premier document HTML</title>
    </head>
    <body>
      <h1>Mon premier titre</h1>
      <p>Mon premier <strong>paragraphe</strong> !</p>
      <script src="monScript.js"></script>
    </body>
  </html>
```


Les balises de premier niveau

La balise <DOCTYPE>

Avant de commencer à écrire quoi que ce soit, votre document doit comporter une instruction « doctype » qui permet d'indiquer de quelle **version de HTML** il va s'agir.

- Avant l'HTML 5, il était nécessaire de bien informer la balise <!DOCTYPE>
Exemple : <!doctype html public "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
- Depuis l'HTML 5, vous serez ravis d'apprendre que tout ceci n'est plus nécessaire
Exemple : <!DOCTYPE html>

La balise <html>

Après avoir renseigné la version HTML, il faut indiquer **où commence le fichier** avec la balise d'ouverture <html> ainsi qu'**où se termine le fichier** avec la balise de fermeture </html>. Entre ses deux balises se trouveront vos balises <head> et <body>.

L'en-tête <head>

L'élément <head> contient les informations sur le document courant, tels que :

- Le **type d'encodage** de la page, <meta charset="utf-8">
- Des **mots-clés** que les moteurs de recherche peuvent exploiter et qui permettront d'être bien référencé, <meta name="description" content="mots-clés">
- Des définitions de **style**, <link rel="stylesheet" content="monStyle.css">
- Le **titre** <title> (nom d'onglet de page sur le navigateur), <title>Mon premier titre</title>
- etc ...

Le corps <body>

Pour l'instant le navigateur ne visualise rien, car les véritables données qu'il va afficher vont se trouver dans la balise <body>. Cette balise détermine le corps du document : c'est ici que vous allez intégrer des titres <h1>, des mots importants , des images , des tableaux <table>, des liens <a>, des formulaires <form>, ...

Les commentaires

Vous pouvez commenter votre code, cela peut s'avérer pratique si vous êtes plusieurs à travailler sur le même document.

Exemple : <!-- Ceci est un commentaire. -->

Les balises d'en-tête <head>

Les balises <meta>

Les balises meta fournissent des **meta-données** au navigateur, c'est-à-dire des informations au sujet de la page en question. Les balises meta sont normalement invisibles dans le navigateur de l'internaute. Elles sont nombreuses et presque aussi anciennes que le web. Certaines sont importantes pour le **référencement naturel** (SEO – Search Engine Optimization), d'autres sont devenues obsolètes ou n'ont aucun impact sur le référencement.

Pour que votre site soit en haut des listes de recherche, il faut préparer ses pages à l'arrivée des robots des **moteurs de recherche**. Car lorsqu'on soumet l'adresse de son site à un moteur, celui-ci l'enregistre puis envoie des « **robots** » (ou crawlers) se balader sur vos pages et enregistrer les informations contenues dans les **balises méta**. Grâce à ses balises vous pouvez commander les robots mais aussi influencer le navigateur.

La meta charset

La meta charset sert à spécifier le type d'encodage utilisé sur la page.

- **iso-8859-1** : définit ce qu'elle appelle l'alphabet latin numéro 1, qui consiste en 191 caractères de l'alphabet latin, chacun d'entre eux étant codé sur un octet (soit 8 bits). ISO 8859-1 reprend le codage des caractères imprimables d'US-ASCII. Implique d'utiliser des entités HTML dès que l'on souhaite insérer un caractère ne faisant pas partie des quelques 189 disponibles (´ = é, è = è, & = &), ne gère que les langues occidentales (latines) et est donc faiblement adaptable.
- **UTF-8** : Universal Character Set Transformation Format1 - 8 bits, c'est un codage de caractères informatiques conçu pour coder l'ensemble des caractères du « répertoire universel de caractères codés », aujourd'hui totalement compatible avec le standard Unicode, en restant compatible avec la norme ASCII limitée à l'anglais de base, mais très largement répandue depuis des décennies. C'est l'encodage le plus utilisé de nos jours.

Exemple : <meta charset="utf-8">

Les meta name

Les meta name permettent de décrire la page HTML, de donner des informations relative au contenu de votre site, de donner quelques mot-clés. Aujourd'hui les meta name « **description** » et « **robots** » et « **viewport** » restent les plus utilisées.

- **Meta name « description »**

Utilisé pour donner un léger descriptif du site ou de la page ainsi que quelques mots-clé, c'est cette petite phrase qui apparaît dans la liste des moteurs de recherche en dessous du lien vers votre site.

Exemple : <meta name="description" content="descriptif et mots-clés">

- **Meta name « robots »**

Utilisé pour donner des indications aux robots sur ce qu'ils doivent faire ou ne pas faire. Pour cela on lui indiquera s'il ne doit pas indexer notre page/site (noindex, nofollow) ou l'inverse (index, follow), vous pouvez également lui demander de réindexer au bout de n jours.

Exemple : <meta name="robots" content="noindex, nofollow">

- **Meta name « viewport »**

Le **viewport** du navigateur est la zone de la fenêtre dans laquelle le contenu web peut être vu. Souvent, cette zone n'a pas la même taille que la page rendue, auquel cas le navigateur fournit des barres de défilement pour que l'utilisateur et l'utilisatrice puissent faire défiler la page et accéder à tout le contenu.

Les appareils à écran étroit (par exemple, les mobiles) rendent les pages dans une fenêtre virtuelle. Par exemple, si l'écran d'un téléphone mobile a une largeur de 640 pixels, les pages peuvent être affichées dans une fenêtre virtuelle de 980 pixels, puis réduites pour tenir dans l'espace de 640 pixels.

De nombreuses pages ne sont pas optimisées pour les mobiles et ne fonctionnent pas (ou du moins n'ont pas l'air de fonctionner correctement) lorsqu'elles sont affichées sur un écran de petite taille. Cette fenêtre virtuelle est un moyen d'améliorer l'aspect des sites non optimisés pour les mobiles sur les appareils à écran étroit.

Cependant, ce mécanisme n'est pas aussi bon pour les pages qui sont optimisées pour les écrans étroits à l'aide de media queries (élément CSS qui permet d'adapter les éléments d'une page en fonction de la taille de l'écran) - si le viewport virtuel est de 980px par exemple, les media queries qui se déclenchent à 640px, 480px ou moins ne seront jamais utilisées, ce qui limite l'efficacité de ces techniques de responsive design. La meta name viewport a donc été créée pour aider les navigateurs à contrer ce problème.

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

- La propriété **width** contrôle la taille de la zone d'affichage. Elle peut être définie sur un nombre spécifique de pixels comme `width=600` ou sur la valeur spéciale **device-width**, qui est la largeur de l'écran en pixels CSS à une échelle de 100%. (Il existe des valeurs `height` et `device-height` correspondantes, qui peuvent être utiles pour les pages comportant des éléments qui changent de taille ou de position en fonction de la hauteur du viewport).
 - La propriété **initial-scale** contrôle le niveau de zoom lors du premier chargement de la page. Les propriétés `maximum-scale`, `minimum-scale` et `user-scalable` contrôlent la manière dont les utilisateurs à zoomer ou dézoomer la page.
- **Les meta name Obsolètes, Inutilisées ou Ignorées**
Il existe d'autres meta name qui sont devenus obsolètes ou qui ne sont plus utilisées voir ignorées par les moteurs de recherche, « **keywords** » (servait à renseigner des mot-clés), « **author** » (servait à donner des informations sur l'auteur du site ou de la page). Pour connaître la liste de toutes les balises meta acceptées par Google recherchez « meta charset google ».

Les meta de redirection et rafraîchissement

Il peut être parfois utile de recharger périodiquement une page surtout si l'on affiche des bannières publicitaires, afin d'en changer souvent. Il est ainsi possible d'ordonner au navigateur de recharger une page toutes les n secondes grâce à Refresh

Exemple : `<meta http-equiv="Refresh" content="n">`

On peut également rediriger le navigateur vers un autre site différent, très utile lorsqu'on change d'hébergeur.

Exemple : `<meta http-equiv="Refresh" content="n;urlDeRedirection">`

Les meta anti-cache

Il est possible d'interdire aux navigateurs de conserver en mémoire-cache vos pages.

Exemple : `<metta http-equiv="Pragma" content="no-cache">`

Certaines fois on va préférer que le cache du navigateur expire après une certaine durée.

Exemple : `<meta http-equiv="Expires" content="Mon, 5 Sep 2022 13:30:00">`

La balise <link>

La balise link permet de relier le document courant et une ressource externe (par exemple : style, favicon). La balise link accepte plusieurs attributs :

- **rel** : signifie relationship, qui correspond à la relation entre la ressource et le document courant. Il prendra différentes valeurs selon la ressource, **stylesheet** pour les Feuilles de style (CSS), **icon** pour l'icône de l'onglet de page, **preload** pour indiquer au navigateur de précharger la ressource,
- **href** : indique le chemin vers la ressource
- **media** : permet de définir en fonction de quel taille d'écran la balise s'active
- **sizes** : quand rel correspond à une icône, indique la taille de l'icône
- **type** : quand rel correspond à une icône, indique le type de fichier
- **as** : quand rel correspond à preload, définit la classe de contenu.

Exemples

`<link rel="stylesheet" href="cheminDuFichierCss.css" media="screen and (max-width : 600px)">`

`<link rel="icon" href="cheminImageFavicon" size="60x60" type="image/png">`

`<link rel="preload" href="maPoliceEcriture.woff" as="font" type="font/woff2">`

La balise <title>

La balise title contient une phrase reprise dans l'onglet du navigateur de l'internaute et les résultats de recherche du moteur. Avec la meta name description, elle fait partie des balise meta les plus importantes pour le référencement auprès des moteurs de recherche.

Exemple : `<title>Bienvenue sur premier site</title>`

Les balises de corps <body>

Types de balise

Il existe deux type de balise :

- Les balises dites « block », ces balises prendront toute la largeur de la page et forceront les éléments suivant à se mettre à la ligne, elles pourront également être redimensionné avec le CSS.
- Les balises dites « inline », ces balises prendront uniquement la largeur de leur contenu et se mettront les unes à la suite des autres jusqu'à ce qu'il n'y a plus de place, elles ne sont pas redimensionnable en CSS.
- Ces deux types pourront être modifiés plus tard selon le besoin avec le CSS.

Les balises de titre

Les balises de « titres » permettent de mettre en évidence des parties de texte. Plusieurs styles de titre sont prédéfinis, identifiés par les balises <h1> (le plus gros) à <h6> (le plus petit). Les balises titres sont de type « block ».

- <h1>Titre</h1> : Titre de niveau 1, un seul par page
- <h2>Titre</h2> : Titre de niveau 2
- <h3>Titre</h3> : Titre de niveau 3
- <h4>Titre</h4> : Titre de niveau 4
- <h5>Titre</h5> : Titre de niveau 5
- <h6>Titre</h6> : Titre de niveau 6

Les balises de structuration de texte

Dans un souci de lisibilité et de structuration, les documents doivent être découpés en paragraphes. Il existe trois balises en HTML qui permettent d'obtenir soit le début d'un paragraphe, soit un saut de ligne, soit effectuer une séparation entre deux textes.

- <p>Texte</p> : Paragraphe, type « block »
-
 : Retour à la ligne « type block »
- <hr> : Ligne de séparation horizontale, type « block »

Les balises de mise en forme de texte

- `Texte` : génère un lien vers une autre page, un site, une adresse mail, etc ... L'attribut **target** permet d'ouvrir le lien dans un nouvel onglet du navigateur, l'attribut **title** permet d'y ajouter un léger descriptif de son utilité, le texte entre les balises sera celui sur lequel on devra cliquer pour accéder au lien, type « inline »
 - Liens vers d'une autre page de votre site ou d'un autre site : on met le l'url de l'autre site ou le chemin de votre page dans l'attribut **href**.
 - Pour les e-mail, plusieurs possibilités pour l'attribut href :
 - lien simple :
 - href="mailto:adresse-mail@mail.com"
 - lien avec sujet :
 - href="mailto:adresse-mail@mail.com?subject=sujet"
 - lien avec personne en copie :
 - href="mailto:adresse-mail@mail.com?cc=autreemail@mail.com"
 - lien avec texte :
 - href="mailto:adresse-mail@mail.com?body=mon message"
 - lien avec sujet, personne en copie et message :
 - href="mailto:adresse-mail@mail.com?subject=_&cc=_&body=_"
- `<address>Texte</address>` : Adresse de contact, met en forme l'adresse, type « block »
- `<progress max="100" value="75"></progress>` : Barre de progression, on définit sa longueur avec les attribut min et max, type « block »
- `<abbr title="Texte">Sigle</abbr>` : Abréviation, montre ce qui est écrit dans l'attribut title, un léger soulignement en point apparaît sous le sigle, type « inline »
- `<dfn title="Definition">Mot</dfn>` : Définition, montre ce qui est écrit dans l'attribut title, type « inline »
- `<kbd>Touche</kbd>` : Pour afficher un format de touche de clavier, type « inline »
- `<pre>Texte</pre>` : Affichage formaté, type « block »
- `<time datetime="2022-09-05">5 septembre 2022</time>` : permet de représenter une période donnée. Cet élément permet d'utiliser l'attribut datetime afin de traduire la date ou l'instant dans un format informatique (permettant aux moteurs de recherche d'exploiter ces données ou de créer des rappels). Type « inline »
- `Texte` : Utilisé pour représenter du texte supprimé (exemple : ~~Texte~~), type « inline »
- `<ins>Texte</ins>` : Utilisé pour représenter du texte ajouté (exemple : Texte), type « inline »
- `^{Texte}` : Met le texte en exposant, type « inline »
- `_{Texte}` : Met le texte en indice, type « inline »

Les balises de citation

Dans certains document il va parfois être utile de faire des citations, pour cela trois balises sont à notre disposition :

- `<blockquote>Texte</blockquote>` : Citation (longue), type block
- `<cite>Texte</cite>` : Citation du titre d'une œuvre ou d'un événement, type inline
- `<q>Texte</q>` : Citation (courte), met entre « guillemets » le texte entre les balises, type inline

Les balises de mise en évidence

Ces balises ont plusieurs fonctions, la première est de changer l'apparence (gras, italique, surligné), l'autre de dire que le mot contenu entre ces balises est un mot important, elles sont donc utiles au référencement !

- `Texte` : Met en évidence une partie du texte (italique), type inline
- `Texte` : Met plus en évidence une partie du texte (gras), type inline
- `<mark>Texte</mark>` : Met en évidence une partie du texte (surligné), type inline

Les balises génériques

Ces balises n'ont aucun effet visuel, elles sont dites « génériques » car elles n'ont pas de fonction particulière, elles ne servent en réalité qu'à une chose, compartimenter du texte.

- `Texte` : Balise de type inline, sert à compartimenter une section d'un texte ou d'un paragraphe.
- `<div></div>` : Balise de type block, sert à compartimenter des portions de paragraphe, voir même compartimenter une page en plusieurs parties.

Les balises sémantiques

Depuis l'apparition du HTML 5 il existe de nouvelles balises de type block qui ont une importance cruciale, d'une part pour le référencement mais aussi pour l'accessibilité.

- `<header></header>` : Zone d'En-tête, du site ou même d'une `<section>`, `<main>`, `<article>`, `<aside>`, ...
- `<nav></nav>` : Liens principaux de navigation, contient généralement des liens `<a>` ou une liste ``
- `<footer></footer>` : Zone de Pied de page, du site ou même d'une `<section>`, `<main>`, `<article>`, `<aside>`, ...
- `<main></main>` : Contenu principale, contient les balises `<section>`, `<article>`, `<aside>`
- `<section></section>` : Section de page, contient les balises `<article>`, `<aside>`
- `<article></article>` : Article (contenu autonome)
- `<aside></aside>` : Informations complémentaires

La balise de script

Pour relier un document HTML à un document contenant du JavaScript ou TypeScript (JavaScript avec un typage strict), on utilise la balise script. Il existe deux manières de l'utiliser, soit en inscrivant directement le code JavaScript entre les balises (à éviter), soit en mettant directement le lien du fichier javascript (js) ou TypeScript (ts) dans l'attribut « src » (source). On précisera également de quel type de script il s'agit avec l'attribut « type ». À savoir, la balise script se met tout le temps à la fin de la balise <body> après tout votre code.

Première méthode : `<script>document.getElementById("monId") ;</script>`

Deuxième méthode : `<script type="text/javascript" src="fichier.js"></script>`

Les balises de liste

En HTML il est possible de réaliser trois catégories de liste, les listes **non ordonnées** , les listes **ordonnées** et les listes de **définition** <dl>. Chacune de ses listes aura ses propres « **items** » (chaque élément de la liste). Si l'on souhaite changer le type de puce/numér, on utilisera l'attribut « **type** ».

- **Listes non ordonnées et ordonnées**

- : Contient les balises , informe qu'il s'agit d'une liste à puces.
 - Attribut « **type** » :
 - **compact** : spécifie une liste compacte
 - **disc** : puce en forme de disque plein
 - **circle** : puce en forme de disque vide
 - **square** : puce en forme de carré plein
- : Contient les balises , informe qu'il s'agit d'une liste numérotée.
 - Attribut « **type** » :
 - **i** ou **I** : Chiffre romain minuscule ou majuscule
 - **a** ou **A** : Lettres minuscule ou majuscule
 - Attribut « **start** » : Permet de définir la valeur de départ
- Texte : Il s'agit d'une puce ou d'un numéro , c'est ici qu'on mettra le texte qui sera affiché

- **Liste de définitions**

- <dl></dl> : Contient les balise <dt> et <dd>, informe qu'il s'agit d'une liste de définition
 - Attribut « **type** » : **compact** (spécifie une liste compacte)
- <dt>Texte</dt> : Il s'agit du mot à définir
- <dd>Texte</dd> : Informe sur la définition du mot dans la balise <dt>

Les balises d'images et de médias

De nos jours il est très rare de voir des sites sans une petite image ou vidéo. Pour cela HTML nous met à disposition une multitude de petites balises qui viendront donner un peu plus de pep's à votre site.

- **Les images**

- `` : Permet d'insérer une image grâce à l'attribut **src** (source, lien url ou chemin d'un fichier dans votre ordinateur), doit toujours être à l'intérieur d'une autre balise (ex : `<p>`, `<figure>`, ...) et doit toujours avoir un attribut **alt** (alternatif), ce dernier sera affiché lorsque l'image ne peut être affichée ou sera lue par les lecteurs d'écran.
- `<figcaption>Texte</figcaption>` : Description de l'image, légende
- `<figure></figure>` : Englobe les balises `` et `<figcaption>`
- Il existe plusieurs formats d'image
 - **JPEG** : utilisé principalement pour les photos (Joint Photographic Expert Group)
 - **PNG** : utilisé pour toutes les images sans fond (Portable Network Graphics)
 - **GIF** : utilisé pour ajouter une petite animation (Graphic Interchange Format)
 - **SVG** : utilisé pour tout, le plus léger (Scalable Vector Graphics)

- **Les audios**

- `<audio>Texte</audio>` : Permet d'insérer un fichier audio, le texte entre les balises s'affiche uniquement si le son ne se lance pas.
- Pour sélectionner une source il existe deux possibilités :
 - Par l'attribut **src** (source, lien url ou chemin d'un fichier dans votre ordinateur) inscrit directement dans la balise audio
 - `<audio src="cheminAudio">Texte</audio>`
 - Par la balise `<source>` : lorsque l'on a des audio de différent format, prendra l'attribut **src** ainsi que l'attribut **type** pour préciser le format.
 - `<source src="cheminAudio" type="audio/format">`
- Les formats les plus répandus supportés par la majorité des navigateurs :
 - **WebM Vorbis** (type="audio/webm") : non supporté par Safari
 - **Ogg Vorbis** (type="audio/ogg") : non supporté par Safari
 - **MP3** (type="audio/mp3")
 - **WAVE** (type="audio/wave")
- Les autres attributs :
 - **muted** : Le son sera mis sur silence à l'ouverture de la page, il faudra créer une interaction pour que l'utilisateur puisse entendre le son.
 - **autoplay** : Le son se lira dès l'ouverture de la page (à éviter, sauf si muted est activé).
 - **controls** : par défaut les boutons de contrôle play/pause, etc ... ne sont pas affichés, on utilise cet attribut pour les faire apparaître.
 - **loop** : active la lecture en boucle.

- **Les vidéos**

- `<video>Texte</video>` : Permet d'insérer un fichier vidéo, le texte entre les balises s'affiche uniquement si le son ne se lance pas.
- Pour sélectionner une source c'est identique aux fichiers audio il existe deux possibilité :
 - Par l'attribut **src** (source, *lien url ou chemin d'un fichier dans votre ordinateur*) inscrit directement dans la balise video
 - `<video src="cheminVideo">Texte</video>`
 - Par la balise `<source>` : lorsque l'on a des audio de différent format, prendra l'attribut **src** ainsi que l'attribut **type** pour préciser le format.
 - `<source src="cheminVideo" type="video/format">`
- Pour définir des sous-titres on utilise la balise `<track>` qu'on insère entre les balises vidéos et auquel on relie un fichier au format WebVTT (Web Video Text Track)
`<track kind="subtitles" src="cheminSousTitre.vtt" srclang="fr" label="Français">`
- Les formats les plus répandus :
 - **WebM Vorbis** (type=«video/webm») : Opera, Firefox, Chrome
 - **Ogg Theora** (type=«video/ogg») : Opera, Firefox, Chrome
 - **Mp4** (type=«video/mp4») : Edge, Chrome, Safari
- Les autres attributs :
 - **muted** : Le son sera mis sur silence à l'ouverture de la page, il faudra créer une interaction pour que l'utilisateur puisse entendre le son.
 - **autoplay** : Le son se lira dès l'ouverture de la page (à éviter, sauf si muted est activé).
 - **controls** : par défaut les boutons de contrôle play/pause, etc ... ne sont pas affichés, on utilise cet attribut pour les faire apparaître.
 - **loop** : active la lecture en boucle.
 - **poster** : affiche une image lorsque la video n'est pas lancée (inutile si autoplay est activé)

Les balises de tableau

Les tableaux sont très largement utilisés dans le développement de pages web. Ils sont indispensables pour pouvoir présenter l'information de manière structurée et facilement lisible. Un tableau peut contenir du texte, des images, des vidéos, des formulaires, d'autres tableaux, ...

Tableaux simples

- `<table></table>` : Balises d'ouverture et de fermeture d'un tableau, elles englobent tout votre tableau
- `<caption>Titre</caption>` : Balises du titre d'un tableau, contenu entre les balises `<table>`, peut importe son emplacement, il sera toujours affiché en haut du tableau
- `<tr></tr>` : Balises délimitant une ligne d'un tableau
- `<th>Texte</th>` : Balises préformatées, utilisé pour les titres de colonne, contenu entre des balises `<tr>`
- `<td>Texte</td>` : Balises des cellules d'une ligne d'un tableau, contenu entre des balises `<tr>`

Tableaux volumineux

- `<thead></thead>` : Permet de délimiter l'en-tête d'un tableau, peut contenir `<tr>` et `<th>`
- `<tfoot></tfoot>` : Permet de délimiter le pied du tableau, peut contenir `<tr>`, `<td>`, `<th>`
- `<tbody></tbody>` : Permet de délimiter le corps du tableau, peut contenir `<tr>`, `<td>`, `<th>`
- Il est conseillé de les écrire dans cet ordre : `thead`, `tfoot` puis `tbody`

Attributs

- ***colspan*** : permet de fusionner des cellules de colonnes, `colspan=«2»`, balises `<td>` ou `<th>`
- ***rowspan*** : permet de fusionner des cellules de lignes, `rowspan= «2»`, balises `<td>` ou `<th>`

Les balises de formulaire

Les formulaires permettent à l'internaute de saisir des informations qui seront envoyées vers le serveur web afin d'y être traitées (souvent grâce à un programme tournant sur le serveur, script PHP par exemple). Tout échange de données entre le poste client et le serveur du site se fait via des saisies effectuées dans des formulaires.

Les saisies peuvent s'effectuer par des champs de saisie, des zones de texte, des listes de choix, des cases à cocher ou des boutons radio à sélectionner.

- `<form></form>` : Balises d'ouverture et de fermeture d'un formulaire, elles englobent tout votre formulaire.
 - Attribut ***method*** : indique la manière dont les données seront envoyées
 - ***get*** : assez peu adaptée car elle est limitée à 255 caractères. La particularité vient du fait que les informations seront envoyées dans l'adresse de la page
 - ***post*** : c'est la méthode la plus utilisée (99% du temps) pour les formulaires car elle permet d'envoyer un grand nombre d'informations. Les données saisies dans le formulaire ne transitent pas par la barre d'adresse

- Attribut **action** : adresse de la page qui va récupérer les données et les enregistrer, en général une page php mais aussi dans un mail ou un fichier txt
- Attribut **enctype** : il est utilisé pour spécifier que les données qui seront présentes dans le formulaire doivent être encodées lors de la soumission au serveur. Ce type d'attribut ne peut être utilisé qu'avec la méthode « POST ».
- **application/x-www-form-urlencoded** : C'est la valeur par défaut. Il encode tous les caractères avant de les envoyer au serveur. Il convertit les espaces en symboles + et le caractère spécial en sa valeur hexadécimale.
- **text/plain** : cette valeur convertit les espaces en symboles + mais les caractères spéciaux ne sont pas convertis.
- **multipart/form-data** : utilisé pour l'envoi de fichiers.

Les champs de saisie

- `<input>` : Champ de formulaire (texte, mot de passe, case à cocher, bouton radio, etc...)
- Attribut **id** : Obligatoire, permet de relier l'input à son label
- Attribut **name** : Obligatoire, sert lors de l'envoi du formulaire
- Attribut **type** : Définit de quel type de champ il s'agit
 - **text** : le visiteur devra saisir du texte
 - **number** : le visiteur devra saisir des chiffres, des flèches Haut et Bas apparaissent pour augmenter ou diminuer le chiffre
 - **range** : prend la forme d'un curseur
 - **password** : le visiteur devra saisir un mot de passe, les caractères sont cachés
 - **email** : le visiteur devra saisir un email
 - **url** : le visiteur devra saisir une adresse url
 - **tel** : le visiteur devra saisir un numéro de téléphone
 - **color** : le visiteur peut sélectionner une couleur
 - **date** : le visiteur peut sélectionner une date
 - **datetime-local** : le visiteur peut sélectionner une date et une heure
 - **month** : le visiteur peut sélectionner un mois et une année
 - **search** : indique uniquement à l'ordinateur qu'il s'agit d'un champ de recherche
 - **checkbox** : case à cocher (carré), le visiteur choisit une valeur parmi d'autres
 - **radio** : case de sélection (cercle), le visiteur choisit une valeur parmi d'autres

- **submit** : bouton d'envoi, généralement placé à la fin du formulaire, c'est lui qui envoie les données du formulaire
- **reset** : bouton de reset, réinitialise le formulaire en le vidant
- **image** : équivalent au submit, représente un bouton sous forme d'image
- **button** : aucun effet visuel, bouton simple
- **file** : le visiteur peut télécharger un fichier
- Attribut **minlength** : permet de limiter le nombre de caractères minimum saisissables
- Attribut **maxlength** : permet de limiter le nombre de caractères maximum saisissables
- Attribut **min** : permet de choisir un chiffre minimum (type number/range)
- Attribut **max** : permet de choisir un chiffre maximum (type number/range)
- Attribut **step** : permet de choisir de combien va s'incrémenter le chiffre (type number/range)
- Attribut **value** : permet de préremplir le champs
 - Pour les input de type submit/reset/image/button affiche la valeur sur le bouton
- Attribut **placeholder** : permet de donner un exemple de ce qui doit être saisi, apparaît grisé et disparaît dès qu'on saisi des données.
- Attribut **checked** : la case sera cochée par défaut (type checkbox/radio)
- Attribut **src** : indique la source d'une image (type image)
- Attribut **disabled** : empêche la saisie
- Attribut **readonly** : permet d'utiliser la zone de saisie comme affichage.
- `<textarea>Texte</textarea>` : Zone de saisie sur plusieurs lignes (agrandissable)
 - Attribut **id** : Obligatoire, permet de relier l'input à son label
 - Attribut **name** : Obligatoire, sert lors de l'envoi du formulaire
 - Attribut **minlength** : permet de limiter le nombre de caractères minimum saisissables
 - Attribut **maxlength** : permet de limiter le nombre de caractères maximum saisissables
 - Attribut **rows** : nombre de ligne en hauteur
 - Attribut **cols** : nombre de colonne en largeur
 - Attribut **disabled** : empêche la saisie
 - Attribut **readonly** : permet d'utiliser la zone de saisie comme affichage.

Les listes de sélection

- `<select></select>` : Balises de liste déroulante, contient les balises `<optgroup>` et `<option>`
 - Attribut **id** : Obligatoire, permet de relier l'input à son label
 - Attribut **name** : Obligatoire, sert lors de l'envoi du formulaire
 - Attribut **size** : Permet d'afficher plusieurs valeurs
 - Attribut **multiple** : Permet de sélectionner plusieurs choix
- `<optgroup></optgroup>` : Groupe d'option d'une liste déroulante, contient les balises `<option>`
 - Attribut **label** : Valeur affichée, nom d'un groupe d'option
- `<option>Texte</option>` : Élément d'une liste déroulante
 - Attribut **selected** : valeur sélectionné par défaut
 - Attribut **value** : valeur qui sera envoyé par lors de la validation du formulaire
- `<datalist id="...">` lié à l'input qui a un attribut **list** du même nom que l'id, entre les balises on utilise les balises `<option>` où cette fois la valeur et le texte entre les balises `<option>` est visible par l'utilisateur

Exemple :

```
<label for="idInput">
<input id="idInput" list="idDatalist" type="text">
<datalist id="idDatalist">
    Votre navigateur ne supporte pas Datalist en HTML5
    <option value="valeur">Valeur</option>
</datalist>
```

Le label

- `<label>Texte</label>` : Affiche ce que le visiteur doit saisir, se place généralement avant l'input/textarea/select sauf pour les input de type radio/checkbox où il se place après.
 - Attribut **for** : permet de relier le label à son champ de saisie via l'attribut **id**

Le bouton

- `<button>Texte</button>` : Bouton de formulaire
 - Attribut **type** : type de bouton
 - **submit** : par défaut, permet l'envoi du formulaire
 - **button** : simple bouton sans action spécifique

Les groupe de champ

- `<fieldset></fieldset>` : Permet de regrouper plusieurs champs sous une même section, contient la balise `<legend>` et peut contenir toutes les balises précédentes sauf `<button>`
- `<legend>Texte</legend>` : Titre d'un groupe de champ

Les balises de dessins et d'interaction

Les zones interactives

- La balise image `` possède un autre attribut permettant de la rendre interactive avec les balises `<map>`, c'est l'attribut ***usemap*** auquel on assigne pour valeur le nom de la map précédé d'un dièse #.
- `<map></map>` : Permet de créer des zones interactives sur une image par exemple.
 - Attribut ***name*** : on y inscrit généralement le nom de la map
- `<area>` : Contenu entre les balises `<map>`, sert à délimiter une zone sur la map
 - attribut ***shape*** :
 - `rect` : rectangle
 - `circle` : cercle
 - `default` : toute l'image
 - `poly` : polygone
 - attribut ***coords*** : permet de définir les coordonnées de la zone sur la map
 - `rect`
 - on délimite le point de départ en haut à gauche sur l'axe x et y
 - on délimite le point d'arrivée en bas à droite sur l'axe x et y
 - `circle`
 - on détermine le point centrale sur l'axe x et y
 - on détermine ensuite le diamètre
 - attribut ***href*** : génère un lien vers une autre page, site, etc ...
 - attribut ***target*** : permet de définir comment le lien va réagir
 - `_blank` : ouvre dans un nouvel onglet
 - attribut ***alt*** : pour définir un texte alternatif si l'image de la map ne s'affiche pas.

Exemple :

```

<map name="motherboard">
  <area shape="rect" coords="60,10,99,49" href="processor.html">
</map>
```

Les zones dynamiques

- `<canvas></canvas>` : Permet de créer et réaliser des formes dynamiques.

Les balises obsolètes

Il existe beaucoup de balises qui sont devenues obsolètes, par l'apparition d'autres balises plus efficaces suite à l'évolution du HTML, ou encore par l'apparition de nouveaux langages (CSS)

- `<small>` : Réduit la taille du texte
- `<big>` : Agrandit la taille du texte
- `<code>` : Pour afficher du code informatiques
- `<strike>` : Barre le texte
- `` : Spécifier une police
- ...

Aller plus loin

Technologies liées à HTML5

- **Canvas** : permet de dessiner au sein de la page web, à l'intérieur de la balise HTML `<canvas>` . On peut dessiner des formes (triangles, cercles...) mais aussi ajouter des images, les manipuler, appliquer des filtres graphiques... Au final, cela nous permet de réaliser aujourd'hui de véritables jeux et des applications graphiques directement dans des pages web !
- **SVG** : permet de créer des dessins vectoriels au sein des pages web. À la différence de Canvas, ces dessins peuvent être agrandis à l'infini (c'est le principe du vectoriel). Le logiciel Inkscape est connu pour permettre de dessiner des SVG.
- **Drag & Drop** : permet de faire « glisser-déposer » des objets dans la page web, de la même façon qu'on peut faire glisser-déposer des fichiers sur son bureau. Gmail l'utilise pour permettre d'ajouter facilement des pièces jointes à un e-mail.
- **File API** : permet d'accéder aux fichiers stockés sur la machine du visiteur (avec son autorisation). On l'utilisera notamment en combinaison avec le Drag & Drop.
- **Géolocalisation** : pour localiser le visiteur et lui proposer des services liés au lieu où il se trouve (ex. : les horaires des salles de cinéma proches). La localisation n'est pas toujours très précise, mais cela peut permettre de repérer un visiteur à quelques kilomètres près (avec son accord).
- **Web Storage** : permet de stocker un grand nombre d'informations sur la machine du visiteur. C'est une alternative, plus puissante, aux traditionnels cookies. Les informations sont hiérarchisées, comme dans une base de données.
- **AppCache** : permet de demander au navigateur de mettre en cache certains fichiers, qu'il ne cherchera alors plus à télécharger systématiquement. Très utile pour créer des applications web qui peuvent fonctionner même en mode « hors ligne » (déconnecté).
- **Web Sockets** : permet des échanges plus rapides, en temps réel, entre le navigateur du visiteur et le serveur qui gère le site web (c'est une sorte d'AJAX amélioré). C'est un peu l'avenir des applications web, qui pourront devenir aussi réactives que les vrais programmes.
- **WebGL** : permet d'introduire de la 3D dans les pages web, en utilisant le standard de la 3D OpenGL (figure suivante). Les scènes 3D sont directement gérées par la carte graphique.
- La plupart de ces technologies s'utilisent avec JavaScript. Il s'agit donc de nouvelles fonctionnalités que l'on peut utiliser en JavaScript.

Les sites web dynamiques (PHP, JEE, ASP .NET...)

- **PHP** : l'un des plus connus. Facile à utiliser et puissant, il est utilisé notamment par Facebook.
- **Java EE (Java)** : très utilisé dans le monde professionnel, il s'agit d'une extension du langage Java qui permet de réaliser des sites web dynamiques, puissants et robustes. Au début, il est un peu plus complexe à prendre en main que PHP.
- **ASP .NET (C#)** : assez semblable à JEE, c'est le langage de Microsoft. On l'utilise en combinaison avec d'autres technologies Microsoft (Windows Server...). Il utilise le puissant framework .NET, véritable couteau suisse des développeurs, qui offre de nombreuses fonctionnalités.
- **Django (Python)** : une extension du langage Python qui permet de réaliser rapidement et facilement des sites web dynamiques. Il est connu pour générer des interfaces d'administration prêtes à l'emploi.

Caractères accentués et spéciaux

Les caractères accentués

Voici la liste des principaux caractères accentués. Pour que l'accent soit reconnu correctement par tous les navigateurs, il faut le remplacer dans le texte par l'entité associée.

Caractère	Entité	Caractère	Entité	Caractère	Entité	Caractère	Entité
à	à	è	è	î	î	ù	ù
á	á	é	é	ï	ï	ú	ú
â	â	ê	ê	ñ	ñ	û	û
ã	ã	ë	ë	ö	ö	ü	ü
ä	ä	æ	æ	ô	ô	ç	ç

Les caractères spéciaux

Voici une liste de quelques caractères spéciaux. Pour que le caractère soit reconnu correctement par tous les navigateurs, il faut le remplacer dans le texte par l'entité associée.

Caractère	Code iso	Entité
©	©	©
®	®	®
™	™	™
€	€	€
¼	¼	¼
½	½	½
¾	¾	¾
“	“	“
”	”	”
„	„	„
‘	'	'
‘	‘	‘
’	’	’
,	‚	‚
~	˜	˜
,	¸	¸
•	•	•

Caractère	Code iso	Entité
§	§	§
£	£	£
¶	¶	¶
·	·	·
«	"	"
«	«	«
»	»	»
‹	‹	‹
›	›	›
¿	¿	¿
“	¨	¨
´	´	´
^	ˆ	ˆ
Espace	 	
...	…	
_	—	
-	–	

Résumé des balises

Balises de premier niveau

- `<!DOCTYPE html>` : Indique qu'il s'agit bien d'une page web HTML
- `<html></html>` : Indique où commence et où se termine la page web HTML
- `<head></head>` : Donne les indications sur la page, son titre, son encodage
- `<body></body>` : Contenu de la page
- `<!-- com -->` : Commentaires

Balises d'en-tête (entre `<head>`)

- `<meta />` : Méta-données de la page web (charset, mots-clés, etc.)
- `<link />` : Pour relier les ressources externes
- `<title></title>` : Ceci est le nom de la page

Balises de titre

- `<h1></h1>` : Titre de niveau 1, un seul par page
- `<h2></h2>` : Titre de niveau 2
- `<h3></h3>` : Titre de niveau 3
- `<h4></h4>` : Titre de niveau 4
- `<h5></h5>` : Titre de niveau 5
- `<h6></h6>` : Titre de niveau 6

Balises de structuration de texte

- `<p></p>` : Paragraphe
- `
` : Retour à la ligne
- `<hr>` : Ligne de séparation horizontale

Balises de mise en forme de texte

- `<a>lien` : Génère un lien vers une autre page ou un site, on met le lien dans l'attribut href, le texte entre les balises sera celui sur lequel on devra cliquer pour accéder au lien
- `<address></address>` : Adresse de contact, met en forme l'adresse
- `<progress></progress>` : Barre de progression
- `<abbr></abbr>` : Abréviation, montre ce qui est écrit dans l'attribut title
- `` : Texte supprimé (barré)
- `<ins></ins>` : Texte inséré (souligné)
- `<dfn></dfn>` : Définition
- `<kbd></kdb>` : Saisie clavier
- `<pre></pre>` : Affichage formaté
- `<time></time>` : Date ou heure
- `` : Met le texte en exposant
- `` : Met le texte en indice

Balises de citation

- `<blockquote></blockquote>` : Citation (longue)
- `<cite></cite>` : Citation du titre d'une œuvre ou d'un événement
- `<q></q>` : Citation (courte), met entre « guillemets » le texte entre les balises

Balises de mise en évidence

- `` : Met en évidence une partie du texte (italique)
- `` : Met plus en évidence une partie du texte (gras)
- `<mark></mark>` : Met en évidence une partie du texte (surligné)

Balises génériques

- `` : Balise de type inline, qui n'a aucun effet visuel mais qui permet de compartimenter du texte
- `<div></div>` : Balise de type block, qui n'a aucun effet visuel mais qui permet de compartimenter

Balises sémantiques

- `<header></header>` : Zone d'En-tête, du site ou même d'une `<section>`, `<main>`, `<article>`, `<aside>`
- `<nav></nav>` : Liens principaux de navigation, contient généralement des liens `<a>` ou une liste ``
- `<footer></footer>` : Zone de Pied de page, du site ou même d'une `<section>`, `<main>`, `<article>`, `<aside>`
- `<main></main>` : Contenu principale, contient les balises `<section>`, `<article>`, `<aside>`
- `<section></section>` : Section de page, contient les balises `<article>`, `<aside>`
- `<article></article>` : Article (contenu autonome)
- `<aside></aside>` : Informations complémentaires

Balise de script

- `<script></script>` : permet d'insérer les scripts, à mettre tout en bas de la balise `<body>`

Balises de liste

- `` : Contient les balise ``, informe qu'il s'agit d'une liste à puces.
- `` : Contient les balise ``, informe qu'il s'agit d'une liste numéroté
- `` : Il s'agit d'une puce `` ou d'un numéro ``
- `<dl></dl>` : Contient les balise `<dt></dt>` et `<dd></dd>`, informe qu'il s'agit d'une liste de définition
- `<dt></dt>` : Il s'agit du mot à définir
- `<dd></dd>` : Informe sur la définition du mot dans la balise `<dt>`

Balises d'images et de médias

- `` : Insérer une image, doit toujours être à l'intérieure d'une autre balise (ex : `<p>`, `<figure>`)
- `<figure></figure>` : Figure (image, code, etc.), contient la balise `` et `<figcaption></figcaption>`
- `<figcaption></figcaption>` : Description de la figure
- `<audio></audio>` : Pour ajouter un fichier audio, entre les balises afficher le texte si le son ne s'affiche/lance pas
- `<video></video>` : Pour ajouter un fichier vidéo, entre les balises afficher le texte si la vidéo ne se lance pas
- `<track>` utilisé avec la balise `<video>`, permet de définir les sous-titres possibles pour votre vidéo.
- `<source>` : Permet au navigateur de sélectionner le fichier qu'il sait lire `<audio>` et `<video>`

Balises de tableau

- `<table></table>` : Balises d'ouverture et fermeture d'un tableau
- `<caption></caption>` : Balises d'ouverture et fermeture du titre d'un tableau, contenu entre les balises `<table>`, peut importe son emplacement sera toujours affiché en haut du tableau
- `<tr></tr>` : Balises d'ouverture et fermeture d'une ligne d'un tableau, contenu entre les balises `<table>`
- `<th></th>` : Balises d'ouverture et fermeture d'en-tête d'un tableau, contenu entre les balises `<tr>`
- `<td></td>` : Balises d'ouverture et fermeture des cellules d'une ligne d'un tableau, contenu entre les balises `<tr>`
- `<thead></thead>` : Pour délimiter l'en-tête du tableau, contient `<tr>` et `<th>`
- `<tbody></tbody>` : Pour délimiter le corps du tableau, contient `<tr>` et `<td>`
- `<tfoot></tfoot>` : Pour délimiter le pied du tableau, contient `<tr>`, `<td>`, `<th>`

Balises de formulaire

- `<form></form>` : Balises d'ouverture et fermeture des formulaires
- `<fieldset></fieldset>` : Groupe de champs, contient la balise `<legend>` et toutes les balises qui suivent
- `<legend></legend>` : Titre d'un groupe de champs
- `<label>Texte</label>` : Affiche ce que le visiteur doit saisir, pour les input de type radio/checkbox se place après l'input
- `<input />` : Champ de formulaire (texte, mot de passe, case à cocher, bouton, etc.)
- `<textarea></textarea>` : Zone de saisie à plusieurs ligne, et agrandissable
- `<select></select>` : Balises d'ouverture et fermeture d'une liste déroulante, contient les balises `<optgroup>` et `<option>`
- `<optgroup></optgroup>` Groupe d'option d'une liste déroulante, contient la balise `<option>`
- `<option></option>` Élément d'une liste déroulante
- `<datalist>` lié à l'input qui a un attribut `list` du même nom que l'id, entre les balises on utilise les balises `option` où cette fois la value et le texte entre les balises `option` est visible par l'utilisateur
- `<button></button>` : un bouton

Balises de dessins et d'interaction

- `<map></map>` : Permet de créer des zones interactives avec la balise `<area>`
- `<area>` : Permet de délimiter une zone sur une `<map>`
- `<canvas></canvas>` : Permet de créer et réaliser des formes, des dessins.

Résumé des attributs

- **charset** : utilisé dans la balise <meta> et indique le type d'encodage (souvent UTF-8)
- **href** : utilisé dans la balise <a>
 - url : indique l'url vers laquelle il renvoi,
 - ancre : pour faire un lien vers une partie de la page, utiliser #nom d'attributs pour qu'il renvoi vers la balise contenant l'id du même nom, page.html#ancre redirige vers l'ancre de la page.html
 - mailto:adresse@gmail.com : génère un email avec l'adresse email de la personne
 - fichier.extension : pour télécharger un fichier (ne pas oublier l'extension)
 - Nom dossier/Nom page : Liens vers une page dans un dossier
 - ../Page : Liens vers une page situé dans un dossier parent
- **id** : permet de donner un nom unique à une balise, pour pouvoir s'y accrocher avec href.
- **title** : affiche une infobulle quand on met le curseur sur un lien, une image
- **target="_blank"** : Ouvre un lien dans une nouvelle fenêtre ou onglet
- **src** : utilisé dans la balise , <audio> et <video> indique la source du média
- **alt** : utilisé dans la balise décrit le contenu de l'image, sert de référencement de l'image, et pour les malvoyant
- **type** : utilisé dans les balises , , (<input> et <button>, voir formulaires)
 - : disc (rond plein), circle (rond vide), square (carré)
 - : 1 (chiffre, par défaut, 1 2 3), I (lettre, A B C)
- **start** : utilisé dans la balise permet de commencer au chiffre souhaité, sans débiter à 1.
- **lang** : utilisé dans la balise <html> pour le référencement (lang="fr")
- **tabindex** :
 - 0 à ... permet de rendre un élément non navigable, navigable
 - -1 permet de rendre un élément navigable, non navigable
- **rôle** : définir avec plus de précision l'utilité ou la fonction d'un élément ou d'un groupe d'éléments.
 - **banner** : indique qu'il s'agit de l'en-tête principale de la page, qui se répète sur chaque page
 - **search** : utilisé pour identifier une section de la page utilisée pour rechercher la page, le site ou la collection de sites.
 - **contentinfo** : indique qu'il s'agit du pied principale de la page, qui se répète sur chaque page
 - **navigation** : utilisé pour identifier les principaux groupes de liens utilisés pour naviguer dans un site Web ou le contenu d'une page.

- **tree** : pour les menus complexes, désigne l'arbre principale du menu
- **treeitem** : pour les menus complexes, désigne les sous menu de l'arbre principale
- **group** : pour les menus complexes, désigne les groupes de sous menu, ou les groupe de cellule pour les formulaire, combiner avec aria-labelledby pour ajouter plus de précision
- **dialog** : pour les modales, permet d'indiquer que le bloc est une fenêtre ou une boîte de dialogue, utilisé pour baliser une boîte de dialogue ou une fenêtre d'application HTML qui sépare le contenu ou l'interface utilisateur du reste de l'application Web ou de la page.
- **document** : indique aux technologies d'assistance avec des modes de lecture ou de navigation d'utiliser le mode document pour lire le contenu contenu dans cet élément.

Tableaux

- **colspan** : permet de fusionner des cellules de colonnes, colspan= « 2 », balises <td> ou <th>
- **rowspan** : permet de fusionner des cellules de lignes, rowspan= « 2 », balises <td> ou <th>
- **colgroup** : permet de définir un groupe de colonnes au sein d'un tableau, utilisé lorsqu'il y a des fusions (colspan)
- **rowgroup** : permet de définir un groupe de lignes au sein d'un tableau, utilisé lorsqu'il y a des fusions (rowspan)
- **scope** : permet de définir le sens de lecture des lignes composant notre tableau, utilisé dans les balises <th>
 - **row** : pour une lecture horizontal (ligne)
 - **col** : pour une lecture vertical (colonne)

Formulaires

- **autofocus** : utilisable une fois par page sur l'une des balises du formulaire, pour quelle soit focus par défaut en arrivant sur la page
- **required** : oblige le visiteur à remplir le champ avant d'envoyer le formulaire
- <form>
 - **method** : indique la manière dont les données seront envoyées
 - **get** : assez peu adaptée car elle est limitée à 255 caractères. La particularité vient du fait que les informations seront envoyées dans l'adresse de la page
 - **post** : c'est la méthode la plus utilisée (99% du temps) pour les formulaires car elle permet d'envoyer un grand nombre d'informations. Les données saisies dans le formulaire ne transitent pas par la barre d'adresse
 - **action** : adresse de la page qui va récupérer les données et les enregistrer, en général une page php mais aussi dans un mail ou un fichier txt

- `<input>`
 - **type**
 - **text** : le visiteur devra saisir du texte
 - **number** : le visiteur devra saisir des chiffres, des flèches Haut et Bas apparaissent pour augmenter ou diminuer le chiffre
 - **range** : prend la forme d'un curseur
 - **password** : le visiteur devra saisir un mot de passe, les caractères sont cachés (uniquement `<input>`)
 - **email** : le visiteur devra saisir un email
 - **url** : le visiteur devra saisir une adresse url
 - **tel** : le visiteur devra saisir un numéro de téléphone
 - **color** : le visiteur peut sélectionner une couleur
 - **date** : le visiteur peut sélectionner une date
 - **datetime-local** : le visiteur peut sélectionner une date et une heure
 - **month** : le visiteur peut sélectionner un mois et une année
 - **search** : indique uniquement à l'ordinateur qu'il s'agit d'un champ de recherche
 - **checkbox** : case à cocher (carré), le visiteur choisit une valeur parmi d'autres
 - **radio** : case de sélection (cercle), le visiteur choisit une valeur parmi d'autres
 - **submit** : bouton d'envoi, généralement placé à la fin du formulaire, c'est lui qui envoie les données du formulaire
 - **reset** : bouton de reset, réinitialise le formulaire en le vidant
 - **image** : équivalent au submit, représente un bouton sous forme d'image
 - **button** : aucun effet visuel, bouton simple, géré par javascript,
 - **file** : le visiteur peut télécharger un fichier
- `<input>` / `<textarea>`
 - **name** : permet de nommer une cellule afin de renseigner sur la catégorie des données saisies, le nom est unique comme l'id, chaque champ doit avoir un nom qui lui est propre
 - exception sur les input de type radio qui ont le même name afin de les mettre dans un groupe d'input du même nom, seuls leurs id/value les différencient
 - **id** : permet d'identifier la cellule, généralement identique à name et permet de le relier à l'attribut for de son label
 - **maxlength** : permet de limiter le nombre de caractères maximum saisissable (type text uniquement)
 - **minlength** : permet de limiter le nombre de caractères minimum saisissable (type text uniquement)
 - **min** : permet de choisir un chiffre minimum (type number/range uniquement)
 - **max** : permet de choisir un chiffre maximum (type number/range uniquement)
 - **step** : permet de choisir de combien va s'incrémenter le chiffre (de 2 en 2, ... type number/range uniquement)
 - **value** : permet de préremplir la cellule (uniquement pour `<input>`/`<radio>`/`<option>`)
 - pour les input de type submit/reset/image/button affiche la valeur sur le bouton

- **placeholder** : permet de donner un exemple de ce qui doit être saisi, apparaît grisé et disparaît dès qu'on saisie des données
- **checked** : la case sera cochée par défaut (uniquement pour type checkbox/radio, pas de valeur)
- **src** : indique l'url d'une image (uniquement pour input type image)
- <label>
 - **for** : est identique à l'id de l'input ou select pour le relier spécifiquement à cette input/select
- <textarea>
 - **rows** : nombre de ligne de hauteur (height)
 - **cols** : nombre de colonne de largeur (width)
- <select>
 - **size** : Permet d'afficher plusieurs valeurs
 - **multiple** : Permet de sélectionner plusieurs choix
- <optgroup>
 - **label** : Valeur affiché, nom d'un groupe d'option de <select>
- <option>
 - **selected** : valeur sélectionné par défaut (uniquement pour <option>)
- <button>
 - **type** : type de bouton
 - button : simple bouton
 - submit : bouton d'envoi de formulaire (obligatoirement entre les balises <form>)

Fichiers Audio & Vidéos

- <audio> & <video>
 - **src** : indiquer la source du fichier(<audio> ou <source> selon le cas)
 - **controls** : afficher les éléments de contrôle pour l'utilisateur (sans valeur)
 - **autoplay** : joue le fichier dès l'ouverture de la page (sans valeur, à éviter)
 - **muted** : son coupé
 - **loop** : joue le fichier en boucle
 - **preload** : indique si le fichier peut être préchargé dès le chargement de la page
 - **auto** : le navigateur décide de lui-même (par défaut, conseillé)
 - **metadata** : charge uniquement les métadonnées
 - **none** : pas de pré-chargement, réduit l'utilisation de la bande passante
- <video>
 - **poster** : Affiche une image tant que la vidéo n'a pas été lancée

Dessins et map interactive

- `<map>`
 - ***name*** : définir le nom de la map
- `<area>`
 - ***shape*** : permet de définir la forme de la zone
 - `rect` : rectangle
 - `circle` : cercle
 - `default` : toute l'image
 - `poly` : polygone
 - ***coords*** : permet de définir les coordonnées de la zone sur la map
 - ***href*** : génère un lien vers une autre page, site, etc ...
 - ***target*** : permet de définir comment le lien va réagir
 - `_blank` : ouvre dans un nouvel onglet
 - ***alt*** : pour définir un texte alternatif si l'image de la map ne s'affiche pas.
- `<canvas>`
 - ***width*** : largeur du canvas
 - ***height*** : hauteur du canvas

Travaux pratiques

1. Créer une première page web

- Objectif : Structure d'une page web minimal
- Balises utilisées : doctype, html, head, meta, title, body, h1 à h6, p, br, hr

2. Créer une page web

- Objectif : Comprendre les différentes balises de mise en forme
- Balises utilisées : div, span, strong, em, mark, abbr, dfn, address, progress, del, ins, sup, sub, kdb, pre, blockquote, cite, q

3. Créer des listes

- Objectif : listes non ordonnées, ordonnées, définitions, imbriquées
- Balises utilisées : ul, ol, li, dl, dt, dd

4. Créer des liens entre différentes pages

- Objectif : Relier des pages entre-elles
- Balises utilisées : a

5. Créer une page web sémantiques

- Objectif : Découvrir les nouveautés du HTML 5
- Balises utilisées : header, footer, nav, main, section, article, aside

6. Créer une page page avec des ressources externes

- Objectif : Relier des ressources externes (CSS, JavaScript)
- Balises utilisées : link, script

7. Créer des tableaux simples et complexes

- Objectif : tableaux simple, cellules fusionnées,
- Balises utilisées : table, caption, tr, th, td, thead, tfoot, tbody

8. Créer des formulaires simples et complexe

- Balises utilisées : form, label, input, textarea, select, button, fieldset, legend

9. Créer une page web avec des images et des médias

- Balises utilisées : img, audio, video, source, track

10. Créer une image interactive

- Balises utilisées : img, map, area