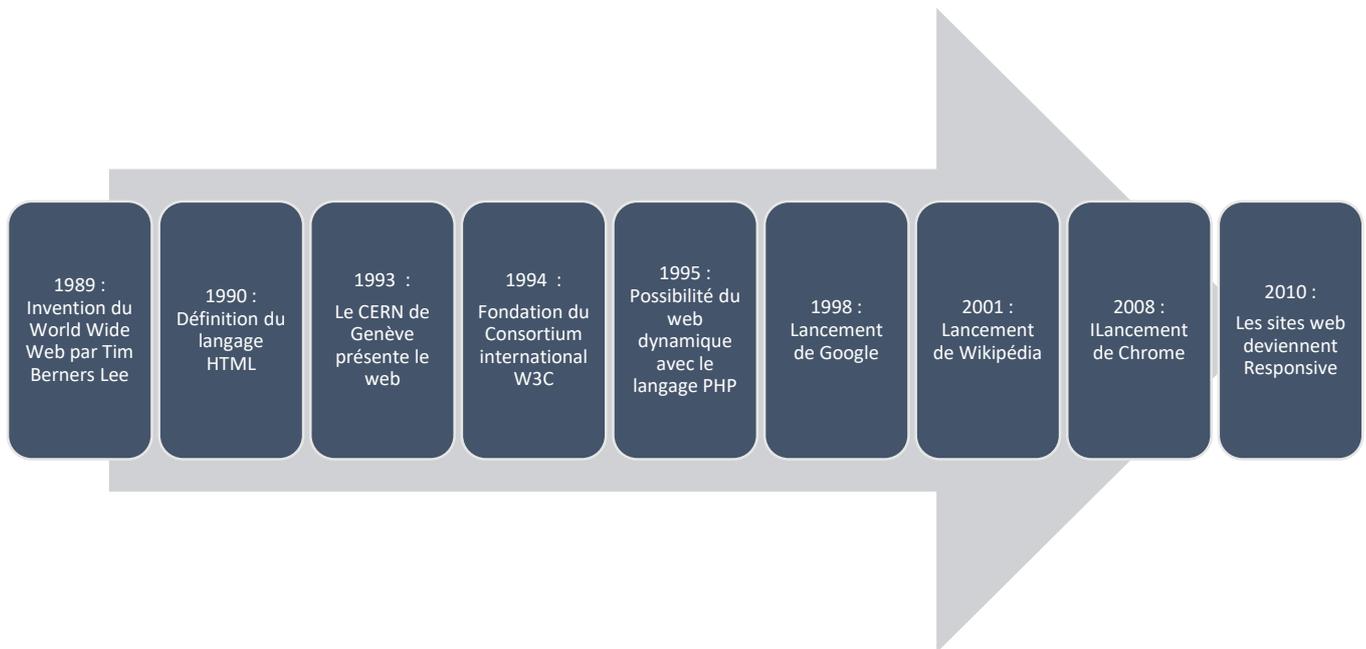


La page WEB

1. Repères historiques :



2. Généralités :

Lorsque nous naviguons sur Internet, nous visitons différents sites et nous regardons des pages avec notre navigateur internet. Ces pages sont écrites en langage HTML (HyperText Markup Language) et c'est notre navigateur qui les interprète pour les afficher de manière lisible à l'utilisateur.

De nos jours, le **HTML** est principalement utilisé comme langage de balisage uniquement dans sa fonction descriptive, c'est-à-dire **pour le contenu**, tandis que les langages pour les feuilles de style comme le **CSS** (Cascading Style Sheets) sont utilisés pour **définir le design**.

La plupart du temps, on aura alors deux balises marquant l'une le début et l'autre la fin, à savoir respectivement `<>` et `</>`. Pour certains éléments, la balise de fin n'est pas nécessaire, nous le verrons plus tard. Par ailleurs, quelques éléments sans contenu existent, comme le saut de ligne `
`.

Outre les éditeurs simples comme le *Bloc-note* de Microsoft, il existe également des éditeurs spéciaux qui peuvent offrir une aide précieuse : des applications vont par exemple mettre en évidence des balises syntaxiques en couleur, ce qui donne une très bonne vue d'ensemble du code écrit. De plus, les erreurs de syntaxe sont également souvent détectées. Une autre caractéristique usuelle est l'auto-complétion, qui soumet des suggestions pour étendre ou compléter votre code lorsque vous écrivez des balises HTML et qui notamment ajoute automatiquement les balises de fin. De plus, les éditeurs HTML ont souvent une fonction de prévisualisation qui permet d'examiner votre travail en cours grâce à un simple clic. Un bon éditeur pour les utilisateurs Windows est par exemple l'éditeur gratuit sous licence GPL [Notepad++](http://notepadplusplus.org/).

3. Squelette d'un document HTML

La conception d'une page web en suivant la philosophie du standard HTML5 spécifié par le W3C répond à quelques réflexes de base. La structure d'un document (l'imbrication des balises à la racine) est la suivante :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Titre de notre page</title>
  </head>
  <body>
    <!-- Le contenu de notre page -->
  </body>
</html>
```

- **Le doctype** : Son rôle est de préciser quel est le **type** de **document** qui va suivre, afin de permettre au navigateur de savoir quel langage il devra interpréter.
- **L'élément racine** `<html>` `</html>` : C'est la structure qui va accueillir la page web selon 2 éléments de hiérarchie :
 - `<head>` `</head>` : **l'en-tête** qui regroupe toutes les données qui ne sont pas représentées directement à l'écran dans le rendu du document : le titre, l'encodage, mais aussi d'autres (méta)informations qui serviront au référencement (pour les moteurs de recherche).
 - `<body>` `</body>` : **le corps** dans lequel on retrouvera tout le reste du contenu HTML, structuré par des balises variées, selon ce que l'on aura à y placer.

4. Structurer son contenu textuel

4.1 Les titres et les sous-titres

Les titres sont très importants dans la structure d'un site Web. Avec le HTML, on peut définir une hiérarchie claire pour tous les titres et sous-titres utilisés. Les balises utilisées vont de `<h1>` à `<h6>`. `<h1>``</h1>` représente le **titre principal de la page Web**. En principe, la balise `<h1>` doit être qu'une seule fois, contrairement aux autres titres `<h2>` et ainsi de suite. Il est important de **respecter l'ordre hiérarchique des balises** et de ne pas sauter aléatoirement entre les différents niveaux, afin que les lecteurs et les moteurs de recherche puissent bien comprendre la structure.

```
<h1>Titre principal</h1>
<h2>sous titre 1</h2>
<h3>sous titre 2</h3>
```

Titre principal

sous titre 1

sous titre 2

4.2 Les paragraphes

Pour définir des paragraphes, on a besoin de la balise `<p>` (p comme paragraphe).

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc nec viverra sem, faucibus laoreet diam.</p>
<p>Vestibulum consequat at purus in porta. Fusce gravida massa sem, consectetur ultrices mauris euismod sit amet. </p>
```

4.3 Le gras et l'italique

En HTML on a la possibilité de mettre en évidence des extraits de texte ou des mots de manière spécifique. Par exemple, avec les balises `<i>` et ``, vous pouvez mettre **en italique** des phrases, des expressions ou des citations. A noter cependant, l'écriture italique ralentit la lecture en général ; il faut donc utiliser ces deux balises avec parcimonie.

Les balises `` et ``, qui permettent de mettre des mots ou des extraits de texte **en gras**, peuvent être en revanche plus utiles. La spécificité de la balise `` est de renforcer spécifiquement un élément textuel important à l'intérieur d'un contenu et d'envoyer un signal au moteur de recherche pour votre référencement naturel.

4.4 Les sauts de ligne

Pour sauter une ligne dans notre contenu, on utilise la balise orpheline (car toute seule) `
`.

4.5 Les listes à puces ou énumérations

Les énumérations ne sont pas utiles que pour dresser des listes de courses : lorsqu'il s'agit de rédiger des textes, elles permettent de **favoriser une lecture fluide** et ainsi d'optimiser l'expérience utilisateur sur un site Web. HTML permet de créer :

- Des **listes à puces ordonnées** (avec numérotation) : à l'aide de la balise `` (« ordered list »)
- Des **listes à puces non-ordonnées** (sans numérotation) : à l'aide de `` (« unordered list »).

Chaque point de la liste est ensuite défini avec ``, qui ne fonctionne qu'en combinaison avec l'une des deux balises `` ou ``.

```
<ol>
  <li>premier élément</li>
  <li>deuxième élément</li>
  <li>troisième élément</li>
</ol>
```

```
1. premier élément
2. deuxième élément
3. troisième élément
```

```
<ul>
  <li>premier élément</li>
  <li>deuxième élément</li>
  <li>troisième élément</li>
</ul>
```

```
• premier élément
• deuxième élément
• troisième élément
```

4.6 Les tableaux

En HTML on utilise essentiellement les tableaux pour **présenter de manière pratique des données complexes**. Chaque tableau comprend au moins ces trois balises :

- **<table>** : la balise `<table>` marque le début d'un tableau HTML. Cependant, on ne peut pas aller très loin avec cette balise seule, car elle n'indique pas le nombre de lignes ou de colonnes à prendre en compte.
- **<tr>** : l'élément `<tr>` (« table row ») permet d'ajouter une ligne au tableau. Le nombre de ligne n'est pas limité.
- **<td>** : c'est seulement après avoir ajouté des colonnes que vous aurez enfin une structure claire de tableau. La balise `<td>` (« table data ») définit les données d'une cellule. Une balise `<tr>` va alors précéder l'ensemble des cellules `<td>` d'une même ligne. Le contenu de la cellule est fixé entre le marqueur d'ouverture `<td>` et le marqueur de fin `</td>`.

```
<table>
<tr>
  <td>case 1</td>
  <td>case 2</td>
</tr>
<tr>
  <td>case 3</td>
  <td>case 4</td>
</tr>
</table>
```

5. Intégrer des images, photos et graphiques

Voici 3 formats d'images couramment utilisés et qui sont supportés par tous les navigateurs courants :

- **JPG** : pour les photos ou les graphiques de haute qualité aux contrastes forts et d'une grande diversité de couleurs, le format JPG est généralement utilisé. Cela permet d'afficher plus de 16 millions de couleurs et de bien compresser des fichiers (bien que ceci implique une perte de qualité).
- **PNG** : graphiques et logos s'enregistrent idéalement au format PNG, qui peut afficher 256 (PNG8) à 16,7 millions de couleurs (PNG24). Contrairement au JPG, le PNG compresses sans perdre en qualité, mais la taille du fichier est toutefois conséquente.
- **GIF** : les fichiers GIF peuvent afficher seulement 256 couleurs, mais sont toujours en demande dans le développement Web, car ils permettent de bien rendre de petites animations, des éléments de navigation ou des graphiques simples.

En général on crée simplement un sous-dossier "images" dans le dossier de votre site Web (contenant également le fichier HTML) et on y stocke toutes les images utiles.

Indépendamment des questions de format, les images sont insérées aux pages HTML grâce à la balise ****.

Il est alors nécessaire de spécifier l'emplacement de stockage de l'image pour que le navigateur puisse logiquement la trouver et l'afficher. Pour cela, on a besoin de **l'attribut *src*** (source) suivi du chemin d'accès du fichier de l'image.

```

```

Les autres attributs sont facultatifs mais souvent utiles pour le référencement du site, notamment l'attribut ***alt*** (alternatif) qui décrit très succinctement l'image.

6. Les liens

Les hyperliens, mieux connus sous leur nom raccourci "liens", expliquent en partie le succès incroyable du Web. Sans ces liens électroniques, qui amènent les utilisateurs vers un autre site Web ou qui déclenchent une action spécifique telle que le téléchargement, il ne serait pas possible d'établir les interconnexions qu'offre le Web. La balise utilisée en HTML pour les liens est ***<a>***

6.1 Les liens internes

Les liens internes permettent de naviguer sur une page Web ou sur d'autres pages web du même site afin d'orienter les visiteurs.

6.1.1 Lien vers une autre page de notre site

On peut faire un lien vers la page « `ma_seconde_page.html` » en renseignant l'attribut ***href*** par **le chemin relatif** de cette page.

```
<a href="ma_seconde_page.html">En route vers la deuxième page</a>
```

6.1.2 Lien vers une rubrique de la même page (ma_page.html)

On peut aussi faire un lien vers un autre endroit de la même page (utile pour un lexique par exemple). Il faut d'abord créer une "**ancree**" à l'endroit que l'on veut atteindre par le lien :

```
<a name="toto">On veut aller ici</a>
```

Puis créer le lien avec

```
<a href="#toto">Lien pour aller là-bas</a>
```

6.2 Les liens externes

Les liens externes correspondent à un envoi vers un autre site Web. On renseignera donc dans l'attribut **href** **le chemin absolu** de la page (incluant le nom de domaine). Ce type de liens est utilisé pour recommander par exemple une page intéressante en rapport avec le sujet de votre page. Il doit apporter une valeur ajoutée à votre site. Cependant, assurez-vous que vous ne placez pas trop de liens sortants et que ces derniers sont dignes de confiance. Sinon, votre appréciation sur les moteurs de recherche risque de se détériorer.

De manière générale, lorsque vous formulez un texte de lien, essayez de le rendre significatif. Les formules bateau comme « cliquez sur ce lien » ou « ici » ne donnent pas au visiteur des informations très pertinentes sur ce qui se cache derrière.

```
<a href="http://www.allocine.fr/film/sorties-semaine/" target="_blank">Les sorties Cinéma de la semaine</a>
```

Dans l'attribut **target**, « *_blank* » permet d'indiquer que la page web citée en dans l'attribut **href** va s'ouvrir dans une nouvelle fenêtre ou un nouvel onglet.

6.3 Les autres liens

Tous les liens ne renvoient pas obligatoirement à des pages HTML. Selon les situations, cliquer sur un lien peut également déclencher un téléchargement ou encore d'activer le lecteur PDF. On met simplement dans l'attribut **href** le chemin relatif du fichier concerné.

7. Le CSS

Nous avons certes souligné que le HTML était la base de tout site Web, mais il faut bien noter que le design relève aujourd'hui d'un langage différent : en développement Web, on utilise en effet exclusivement les feuilles de style en cascade (**Cascading Style Sheets soit CSS**) pour les **couleurs** de chacun des éléments, la mise en page, la police et taille du texte ou encore bien d'autres éléments visuels. La séparation stricte du contenu et de la conception design facilite considérablement la gestion de grands projets Web.

On écrit ce fichier séparément (avec Notepad++ par exemple) dont l'extension est **.CSS** et la première chose à faire est de lier ce fichier (appelé *style.css*) au fichier HTML, pour cela on place dans l'entête du fichier HTML le code suivant :

```
<link rel="stylesheet" href="style.css" />
```

7.1 Sélectionner une balise pour appliquer un style

Dans un code CSS, on trouve 3 éléments différents avec la syntaxe de la figure suivante :

- **Des noms de balises** : on écrit les noms des balises correspondante en HTML dont on veut modifier l'apparence. Par exemple, si je veux modifier l'apparence de tous les paragraphes <p>, on écrit p (sans les symboles < et >).
- **Des propriétés CSS** : les « effets de style » de la page sont rangés dans des propriétés propres au CSS et dont on verra les principales.
- **Des valeurs** : Pour chaque propriété, on doit indiquer une valeur.

```
balise1
{
    propriete1: valeur1;
    propriete2: valeur2;
    propriete3: valeur3;
}
```

Dans l'exemple ci-après, on indique que tous les paragraphes du fichier HTML seront écrits en bleu.

```
p
{
    color: blue;
}
```

7.2 Appliquer un style avec les attributs class et id

Avec la méthode vue en 6.1, toutes les balises du fichier HTML auront le même style. Il nous faut utiliser un autre moyen pour affecter un style particulier à un paragraphe par exemple. Il existe deux attributs :

- L'attribut **class** : valable sur plusieurs balises
- L'attribut **id** : valable que sur une seule balise

7.2.1 L'attribut class

C'est un attribut que l'on peut mettre sur n'importe quelle balise HTML, on donnera comme valeur à cet attribut un nom permettant d'identifier la balise concernée.

Dans l'exemple suivant, nous avons identifier notre premier paragraphe avec l'attribut class ayant pour valeur *intro*.

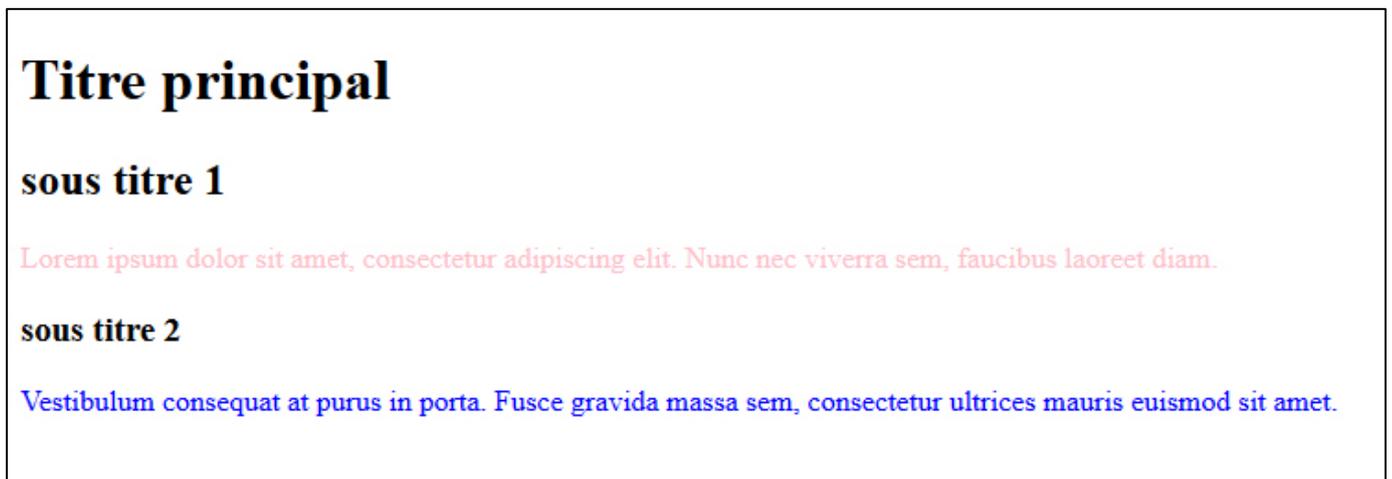
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <link rel="stylesheet" href="style.css" />
    <title>Titre de notre page</title>
  </head>
  <body>
    <h1>Titre principal</h1>
    <h2>sous titre 1</h2>
    <p class="intro">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc nec viverra sem, faucibus laoreet diam.</p>
    <h3>sous titre 2</h3>
    <p>Vestibulum consequat at purus in porta. Fusce gravida massa sem, consectetur ultrices mauris euismod sit amet. </p>
  </body>
</html>
```

Et dans notre fichier CSS, nous avons un style pour tous les paragraphes et un style particulier pour les balises ayant l'attribut class dont la valeur est intro. On notera que le nom de la classe est précédé du caractère « . »

```
P
{
    color: blue;
}

.intro
{
    color: pink;
}
```

Le résultat, dans le navigateur internet est donc sans surprise :



7.2.2 L'attribut id

Il fonctionne de la même manière que class dans le fichier HTML à un détail près. En effet il ne peut être utilisé qu'une seule fois !

L'exemple ci-après montre son utilisation dans le fichier HTML dans la balise img.

```

```

Et dans le fichier CSS le id est précédé du caractère « # ». Dans l'exemple suivant, l'image identifiée *logo* sera entourée d'un contour noir d'épaisseur 1 pixel.

```
#logo
{
    border: 1px black;
}
```

7.3 Les balises universelles

En fait, on a inventé deux balises dites **universelles**, qui n'ont aucune signification particulière mais qui peuvent être utiles :

- ** **: c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement. On va par exemple pouvoir placer une certaine partie du texte d'un titre ou d'un paragraphe dans un élément `` pour ensuite pouvoir lui appliquer un style CSS particulier, chose qu'il nous était impossible de faire jusqu'à présent.

Exemple : Imaginons que nous voulions mettre en jaune fluo la partie *titre* de sous-titre 2.

```
<h3>sous <span>titre</span> 2</h3>
```

```
span
{
  background-color:yellow;
}
```

- **<div> </div>**: c'est une balise de type **block**, qui entoure un bloc de texte. Ces balises créent un nouveau « bloc » dans la page et provoquent donc obligatoirement un retour à la ligne. `<div>` est une balise fréquemment utilisée dans la construction d'un design, comme nous le verrons plus tard.

Exemple : Imaginons que nous voulions rassembler dans un même bloc le sous-titre 1 et notre premier paragraphe pour mettre un arrière-plan vert.

```
<div>
<h2>sous titre 1</h2>
<p class="intro">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc nec viverra sem, faucibus laoreet diam.</p>
</div>
```

```
div
{
  background-color:green;
}
```

Remarque : le résultat est différent de celui où j'applique pour chaque balise l'arrière-plan vert.

7.4 Et des balises contenues dans d'autres

On peut également appliquer un style dans des balises contenues dans d'autres balises. On utilisera la syntaxe suivante :

```
div h2
{
  color: white;
}
```

Cela signifie que toutes les balises `<h2>` contenues dans une balise `<div>` seront affecté de ce style.

STRUCTURE DU DOCUMENT

```
<!DOCTYPE>
Version du (X)HTML
<html>
Document HTML
<head>
Informations de la page
<body>
Contenu de la page
```

INFORMATIONS DE LA PAGE

```
<title>
Titre de la page
<meta />
Méta-données
<base />
Adresse de la base
<link />
Ressource externe
<style>
Définition de styles
<script>
Définition de scripts
```

BALISES DE STRUCTURE

```
<div>
Enveloppe en bloc
<span>
Enveloppe en chaîne
<section>
Section
<nav>
Navigation
<header>
Header
<footer>
Footer
<article>
Article
<aside>
Contenu associé / secondaire
```

LIENS

```
<a href="">
Lien vers un fichier
<a href="mailto:">
Lien vers un mail
<a name="">
Ancre
<a href="#name">
Lien vers une ancre
```

MISE EN FORME DU TEXTE

```
<h[1-6]>
Niveaux de titres
<p>
Paragraphe
<br />
Saut de ligne
<strong>
Mise en gras (SEO)
<b>
Mise en gras
<em>
Mise en italique (SEO)
<i>
Mise en italique
Citation externe
<sup>
Surélévation
<sub>
Sous-élévation
<code>
Code
<address>
Adresse
<cite>
Citation
```

TABLEAUX

```
<table>
Tableau
<caption>
Titre du tableau
<thead>
En-tête du tableau
<tbody>
Corps du tableau
<tr>
Ligne
<td>
Cellule
```

LISTES

```
<ul>
Liste
<ol>
Liste ordonnée
<li>
Élément d'une liste
<dl>
Liste de définitions
<dt>
Terme
<dd>
Définition
```

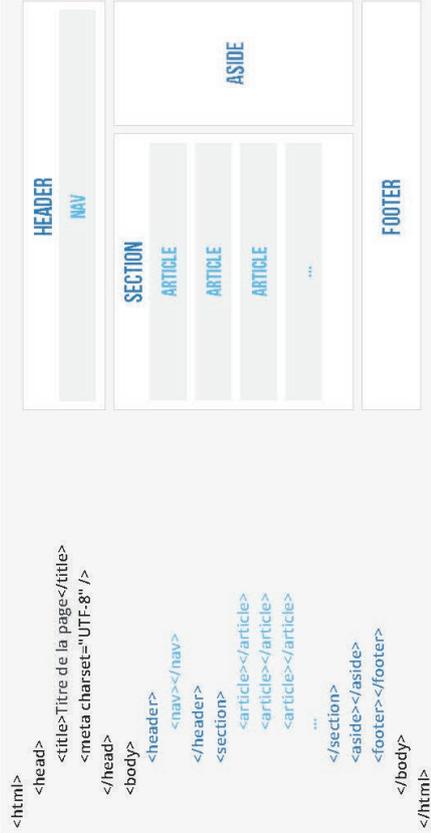
FORMULAIRES

```
<form>
Zone d'un formulaire
<input />
Champ d'un formulaire
<textarea>
Champ texte multi-lignes
<button>
Bouton
<select>
Liste déroulante
<optgroup>
Groupe d'options
<option>
Option d'une liste
<label>
Titre d'un élément
<fieldset>
Groupe de champs
<legend>
Titre d'un fieldset
<datalist>
Liste de choix
<output>
Résultat d'un calcul
```

MÉDIAS

```
<img>
Image
<map>
Zoning sur une image
<area>
Zone du zoning
<canvas>
Bloc graphique
<figure>
Conteneur de média
<figcaption>
Titre d'un conteneur
<audio>
Son
Ressources d'un média
<source>
Listes d'un média
<track>
Vidéo
<video>
Vidéo
<object>
Objet externe
<param>
Paramètre d'un objet
```

STRUCTURE D'UN DOCUMENT HTML 5



DOCUMENTATION

www.w3schools.com
Documentation de référence sur le HTML 5, le CSS 3, et d'autres langages de programmation ...

www.openclassrooms.com
Excellents tutoriels proposés gratuitement, avec beaucoup d'explications et d'exemples

www.alsacreation.com
Communauté dédiée à l'apprentissage des standards web (W3C, HTML, CSS, JS)

RESSOURCES ET INSPIRATION

www.webcreme.com
www.cssmania.com
www.themeforest.net
www.wordpress.com
www.webdesign-inspiration.com
www.thebestdesigns.com
www.art-spire.com/category/webdesign/

PIXELCRÉATION

Support réalisé par Jacques Danilo / Graphiste, photographe et formateur freelance
www.pixelcrea.com

