

# 1 Web et Internet

## Histoire 1

En 1989, **Tim Berners-Lee**, qui s'occupe de la gestion des documents scientifiques au **CERN**, propose avec le projet *mesh* (filet) de faire traverser le réseau internet à l'**hypertexte** imaginé par **Ted Nelson** en 1965. C'est la naissance du *World Wide Web* : des liens hypertextes relient des documents stockés sur des machines différentes et l'utilisateur d'une machine cliente peut naviguer de lien en lien à travers ce réseau logique d'informations qui se superpose au réseau physique d'internet.

Avec l'ouverture au public des standards du Web en 1993 et leur standardisation organisée par le **W3C**, qui assure la portabilité des pages Web sur toutes les machines, le Web s'impose à partir du milieu des années 90 comme le service le plus populaire disponible sur Internet.

## 2 Les fondements du Web

### 2.1 Architecture client / serveur

#### Point de cours 1

Comme la plupart des services d'Internet, le Web repose sur une **architecture client / serveur** : pour obtenir une page Web, le **client** la demande au **serveur** qui la détient.

Le client et le serveur sont des logiciels et peuvent être installés sur n'importe quelle machine.

Le logiciel client s'appelle un **navigateur**, comme par exemple *Firefox* ou *Chrome*.

### 2.2 URL : l'adressage logique

#### Point de cours 2

Les documents disponibles sont localisés sur le Web par une adresse logique, **Uniform Resource Locator**.

La syntaxe des URL est standardisée, par exemple décomposons :

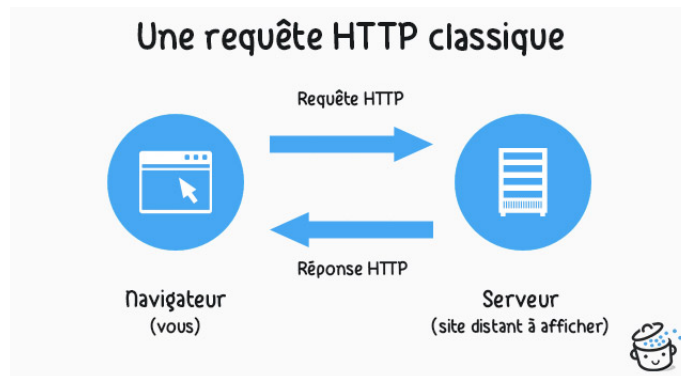
```
https://www.gnu.org/gnu/linux-and-gnu.fr.html
```

- le protocole est `https`;
- le nom de domaine sur Internet de la machine est `www.gnu.org`, le préfixe `www` étant le nom usuel du dossier public sur un serveur Web;
- `gnu/linux-and-gnu.fr.html` est le chemin vers la ressource sur le serveur : le fichier `linux-and-gnu.fr.html` qui se trouve dans le dossier `gnu` lui-même dans le dossier public `www`.

## 2.3 HTTP : le protocole

### Point de cours 3

Lors d'un échange Web, le **client** et le **serveur** dialoguent selon des règles fixées par un protocole standardisé, **HyperText Transfer Protocol**.



Source : <https://wpmarmite.com/wordpress-https/>

Si la page demandée n'est pas disponible, le serveur retourne le célèbre code d'erreur 404. Nous étudierons plus en détail le protocole HTTP un peu plus tard dans l'année.

## 2.4 HTML : le langage

### Point de cours 4

Les **premières pages Web** sont des **pages statiques**, c'est-à-dire que la page retournée par le serveur ne dépend pas du client. Le premier langage du Web est **HyperText Markup Language** qui est un langage de description de documents à l'aide de **balises**. Un fichier écrit en HTML est un simple fichier texte qui est interprété par le navigateur pour obtenir un affichage. Pour éditer un fichier texte, on utilise un **éditeur de textes** comme **Notepad++**.

HTML permet de définir la structure d'un document en le découpant en **éléments** logiques délimités par une **<balise>** ouvrante et une **</balise>** fermante. En particulier il permet de créer des liens hypertextes entre deux éléments d'un même document ou de documents différents.

L'agencement des **balises** doit respecter des règles de bonne formation : pour les balises imbriquées on procède comme pour les parenthèses.

Balises mal imbriquées

```
<div> Ma partie
  <p> Mon paragraphe
</div>
  </p>
```

Balises bien imbriquées

```
<div> Ma partie
  <p> Mon paragraphe </p>
</div>
```

On donne ci-dessous la structure minimale d'un fichier HTML :

- Une balise `<!DOCTYPE html>` qui déclare le type du document.

- Une balise racine <html>.
- Une balise d'entête <head> avec des meta-informations qui ne sont pas affichées.
- Une balise de corps <body> qui contient tout le contenu affiché.
- Des commentaires dans une balise spéciale <!-- ici le commentaire --> .

## Code source HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <h1 class="titre">Langage HTML</h1>
    <h2 class="titre">Recommandation</h2>
    <p>Un <a id="monsie" href="https://frederic-
      junier.org/wp">Super site</a>
      à visiter absolument ! </p>
    <!-- Commentaire : sauts de ligne et indentation
      pas affichés mais
      facilitent la relecture -->
  </body>
</html>
```

## Affichage dans le navigateur

# Langage HTML

## Recommandation

Un [Super site](https://frederic-junier.org/wp) à visiter absolument !

## 3 Langages du Web

### 3.1 Principe de séparation



#### Point de cours 5

Une page Web est un document de structure non linéaire (liens hypertextes), avec des contenus riches (images, video) et des interactions possibles côté client (navigateur) ou côté serveur (accès à une base de données).

On distingue les pages Web **statiques** dont le contenu ne dépend que de l'utilisateur et de ses interactions, des pages Web **dynamiques**.

Un **principe de séparation** des fonctionnalités est appliqué dans la réalisation d'une page Web :

- La structure du document est définie par le langage HTML. Pour les pages Web dynamiques, le code HTML peut être généré par le serveur et envoyé au client à sa demande.
- L'apparence et la disposition des éléments sont définies par le langage CSS.
- La gestion des événements utilisateurs (clics ...) est assurée par le langage Javascript qui peut modifier la structure du document HTML dans le navigateur du client.

## 3.2 HTML pour la structure



### Point de cours 6

Le langage HTML permet de définir la structure d'une page Web.

Il existe deux grandes catégories de balises :

- **balises en-ligne** qui ne sont pas affichées après un saut de ligne, se juxtaposent et occupent juste la place nécessaire à leur contenu en largeur. Certaines de ces balises sont vides, elles n'ont pas de contenu et donc pas de balise fermante.
- **balises blocs** qui sont affichées après un saut de ligne, se superposent et occupent toute la place possible en largeur.

Certaines balises ont une **sémantique** prédéfinie : `<article>`, `<section>`, très utile pour l'indexation de la page par un moteur de recherche. D'autres balises comme `<span>` (en-ligne) ou `<div>` (bloc) sont neutres et personnalisables par le développeur.

### Méthode

- Les principales balises de type **en-ligne** qu'il faut connaître :

Balise	Sémantique	Syntaxe
<code>&lt;a&gt;</code>	lien hypertexte	<code>&lt;a href="https://site.org/index.html"&gt;Lien&lt;/a&gt;</code>
<code>&lt;br&gt;</code>	saut de ligne	<code>&lt;br&gt;</code>
<code>&lt;img&gt;</code>	insertion d'image	<code>&lt;img src="https://linux.org/Tux.png" alt="tux"&gt;</code>
<code>&lt;strong&gt;</code>	texte très important	<code>&lt;strong&gt;important&lt;/strong&gt;</code>
<code>&lt;em&gt;</code>	texte important	<code>&lt;em&gt;important&lt;/em&gt;</code>
<code>&lt;span&gt;</code>	balise neutre	<code>&lt;span&gt;contenu&lt;/span&gt;</code>

En général, le contenu de la balise `<strong>` est en gras et celui de la balise `<em>` en italique.

- Les principales balises de type **bloc** qu'il faut connaître

Balise	Sémantique	Syntaxe
<code>&lt;p&gt;</code>	paragraphe	<code>&lt;p&gt;mon paragraphe. &lt;/p&gt;</code>
<code>&lt;li&gt;</code>	item de liste	<code>&lt;li&gt;item&lt;/li&gt;</code>
<code>&lt;ol&gt;</code>	liste ordonnée	<code>&lt;ol&gt;&lt;li&gt;lundi&lt;/li&gt;&lt;li&gt;mardi&lt;/li&gt;&lt;/ol&gt;</code>
<code>&lt;ul&gt;</code>	liste non ordonnée	<code>&lt;ul&gt;&lt;li&gt;basket&lt;/li&gt;&lt;li&gt;short&lt;/li&gt;&lt;/ul&gt;</code>
<code>&lt;div&gt;</code>	balise neutre	<code>&lt;div&gt;&lt;p&gt;paragraphe 1&lt;/p&gt; &lt;p&gt;paragraphe 2&lt;/p&gt;&lt;/div&gt;</code>
<code>&lt;h1&gt;</code>	titre de niveau 1	<code>&lt;h1&gt; Section &lt;/h1&gt;</code>
<code>&lt;h2&gt;</code>	titre de niveau 2	<code>&lt;h2&gt; Sous-section &lt;/h2&gt;</code>

Il existe six niveaux de titres de `<h1>` à `<h6>` par ordre décroissant d'importance.

## 3.3 CSS pour la mise en forme

## Point de cours 7

Le langage CSS permet de définir l'apparence et la disposition des éléments HTML.

Pour respecter le principe de séparation, les **définitions de style CSS** sont regroupées dans un fichier texte d'extension `.css` qui est relié au fichier source HTML par une balise `link` placée dans l'entête du document. On parle souvent de **feuille de style CSS**.

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

On peut aussi définir des règles CSS directement dans le fichier HTML dans un attribut de balise `style` ou dans l'entête du document à l'intérieur d'une balise `<style>`. Si un élément n'a pas de règle de style particulière il applique celle de son plus proche parent. Si un élément est ciblé par plusieurs règles, la plus locale s'applique :

attribut `style` > balise `<style>` > fichier `.css`

Le principe d'une règle de style CSS est d'appliquer un ensemble de définitions (propriété : "valeur") à un ensemble d'éléments HTML défini par un **sélecteur**.

```
p {
  propriete1 : "valeur";
  propriete2 : "valeur";
}

#monidentifiant {
  propriete1 : "valeur";
}

.uneclasse {
  propriete2 : "valeur";
}
```

Un **sélecteur** CSS peut être :

- Le type d'un élément HTML par exemple `p` pour cibler tous les paragraphes ;
- Un **identifiant** avec le préfixe `#` pour cibler l'unique élément qui a été marqué par cet identifiant dans le document HTML avec la syntaxe `<element id="monidentifiant">`.
- Une **classe** avec le préfixe `.` pour cibler un ensemble d'éléments, pas forcément de même type, qui ont été marqués par cette classe dans le document HTML avec la syntaxe `<element class="uneclasse">`.
- Une combinaison de sélecteurs, voir [https://www.w3schools.com/Css/css\\_combinators.asp](https://www.w3schools.com/Css/css_combinators.asp).

Voir la page [https://www.w3schools.com/Css/css\\_intro.asp](https://www.w3schools.com/Css/css_intro.asp) pour la diversité des actions possibles avec une feuille de style CSS sur un même code source HTML.

On donne ci-dessous un exemple de feuille de style CSS associée à un fichier source HTML avec l'affichage obtenu.

## Code source HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <link href="style.css" rel="
      stylesheet" type="text/
      css" />
  </head>
  <body>
    <h1 class="titre">Langage HTML</
      h1>
    <h2 class="titre">Recommandation<
      /h2>
    <p>Un <a id="monsite" href="https
      ://frederic-junier.org/wp">
      Super site</a>
      à visiter absolument ! </p>
  </body>
</html>
```

## Code CSS

```
p {
  font-style: normal;
}

#monsite {
  font-style: italic;
}

.titre {
  border: 2px solid black;
}
```

## Affichage obtenu

**Langage HTML**

**Recommandation**

Un [Super site](https://frederic-junier.org/wp) à visiter absolument !