

Cours HTML / CSS

Introduction	3
Présentation formateur et apprenants	3
Définition du cadre individuel et collectif de la formation	3
Présentation du programme et du déroulé de la journée	3
1. Les bases.....	5
1.1 Histoire du Web	5
1.2 Les principaux langages et outils du Web.....	9
1.3 Les grands principes du Web	12
2. Mise en pratique.....	17
2.1 Choix d'un éditeur de texte	17
2.2 Création du répertoire de travail	18
2.3 Création de la première page internet.....	19
2.4 Structuration sémantique des pages HTML.....	21
2.5 Insertion liens, ancrs, et images	37
2.6 Les tableaux	40
2.7 Les Formulaires	44
2.8 Audio et vidéo	59
2.8.1 La balise video.....	60
2.8.2 Balise Audio	62
2.8.3 Les éléments embarqués (frameset, Iframe, etc...).....	63
3. Les éléments blocks et inline.....	65
3.1 Généralités	65
3.2 Notion de flux et principe des boîtes	66
4. CSS	69
4.1 Définition	69
4.2 CSS - 3 modes d'intégration.....	70
4.3 La syntaxe du CSS.....	71
4.4 Les sélecteurs du CSS	72
4.5 Mise en forme du texte et autres éléments	96
4.6 La mise en page.....	102
4.7 La décoration 1, les bases	128
4.8 La décoration 2.....	153
4.9 Le modèle table cell	156
4.10 Le modèle flexbox.....	159
4.11 Le modèle grid (grid layout)	163
4.12 couplage grid et flexbox.	218

4.13 Transformations et animations	220
4.14 L'héritage et la cascade.....	228
4.15 Le Responsive Web Design.....	233

Présentation formateur et apprenants

Attendus de la formation

Objectifs personnels et professionnels

Définition du cadre individuel et collectif de la formation

Travail individuel et travail collaboratif.

Liste des principales ressources

- MDN Mozilla / Alsacreations
- [Canibase.com](https://caniuse.com)
- Openclassrooms / Udemy / Dyma...
- <https://html5up.net/>
- <https://www.welcometothejungle.com/>
- <https://flatuicolors.com/palette/defo>
- [Adobe Kuler](https://kuler.adobe.com/)
- <https://fontforge.org/en-US>

La liste sera plus complétée au fur et à mesure dans la semaine.

Généralités sur les métiers du Web & qualités requises

Rigueur, proactivité, persévérance, curiosité intellectuelle

Présentation du programme et du déroulé de la journée

- 1 Test court www.kahoot.it
- 2 Correction du test
- 3 Cours les bases du Web
- 4 Exercices HTML
- 5 Correction des exercices

Mini-projet

Dans le cadre de cette formation HTML / CSS vous devrez en 5 jours développer un mini site comprenant :

1. Une page d'accueil
2. Une page service / vente
3. Une page formulaire de contact

Qualités que doit développer un développeur 😊

1. Rigueur
2. Persévérance
3. Sens de l'autonomie et capacité à travailler en équipe
4. Curiosité intellectuelle
5. Flexibilité / adaptabilité / polyvalence
6. Humilité
 - a. Vis-à-vis de ses camarades de classes, collègues...
 - b. Vis-à-vis des langages
7. Ponctualité

Conseils :

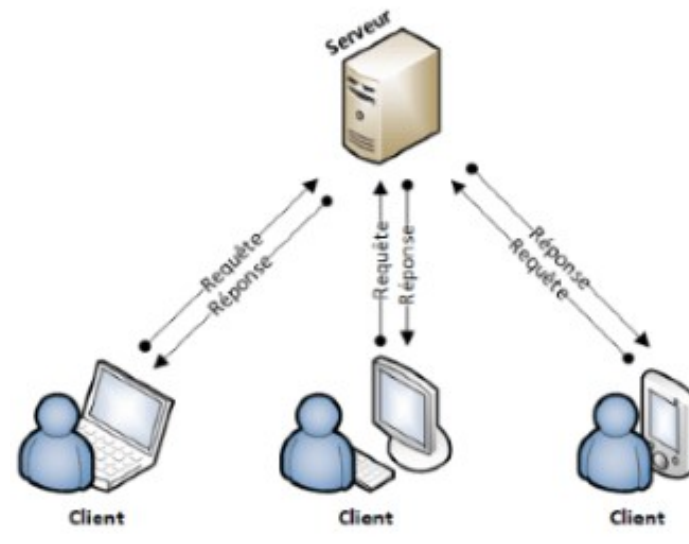
Coder régulièrement, si possible tous les jours : le volume de connaissances assimilées est très important, et il est facile d'oublier ce que l'on a appris si l'on ne code pas régulièrement.

Le développement nécessite de la concentration, il est préférable de couper toute source de distraction (téléphone, messagerie...). Il faut éviter de coder tout en parlant (ne pas faire comme le formateur 😊).

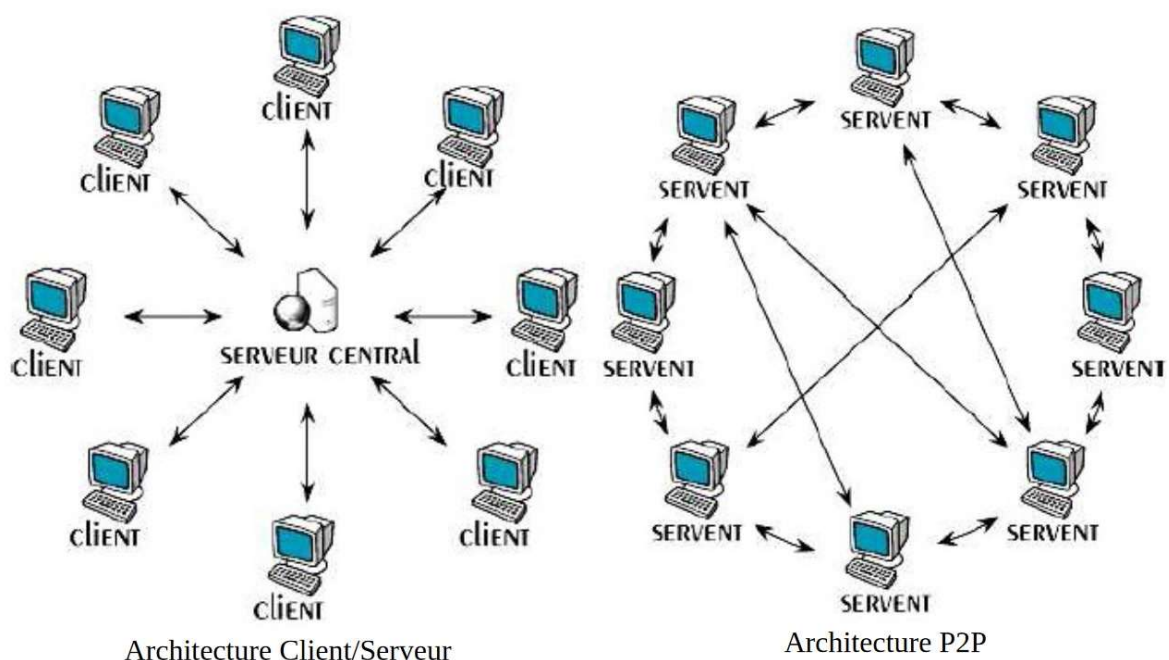
1. Les bases

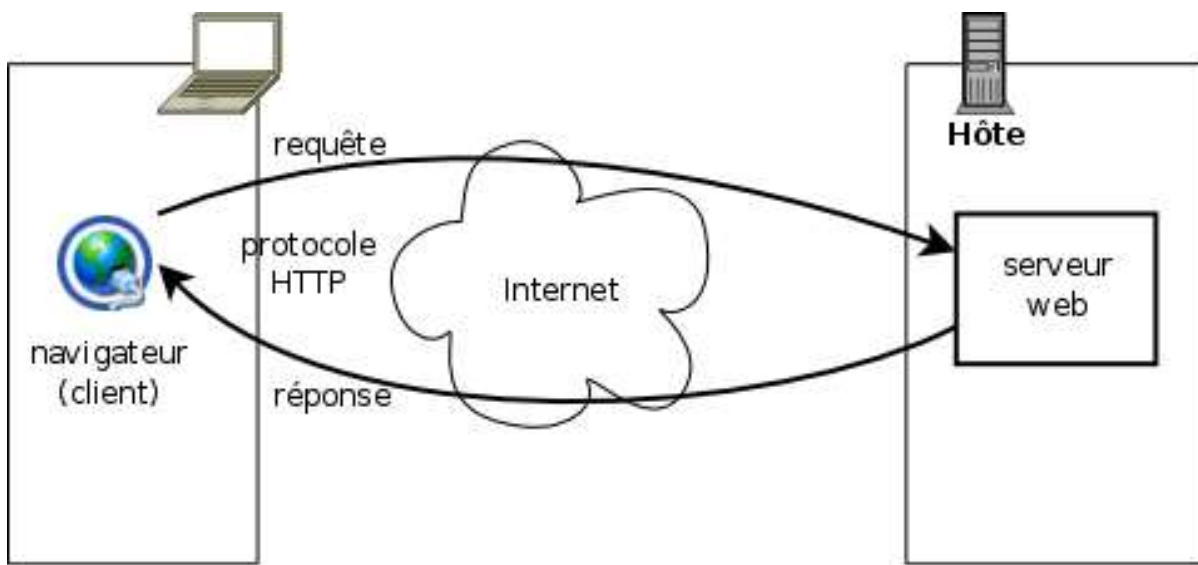
1.1 Histoire du Web

1.1.1 L'architecture client-serveur.



Pour appréhender internet, il est important de comprendre qu'il existe dans les grandes lignes **2 types de machines**. D'une part, les **machines clientes** connues par tous en tant que "simples utilisateurs", sur lesquelles nous exécutons nos tâches quotidiennes. Elles ne communiquent pas directement avec les autres ordinateurs clients (sauf exceptions machines client-serveur), ce sont des ordinateurs qualifiés de **serveurs** qui permettent la communication entre les clients.





1.1.2 L'ancêtre d'internet ARPANET

Aujourd'hui, le Web évoque internet, le World Wide Web créé dans les années 1989-92 par **Tim Berners-Lee** suivi de **Robert Cailliau**. Web signifie en anglais la toile, son origine est plus ancienne.

ARPANET:

Pour résumer grossièrement internet est un réseau d'ordinateurs communicant entre eux, mais l'idée de communication entre machines éloignées géographiquement est plus ancienne que la création du Web. L'armée américaine avait dès les années 1960 travaillé sur un réseau à transfert de paquets appelé ARPANET.

ARPANET ou Arpanet (acronyme anglais de « Advanced Research Projects Agency Network », souvent typographié « ARPAnet »¹) est le premier réseau à transfert de paquets développé aux États-Unis par la DARPA. Le projet fut lancé en 1962, mais ARPANET ne vit le jour qu'en 1969. Sa première démonstration officielle date d'octobre 1972. (source wikipedia).

1.1.3 Le World Wide Web

- 1989-1992 : Origine
- 1995-1996 : HTML 2.0
- 1997 : HTML 3.2. et 4.0
- 2000-2006 : XHTML
- 2007 à nos jours : HTML5 et abandon du XHTML 2
- W3C visait la finalisation de la spécification HTML5 en 2014.

1.1.4 Le protocole HTTP

HTTP (***Hypertext Transfer Protocol*** : protocole de transfert hypertexte) comme son nom l'indique, il permet la communication entre des machines client-serveur. C'est donc un protocole de communication. Il a été développé pour le **World Wide Web**.

http(s):// indique le protocole Hyper Text Transfert Protocole (secured) utilisé

www. correspond au sous-domaine par défaut, et que l'on peut personnaliser

Dans **monsite.com** **monsite** représente le domaine acheté, **com** indique l'extension choisie (TLD), elle est générique ou nationale (fr, sk, uk...).

1.1.5 L'URL

Sur le Web, tout est URL : pages, images, fichiers CSS, JS... Mais comment localiser les données spécifiques (ressources) de manière uniforme ?

C'est le rôle de de l'URL.

L'URL est le sigle anglais de **Uniform Resource Locator** ou localisateur uniforme de ressource en français.

1.2 Les principaux langages et outils du Web

1.2.1 HTML

L'HTML (Hypertext Markup Language) est un langage de balisage permettant de représenter des pages web.

Ce langage permet comme son nom l'indique d'écrire de l'hypertexte, mais aussi :

- de structurer les pages à l'aide de balises sémantiques
- de mettre en forme le contenu
- de créer des formulaires de saisie ou d'envoi de fichiers.
- d'interagir avec d'autres langages telles que JavaScript ou CSS.

1.2.2 CSS

Les feuilles de style en cascade, ou CSS (Cascading Style Sheets), forment un langage qui permet de mettre en page, animer, et styliser la présentation de documents HTML, XML, et SVG.

- 1995: Les premiers développements
- 1996: CSS1, des implémentations difficiles pendant la guerre des navigateurs
- 2001: CSS2, des ambitions précipitées
- 2007: CSS 2.1, retour aux implémentations
- 2008: CSS3 se généralise

1.2.3 JavaScript

Langage de programmation souvent associée à HTML et CSS pour rendre les pages internet plus dynamiques et interactives. Il permet également de vérifier les formulaires avant envoi au serveur. C'est un langage au départ Front (s'exécute du côté client), mais couplé avec NodeJs, il permet de gérer le Front et le Back (côté serveur).

1.2.4 PHP

Le PHP (Hypertext Preprocessor) C'est un langage programmation libre, s'exécutant sur le serveur (langage back). Il permet de traiter les données envoyées par le client et de produire des pages Web dynamiques. Il peut également générer des pages HTML et gérer une base de données.

1.2.5 Les API

Une API (Application Programming Interface) est un ensemble de fonctions et de classes fournies par une bibliothèque logicielle, un service web, ou un système d'exploitation, de sorte qu'ils puissent être utilisés pour programmer un logiciel qui les emploie.

Tous les navigateurs implémentent des API standards exploitables en JavaScript :

- Les **DOM** (Document Model Object) modifient les éléments dans la page Web, grâce notamment à la syntaxe `"document.getElementById"`.
- Les **BOM** (Browser Object Model) sont des éléments affectant le navigateur internet, ils permettent de manipuler le navigateur, au travers d'objets navigator, location, document, history, screen, window...

A la différence du DOM, il n'y a pas standard strict et officiel pour implémenter dans le cadre du BOM. De cette manière, les navigateurs peuvent implémenter plus librement.

Il existe de nombreux BOM: Géolocalisation (latitude et longitude), Web Storage (stockage dans le navigateur), Offline (fonctionnement sans connexion), Bandwith (qualité de la connexion), Battery (état de la batterie), GetHours (heure et date)...

1.2.6 APACHE

APACHE est un serveur Web HTTP très populaire utilisé pour gérer les requêtes du WWW. Il peut aussi effectuer le stockage de données (BDD), assurer d'autres services tels que l'envoi d'e-mails, le transfert de fichiers par FTP... (il n'est pas le seul voir Nginx, Cherokee, Lighttpd).



Autorisez-vous [www.darty.com](#) à accéder à votre localisation ?

[En savoir plus...](#)

☐ Se souvenir de cette décision

Autoriser l'accès à la localisation

Ne pas autoriser



Autorisez-vous [siecledigital.fr](#) à envoyer des notifications ?

[En savoir plus...](#)

Autoriser les notifications

Plus tard



Firefox utilise Google Location Services pour déterminer votre **localisation** via l'adresse IP de votre PC, des informations sur les points d'accès sans fil environnants, et un identifiant client aléatoire.

Le standard "Web Push" permet aux sites d'envoyer des **notifications** à leurs utilisateurs sous forme de nouveaux messages ou de contenu mis à jour.

1.3 Les grands principes du Web

1.3.1 W3C

Le World Wide Web Consortium est un organisme de normalisation qui est chargé de promouvoir la compatibilité des technologies du WWW : HTML, XHTML, XML, CSS, PNG, SVG...

<http://www.w3c.fr> et <http://www.w3.org>

Il existe un autre groupe de travail, WHATWG (fondé en 2004 par Opera, Apple et Mozilla).

<https://whatwg.org>

HTML a deux grands rôles :

- l'**accessibilité** d'un même document par tous
- la structuration **sémantique** d'un document

Afin d'optimiser ces deux concepts, HTML offre une diversité de **balises** et **attributs**.

1.3.2 Accessibilité

L'accessibilité pour le développement web permet l'utilisation des sites web par le plus grand nombre de personnes, même lorsque leurs capacités sont limitées, d'une manière ou d'une autre.

Les utilisateurs déficients

HTML définit des moyens de créer du contenu web accessibles aux personnes ayant un handicap.

Pour une image par exemple, un texte alternatif sera rendu en audio.

<http://www.accessiweb.org>

<http://references.modernisation.gouv.fr/rgaa-accessibilite/criteres.html>

ARIA (Accessible Rich Internet) est un standard qui permet notamment des interactions clavier.

<https://www.w3.org/TR/wai-aria-1.1/>

Les terminaux

L'accessibilité guide les intégrateurs vers la construction d'une interface Web unique qui va s'adapter à tous les types de terminaux, quel qu'ils soient, en fonction de :

- leurs constructeurs : Microsoft, Google, Mozilla, Apple...
- leurs dimensions, de l'écran le plus petit au plus grand

leurs environnements : heure, connexion, lieu, batterie, langue... de l'utilisateur

Neutralité du Web

Pour résumer grossièrement c'est un principe qui vise à garantir à chacun la même bande passante sans discrimination.

Les navigateurs :

Ce sont des logiciels permettant d'accéder au WWW et d'afficher ainsi les pages internet. Le premier navigateur internet est [WorldWideWeb](#) créé en 1990 par [Tim Berners-Lee](#). Les navigateurs utilisent des fonctionnalités complexes et n'ont pas les mêmes moteurs de rendu*, cela aboutit à des résultats différents pour un même code source.

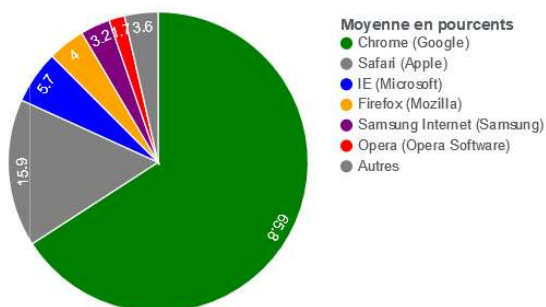
Moteurs de rendu HTML

Un moteur de rendu HTML est un composant logiciel de base qui permet aux logiciels d'afficher les éléments d'une page Web , il en existe plusieurs :

- **Gecko** : Firefox Mobile, Mozilla Firefox, Pale Moon...
- **KHTML**, WebKit ou Blink : Epiphany, Google Chrome, Opera, Safari
- **Presto** : Opera Mini, Opera Mobile, Opera Wii, Nintendo DS Browser...
- **Trident** : Internet Explorer/ Edge, Internet Explorer Mobile...

Les parts de marché des navigateurs Web dans le monde, toutes plateformes confondues
(octobre 2020 - [mettre à jour](#))

Source ↕	Chrome (Google) ↕	Safari (Apple) ↕	Firefox (Mozilla) ↕	IE + Edge (Microsoft) ↕	Opera (Opera Software) ↕	Samsung Internet (Samsung) ↕	UC Browser (UCWeb) ↕	Autres ↕
StatCounter ¹³	66,1 %	17,2 %	4,0 %	4,4 %	2,1 %	3,2 %	1,2 %	1,8 %
NetMarketshare ¹⁴	67,5 %	12,5 %	3,4 %	7,0 %	1,5 %	3,8 %	0,6 %	3,6 %
W3Counter ¹⁵	63,8 %	17,9 %	4,6 %	5,6 %	1,6 %	2,7 %	NC	3,8 %
Moyenne	65,8 %	15,9 %	4,0 %	5,7 %	1,7 %	3,2 %	0,6 %	3,1 %



1.3.3 Le Web sémantique

Il s'agit d'optimiser la structure du document HTML afin qu'il soit clairement identifiable par les "user-agent", ces robots (bots) sur différents types de machines, mais aussi plus intelligible par les différents groupes d'utilisateurs.

Selon le W3C, « le Web sémantique fournit un [modèle](#) qui permet aux données d'être partagées et réutilisées entre plusieurs applications, entreprises et groupes d'utilisateurs »³. L'expression a été inventée par [Tim Berners-Lee](#)⁴ (inventeur du Web et directeur du W3C). Wikipedia

Les moteurs de recherche :

130 mille milliards de pages présentes dans l'index de Google.

En France Google concentre plus de 90% des requêtes effectuées sur les moteurs de recherche, dont plus de la moitié sont effectuées sur des smartphones..

Le reste est partagé entre Bing de Microsoft, Yahoo, Ask, Duckduckgo, Qwant...

Ce chiffre tombe à 60% niveau mondial, notamment avec le russe Yandex et le chinois Baidu.

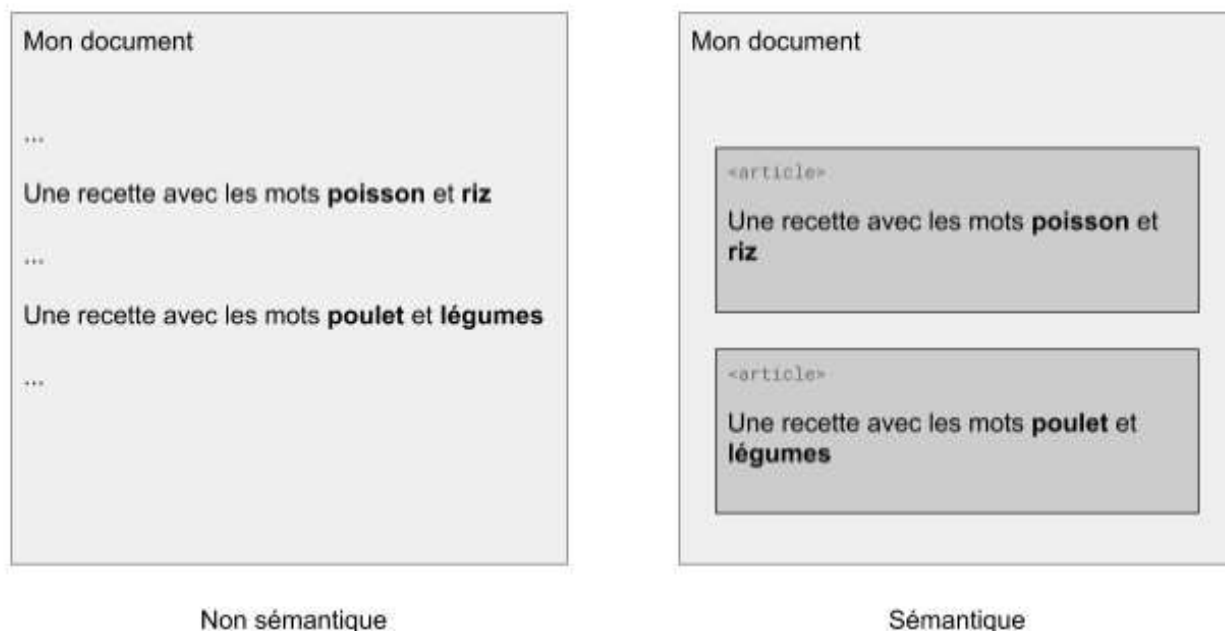
Dans une page de résultat (SERP), Google affiche en général dix extraits (snippets).

Les textes affichés correspondent à vos balises "title" et "meta description", et l'URL.

Lorsqu'on recherche un nom de marque, le snippet affiche en plus un "sitelink" qui correspond aux pages les plus représentatives du site.



Nous verrons en fin de cours l'optimisation du référencement naturel, le "SEO".



Le bénéfice d'un document sémantique se retrouve lors d'une recherche sur un moteur de recherche.

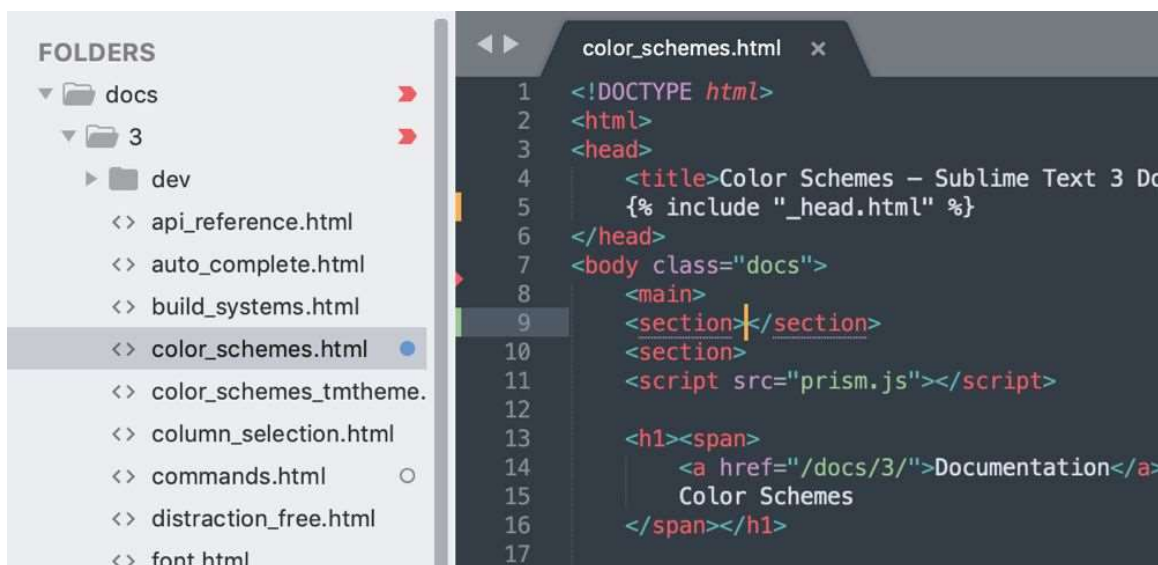
Termes recherchés	Document non sémantique	Document sémantique
poisson + légumes	apparaît dans les résultats comme pertinent l'internaute n'est pas satisfait du résultat	n'apparaît pas dans les premiers résultats l'internaute a peu de chance d'ouvrir le document
poisson + riz	apparaît dans les résultats comme pertinent l'internaute est satisfait du résultat	apparaît dans les résultats comme très pertinent l'internaute est satisfait du résultat

2. Mise en pratique

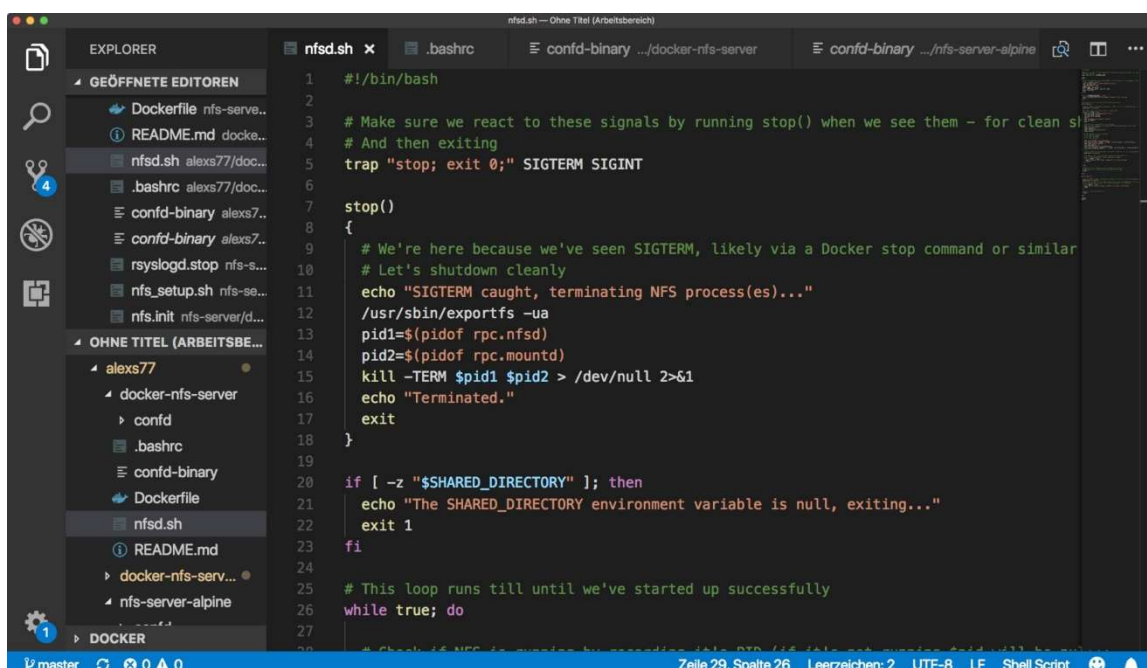
2.1 Choix d'un éditeur de texte

Il existe de nombreux éditeurs de texte gratuits ou payants (**Visual Studio Code**, **Sublime Text**, **Brackets**, **Atom**, **PHPStorm**...). Choisissez un éditeur ou un environnement de développement qui vous convienne.

2.1.1 Sublime Text



2.1.2 Visual Studio Code

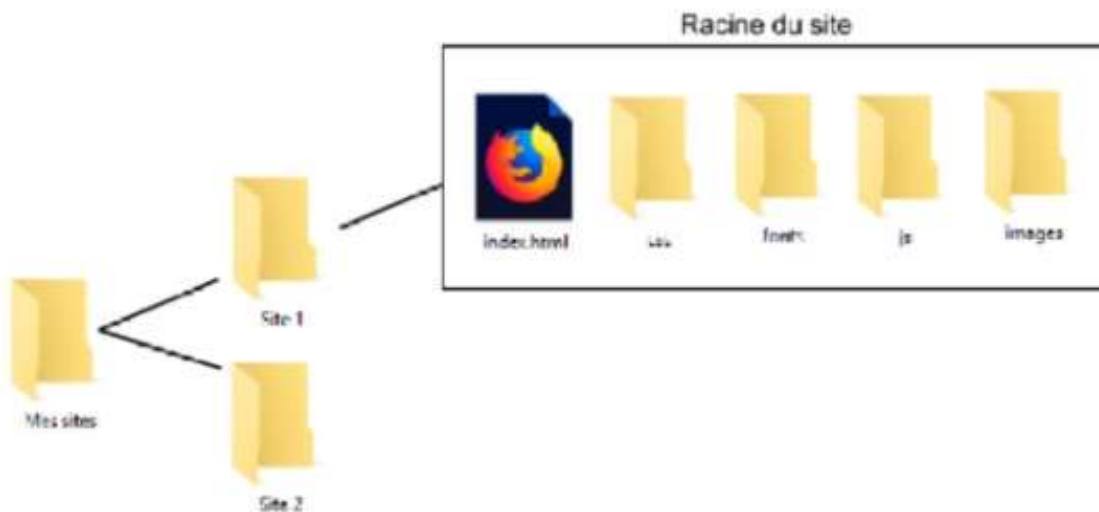


2.2 Création du répertoire de travail

Création du répertoire de travail :

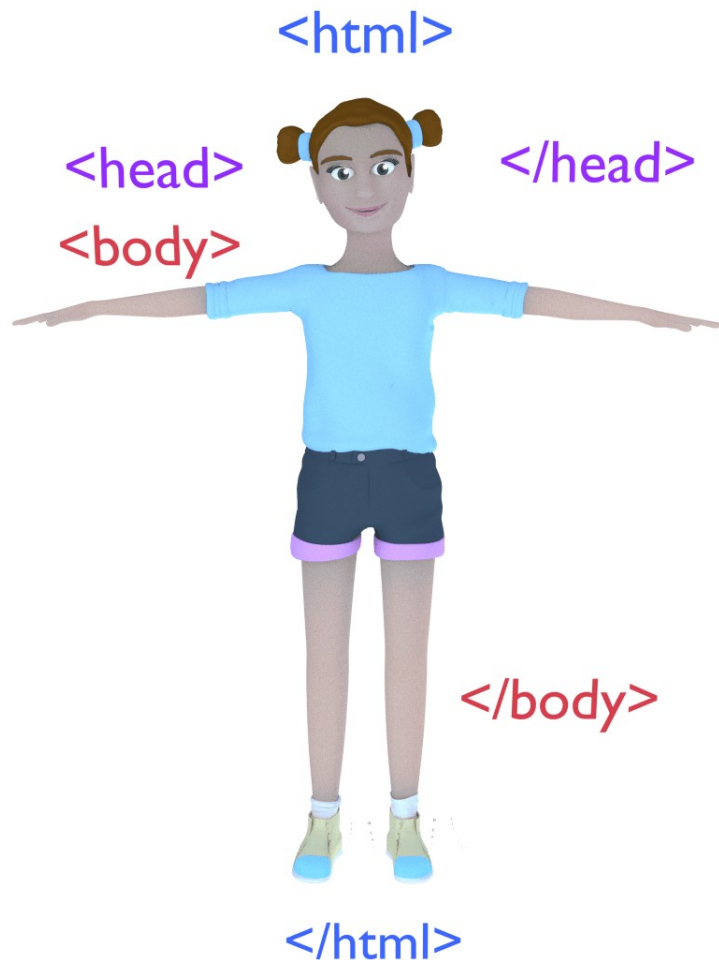
Dans un premier temps, il faut créer un répertoire globale intitulé « sites » ou « mes sites »

Ensuite, créer un sous-répertoire par site internet.



2.3 Création de la première page internet

2.3.1 Structure d'une page internet de base



2.3.2 Les balises

Comme nous l'avons vu précédemment, HTML est un langage qui utilise des balises. Ces balises s'ouvrent avec un chevron ouvrant `<` et se ferment avec un chevron fermant `>` :

`<balise>`

Il existe deux types de balises, les **balises en paires**:

`<title>Titre de ma page</title>`

Et les **balises orphelines** (ou auto-fermantes)

``

Dans les deux cas une balise doit toujours être fermée par un `/` (slash)

En HTML comme dans de nombreux langages de programmation, le code se structure par **niveaux hiérarchiques** avec des **éléments s'imbriquant** les uns dans les autres :

Exemple de niveaux hiérarchiques		
1 ^{er} niveau Parents	2 ^{ème} niveau Enfants directs	3 ^{ème} niveau Petits enfants
<code><section></code>		
	<code><h2>1er titre section 1</h2></code>	
	<code><article></code>	
		<code><h2>Titre de l'article 1</h2></code>
		<code><p></code> Lorem ipsum dolor sit amet, consectetur <code></p></code>
	<code></article></code>	
	<code><h2>2ème titre section 1</h2></code>	
	<code><article></code>	
		<code><h2>Titre de l'article 2</h2></code>
		<code><p></code> Lorem ipsum dolor sit amet, consectetur <code></p></code>
	<code></article></code>	
<code></section></code>		

2.4 Structuration sémantique des pages HTML

Une balise sémantique est une balise qui donne une indication sur le contenu qu'elle contient. Comprenez que les balises `<div>` ou `` ne peuvent donner aucune indication sur les éléments qu'elles contiennent du fait que la documentation HTML ne donne pratiquement aucune limite au type de contenu possible entre ces balises, ce ne sont donc pas des balises sémantiques.

Le HTML5 nous offre justement une série de nouveaux éléments dit « structurants » et qui vont nous permettre de préciser le sens de certains contenus. Ces éléments sont dits « structurants » puisqu'ils ont une visée purement sémantique.

2.4.1 Les balises sémantiques

Nom de l'élément	Description
header	Représente l'en-tête ou le haut d'une page
nav	Représente un menu de navigation dans une page
main	Représente le contenu principal de l'élément <code>body</code>
section	Représente une section dans une page, c'est-à-dire un groupement thématique de contenu
article	Représente une partie de page qui se suffit à elle-même, comme un post sur un blog par exemple
aside	Représente une partie de contenu non lié au reste
footer	Représente le pied d'une page

Autres balises sémantiques

figure	Cette balise permet de déclarer une figure, c'est à dire un média, une image, une vidéo associée à une description. La description doit être placée dans la balise figcaption.
figcaption	Cette balise permet de déclarer la description associée à une figure.
time	Cette balise permet de déclarer une date ou une date avec l'heure avec sa valeur compréhensible par les autres machines: Le <code><time datetime="2018-01-01">jour de l'an</time></code>
mark	Cette balise permet de mettre du texte en surbrillance par rapport au reste du contenu.



Exemple de structuration du contenu avec les balises sémantiques

```

<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset="utf-8">
    <meta name="viewport"
      content="width=device-width,initial-scale=1.0,user-scalable=no">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <!--En tête de la page-->
    <header>
      
      <nav><!--Menu de navigation principal--></nav>
    </header>

    <h1>Titre principal de ma page</h1>

    <!--Contenu qui se suffit à lui même-->
    <article>
      <header>
        <nav><!--Menu de navigation interne--></nav>
      </header>
      <article>
        <h1>Titre de l'article</h1>
        <p>Contenu de l'article</p>
      </article>
      <footer><!--Infos sur l'auteur, etc.--></footer>
    </article>

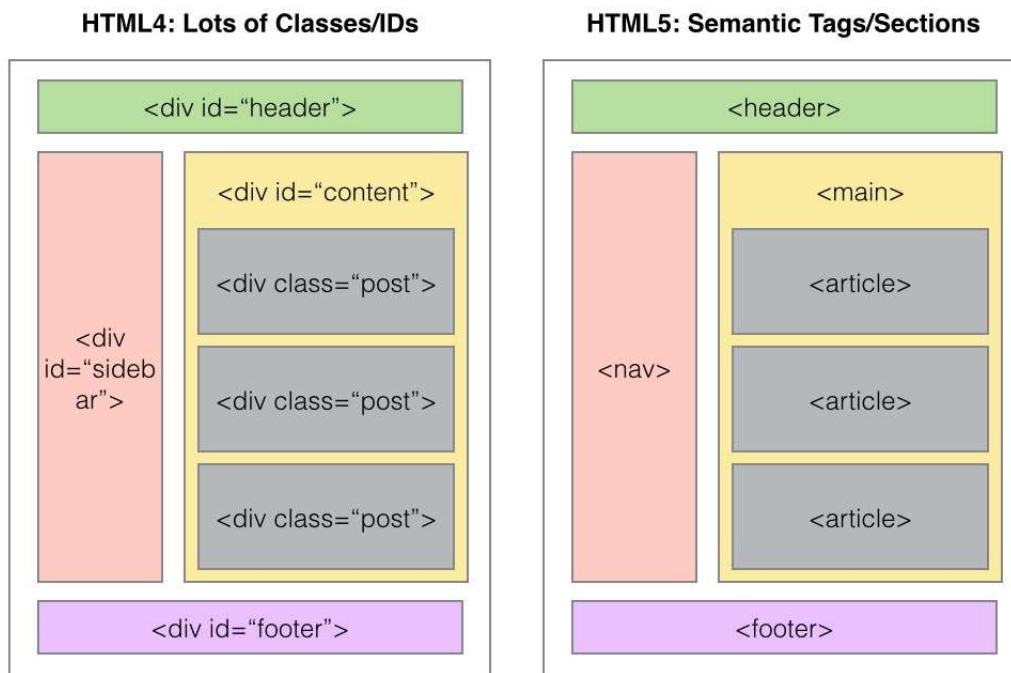
    <!--Autre contenu pas forcément pertinent hors contexte-->
    <section>
      <aside><!--Une publicité--></aside>
      <aside><!--Liens vers les réseaux sociaux--></aside>
    </section>

    <!--Pied de la page-->
    <footer>
      <p>© Pierre Giraud / https://www.pierre-giraud.com</p>
    </footer>
  </body>
</html>

```


2.4.2 HTML4 vs HTML5

HTML4 vs HTML5 Page Structure on a Blog



2.4.3 Les attributs

Les attributs sont des informations complémentaires qui sont ajoutées au sein de la balise, ils sont généralement suivis d'une valeur :

<balise attribut="valeur"></balise>

Par exemple, pour une image, il faut indiquer la source du fichier que l'on souhaite afficher :

La majorité balises fonctionnent en paires : une balise ouvrante **<p>** et une balise fermante **</p>**

D'autres balises sont auto fermantes ou dites orphelines :

**
, <hr/>, **

2.4.4 Première ligne de codes

- Le **doctype** spécifie la version de html (ici HTML5)
- L'en-tête **<head>** permet de renseigner des informations qui ne seront pas visibles sur la page
- Le corps de la page **<body>** contient le contenu visible
- Le code est **indenté**, les balises sont en retrait de leurs parentes
- Le **<title>** est utilisé par le navigateur (onglet) et par les moteurs de recherche (titre d'un résultat de recherche)
- Le **<charset>** correspond au jeu de caractère du fichier. Il détermine comment les caractères spéciaux vont s'afficher (accents, idéogrammes chinois et japonais, caractères arabes, etc.).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ma première page HTML</title>
    <meta charset="utf-8">
  </head>
  <body>
    </body>
</html>
```

Structure de votre page html

Doctype :

HTML4

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
```

HTML5

Le doctype est toujours le même² :

```
<!DOCTYPE html>
```


2.4.5 L'indentation

Pour faciliter la compréhension du code, il est primordial de respecter les imbrications et l'indentation. L'idée est de rendre le code plus lisible et maintenable. Pour cela nous écrivons le plus à gauche les conteneurs et nous décalons d'une tabulation les éléments contenus :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>

    <div>
      <p>Un premier paragraphe</p>
      <p>Un autre paragraphe</p>

      <ul>
        <li>Un élément de liste</li>
        <li>Un autre élément de liste</li>
      </ul>
    </div>

    <p>Un troisième paragraphe</p>

    <div>
      <p>Un dernier paragraphe</p>
    </div>
  </body>
</html>
```

2.4.6 Les commentaires

Comme la plupart des langages, HTML permet d'inscrire des commentaires. En revanche, les commentaires monolignes et multilignes se font de la même manière :

Il faut d'abord préciser le début du commentaire par **<!--** - suivi du texte et terminer par : **-->**

Un commentaire doit être compris entre <!-- et -->

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ma première page HTML</title>
    <meta charset="utf-8">
  </head>
  <body>
    <!--Deux titres h1 et h2-->
    <h1>Mon titre principal</h1>
    <h2>Je suis un titre important</h2>

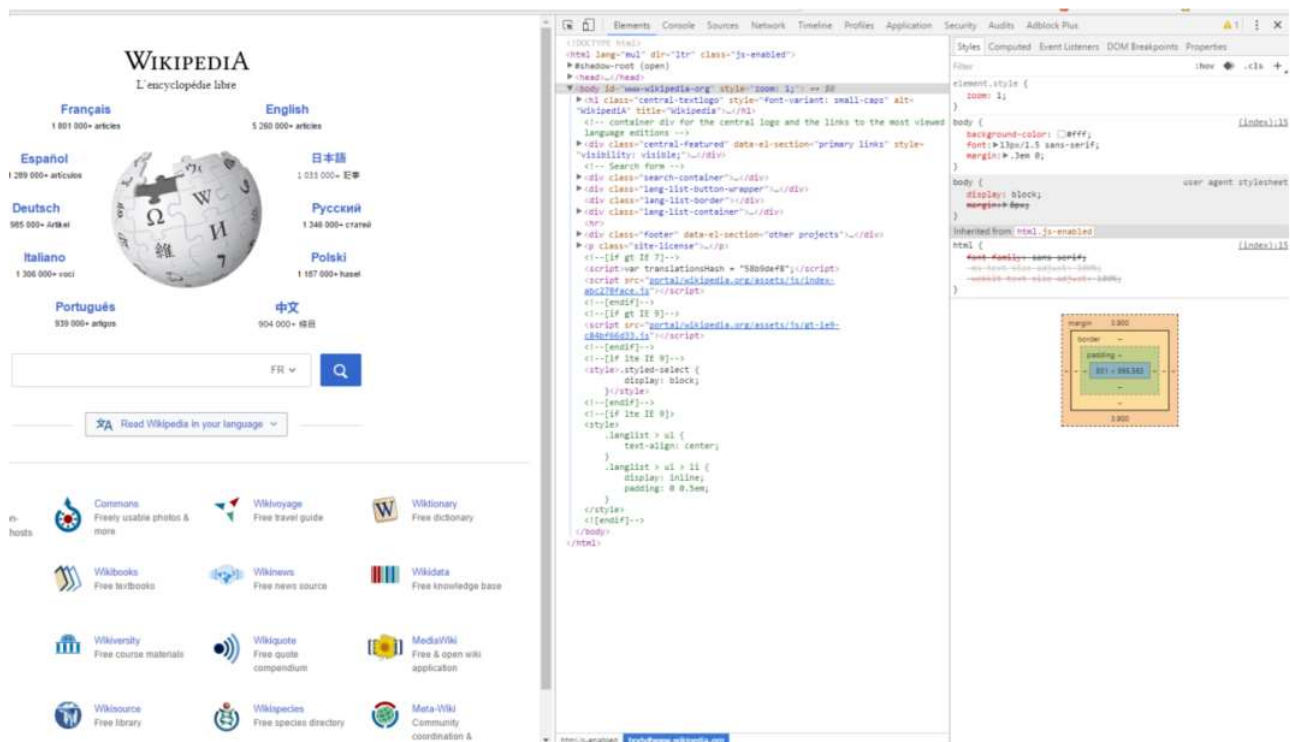
    <!--Deux paragraphes différents-->
    <p>Voici mon premier paragraphe.</p>
    <p>Et en voilà un second !</p>
  </body>
</html>
```

2.4.7. L'inspecteur et le cache

a- L'inspecteur

L'inspecteur est un outil intégré dans la plupart des navigateurs, c'est un outil très utile pour déceler les bugs. Il est notamment présent au sein des navigateurs Firefox, Brave, Chrome, Edge, Opera...

L'inspecteur affiche le code HTML / CSS et JavaScript.



b- Ouverture de l'inspecteur

L'inspecteur du navigateur web s'ouvre :

Navigateur	Méthodes d'ouverture
Firefox	F12 ou CTRL+MAJ+I Clic droit sur la page Web > inspecter menu Firefox > Outils supplémentaires > Outils de développement
Chrome	F12 ou CTRL+MAJ+I Clic droit sur la page Web > inspecter
Opera	CTRL+MAJ+I Clic droit sur la page Web > inspecter élément
Edge	F12 ou CTRL+MAJ+I Clic droit sur la page Web > inspecter menu Firefox > Outils supplémentaires > Outils de développement
Safari	<ol style="list-style-type: none">1. Activer le menu développeur dans les Préférences de Safari, onglet Avancé.2. Command+ Option+ I.

Linus Torvalds

113 langues

Linus Benedict Torvalds, né le 28 décembre 1969 à Helsinki en Finlande, est un informaticien américano-finlandais^{2,3}.

Il est notamment connu pour avoir créé en 1991 (à 21 ans) le noyau Linux, dont il continue de diriger le développement, étant considéré comme le « dictateur bienveillant à vie » (*Benevolent Dictator for Life*) de celui-ci. Il a également créé le logiciel de gestion de versions décentralisée Git et le logiciel d'enregistrement et de planification des plongées Subsurface (en).

En 2012, il a été honoré avec Shinya Yamanaka du prix Millennium Technology, décerné par la Technology Academy Finland (en) « en reconnaissance pour sa création d'un nouveau système d'exploitation open source pour les ordinateurs ayant conduit au noyau Linux, largement utilisé⁴ ». Il est également lauréat en 2014 du Computer Pioneer Award, décerné par l'IEEE Computer Society⁵.

Sommaire [masquer]

- Biographie
 - Jeunesse
 - Études
 - Carrière
 - Vie personnelle
- Travail sur le noyau Linux
 - Débuts du développement
 - Essor de GNU/Linux
- Publications
- Distinctions
- Notes et références
- Annexes
 - Articles connexes
 - Liens externes

Biographie [modifier | modifier le code]

Jeunesse [modifier | modifier le code]

Linus Torvalds fait partie de la communauté des Finlandais suédophones (*Finlandssvensk*), une population représentant 6 % des habitants de la Finlande. Son père, Nils Torvalds, est journaliste de télévision et de radio et aussi membre actif du Parti populaire suédois (le parti

Linus Torvalds

←

→

↺

☆

Enregistrer sous...

Enregistrer la page dans Pocket

Tout sélectionner

Effectuer une capture d'écran

Code source de la page

Inspecter les propriétés d'accessibilité

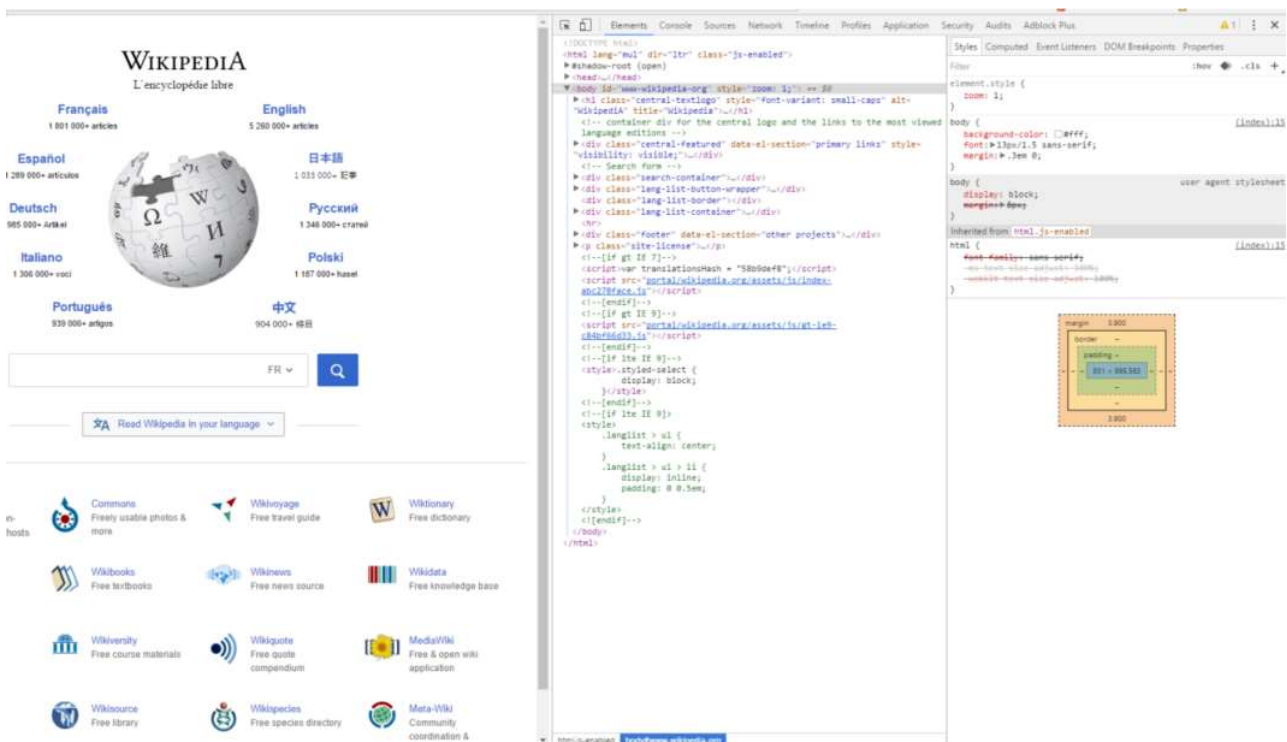
Inspecter

Page Captures d'écran Web - Fireshot >

Video DownloadHelper >

Bloquer un élément

Naissance	28 décembre 1969 (52 ans) <div>Helsinki </div>
Nom de naissance	Linus Benedict Torvalds
Nationalités	Finlandaise <div></div> Américaine (depuis septembre 2010) <div></div>
Domicile	Dunthorpe (en) (depuis 2004)
Formation	Université d'Helsinki (maîtrise) (1988-1996)
Activités	Programmeur, développeur
Père	Nils Torvalds
Mère	Anna Torvalds (d)
Fratie	Leo Torvalds (d) Alexander Torvalds (d) Christopher Torvalds (d) Sara Torvalds (d)



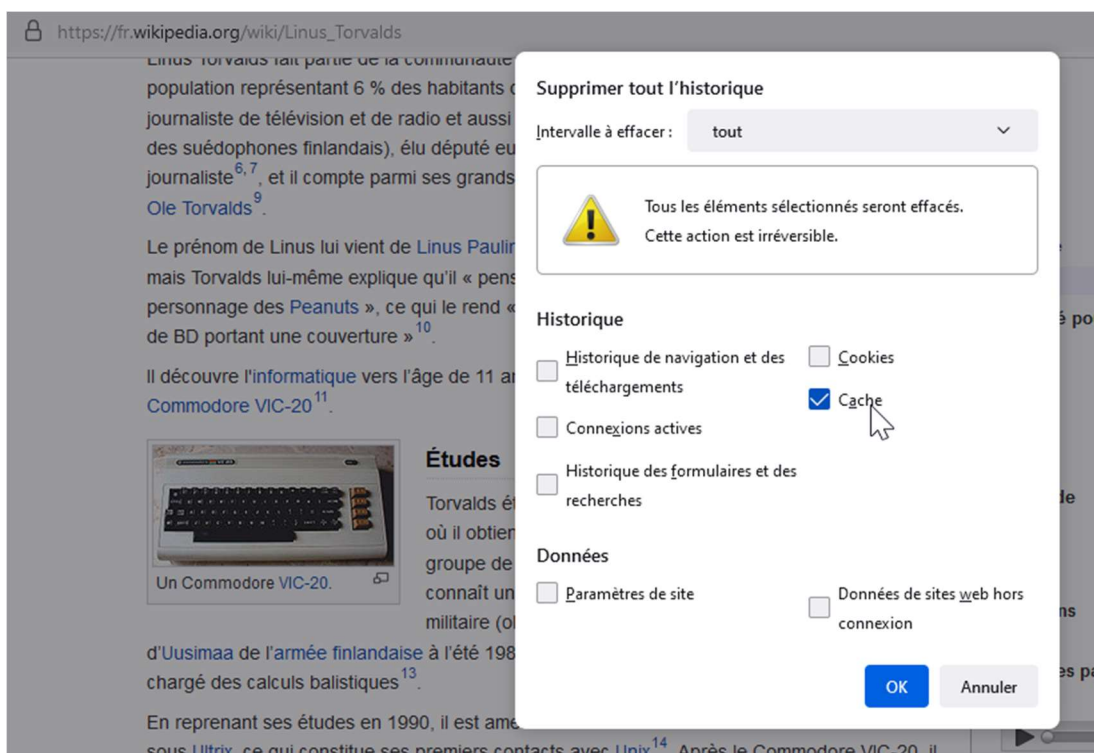
c- Le cache

Une des erreurs fréquentes est de penser que les modifications apportées au CSS n'ont pas fonctionné car elles n'apparaissent pas à l'écran. Les navigateurs pour accélérer l'affichage des pages mettent en cache certaines données, notamment celles relatives au CSS.

Le cache est une partie de la **mémoire vive** très rapide et volatile en opposition à la mémoire morte qui est bien plus lente, mais qui conserve les données après extinction de l'ordinateur. L'usage du cache permet à l'ordinateur un accès plus rapide aux données qu'il va fréquemment utiliser le temps de la session utilisateur.

Utilisée dans le cadre du CSS, le cache accélérera notamment l'affichage. Néanmoins, il est possible que certaines anciennes données restent en cache et que les modifications effectuées ne soient pas effectives. Dans ce cas, il est nécessaire de **vider le cache**.

Pour Firefox, Chrome et Edge la combinaison de touche **ctrl+alt+suppr** permet d'accéder au menu de suppression du cache :



2.4.8 Les titres, paragraphes et sauts de ligne

Les « titres » sont inscrits entre les balises `<h1>`, `<h2>`, `<h3>`... `<h6>` (de la plus grande taille à la plus petite) en ouverture, et `</h1>`... `</h6>` en fermeture.

Les paragraphes sont compris entre les balises `<p>` et `</p>`.

Chaque fois que l'on ferme une balise de « titre » ou un paragraphe un saut de ligne est automatiquement créé.

Indépendamment de ces balises, il est possible de faire un saut de ligne avec la balise `
`, mais c'est une mauvaise pratique.

2.4.9 Emphase, gras et italique

 ou <i> ?

La balise **** sert à faire une emphase, c'est-à-dire mettre en valeur une information, c'est le navigateur qui la représente par défaut en italique. Pour mettre en italique il faut privilégier la balise **<i>**.

Néanmoins, il faut privilégier le CSS pour effectuer la mise en forme.

La balise **** n'a pas pour objectif de mettre en gras, mais d'indiquer une **haute importance** au texte sélectionné.

2.4.10 Listes ordonnées et listes non ordonnées

ol = Ordered List. => Liste Ordonnée (Chiffres décimaux, chiffres romains, lettres...)

ul = Unordered List => Liste Non Ordonnée

Pour créer une liste d'éléments il faut combiner deux balises :

- **** ou **** pour ouvrir et fermer la liste
- **** pour chaque élément de liste

Pour une liste à puces :

```
<ul>  
  <li>Pomme</li>  
  <li>Fraise</li>  
  <li>Kiwi</li>  
</ul>
```

Pour une liste ordonnée :

```
<ol>  
  <li>Pomme</li>  
  <li>Fraise</li>  
  <li>Kiwi</li>  
</ol>
```


Attention, si l'on souhaite faire des listes imbriquées, il faut inclure totalement la liste imbriquée dans le parent :

```
<ul>
  <li>1</li>
  <li>2
    <ul>
      <li>2.1</li>
      <li>2.2</li>
    </ul>
  </li>
  <li>3</li>
</ul>
```

Exercice d'application : classement meilleurs joueurs / joueuses

Créer une page internet avec :

En entête « Classement » (balise head)

En titre (H1) : Liste de constructeurs d'automobiles :

Reproduire la liste suivante :

1. Renault
2. Peugeot
3. Audi
4. BMW
5. Citroen
6. Fiat
7. Mercedes
8. Opel
9. Dacia
10. Ford
11. Nissa
12. Toyota

Correction :

```
<ol>
  <li>Renault</li>
  <li>Peugeot</li>
  <li>Audi</li>
  <li>BMW</li>
  <li>Citroen</li>
  <li>Fiat</li>
  <li>Mercedes</li>
  <li>Opel</li>
  <li>Dacia</li>
  <li>Ford</li>
  <li>Nissa</li>
  <li>Toyota</li>
</ol>
```

Exercice 2 :

Faire un classement des 10 meilleurs joueurs de foot / basket / ou tout autre discipline de votre choix.

Correction:

```
<> championnat.html > html > body > ol
1  <!DOCTYPE fr>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Classement</title>
7  </head>
8  <body>
9    <h1>Classement des meilleurs joueurs de football de tous les temps</h1>
10
11    <ol>
12      <li>Pelé</li>
13      <li>Diego Maradona</li>
14      <li>Lionnel Messi</li>
15      <li>Christiano Ronaldo</li>
16      <li>Zinedine Zidane</li>
17      <li>Ronaldo</li>
18      <li>Rivaldo</li>
19      <li>Samuel Eto'o</li>
20      <li>Arjen Robben</li>
21      <li>Alan Shearer</li>
22    </ol>
23
24  </body>
25  </html>
```

Exercice – reproduire la liste suivante

1. Renault
 - France
 - Clio
 - 1990
2. Peugeot
 - France
 - 206
 - 1998
3. Audi
 - Allemagne
 - A4
 - 1994
4. BMW
 - Allemagne
 - Serie 3
 - 1994
5. Toyota
 - Japon
 - Corolla
 - 1966
6. Kia
 - Corée du Sud
 - Sportage
 - 1994
7. Mobius Motors
 - Kenya
 - Mobius 1
 - 2010

Correction :

```
<ol>
  <li>Renault
    <ul>
      <li>France</li>
      <li>Clio</li>
      <li>1990</li>
    </ul>
  </li>
  <li>Peugeot
    <ul>
      <li>France</li>
      <li>206</li>
      <li>1998</li>
    </ul>
  </li>
  <li>Audi
    <ul>
      <li>Allemagne</li>
      <li>A4</li>
      <li>1994</li>
    </ul>
  </li>
  <li>BMW
    <ul>
      <li>Allemagne</li>
      <li>Serie 3</li>
      <li>1994</li>
    </ul>
  </li>
  <li>Toyota
    <ul>
      <li>Japon</li>
      <li>Corolla</li>
      <li>1966</li>
    </ul>
  </li>
  <li>Kia
    <ul>
      <li>Corée du Sud</li>
      <li>Sportage</li>
      <li>1994</li>
    </ul>
  </li>
  <li>Mobius Motors
    <ul>
      <li>Kenya</li>
      <li>Mobius 1</li>
      <li>2010</li>
    </ul>
  </li>
</ol>
```

2.4.11 Les listes de définitions

Pour créer une liste de définitions 3 balises sont nécessaires :

<dl> </dl> => Definition List. <dl> ouvre la liste de définitions.

<dt></dt> => Definition Term. Entre ces balises le terme qui sera défini.

<dd> </dd> => Definition Description. Dans cette partie est écrit le sens du terme, la définition.

```
<dl>
  <dt>CMS</dt>
  <dd>
    CMS : Content Managemennt System, est un SGC (Système de Gestion de Contenu)
  </dd>
</dl>
```

Résumé

- Le HTML comporte de nombreuses balises qui permettent d'organiser la page ou d'y ajouter des éléments.
- Ces balises en plus de structurer la page, donnent des indications sur leur contenu.
- Les paragraphes sont définis par la balise <p></p>
- Les titres par les balises <h1> à <h6>
- Les sauts de lignes par la balises orpheline

- On peut mettre certain mots en valeurs avec les balises et
- Les listes sont créés grâce à des balises englobées dans une balise ou .

2.5 Insertion liens, ancrs, et images

Les liens

Le lien vers un autre site est également créé via une balise. C'est une balise qui comportera à chaque fois au moins un attribut puisqu'il faut toujours renseigner une destination.

Exemple un lien vers google :

```
<a href="http://www.google.fr">Cliquez ici</a>
```

L'attribut href permet de définir la cible.

Le contenu de la balise permet de définir l'intitulé.

Ouvrir un lien dans un nouvel onglet

On va utiliser l'attribut « **target** » et lui attribuer la valeur « **_blank** »

```
<a target="_blank" href="http://www.google.fr">Cliquez ici</a>
```

Lien vers une adresse email

Il faut utiliser : **mailto** :

```
<a href="mailto:alex.formation@gooddle.co">Cliquez ici</a>
```

Lien vers un numéro de téléphone

Utilisation de **tel:xxxxxxxxxxx**

```
<a href="tel:+1800229933 ">Cliquez ici</a>
```

Les liens relatifs

Lien vers une page interne au site:

```
<a href="_contact.html">Contactez-nous</a>
```

Lien vers une page interne mais dans un autre dossier:

```
<a href="_dossier/contact.html">Contactez-nous</a>
```

Si le lien est dans un dossier parent:

```
<a href="../contact.html">Contactez-nous</a>
```

Les ancres

Une ancre est une partie définie de la page courante, pour définir une ancre il faut renseigner un id. Créer un lien vers une ancre est particulièrement utile pour effectuer un lien au sein d'une même page ou sur une partie précise d'une autre page.

```
<p id="description-produit">Ce produit est génial...</p>
```

Lien vers l'ancre "description-produit":

```
<a href="#description-produit">Voir la description</a>
```


Le JPEG (*Joint Photographic Expert Group*)

- Format conçu pour réduire le poids des photo
- Qualité très fidèle à l'original
- Bon rendement de compression
- En revanche il est peu performant sur les aplats (c'est pourquoi il est plus utilisé pour les photos que les images)

Le PNG (*Portable Network Graphics*)

- C'est le format le plus récent
- Il est adapté à toutes les images mais pas aux photos
- Il gère la transparence et n'altère pas la qualité de l'image

Le GIF

- Très lourd
- Ne gère que 256 couleurs
- Peut être animé (mais doucement hein) !

A ces formats reconnus depuis de nombreuses années, il faut mentionner le format **WebP** mis en place par Google. Ce format permettrait de réduire le temps d'affichage des images.

Insérer une image

Il faut utiliser la balise autofermante `img` et préciser l'attribut `alt`

```

```

2.6 Les tableaux

Avant le développement du CSS, les tableaux ont été en dehors de leur fonction d'affichage de données tabulaires, une manière de structurer les pages internet. Désormais, seule leur fonction première est retenue.

<table> est la première balise d'un tableau, elle regroupe l'ensemble des lignes et des cellules.

<tr> regroupe les cellules d'une même ligne.

<td> regroupe le contenu d'une cellule de tableau. Contient les attributs "**colspan**" (cellules fusionnées en colonne) et "**rowspan**" (cellules fusionnées en ligne).

Les lignes **<tr>** et les cellules **<td>** suffisent à créer un tableau mais on sémantise avec :

<thead> correspond à l'en-tête du tableau, regroupe les cellules d'en-tête du tableau.

<th> indique une cellule d'en-tête, correspond au **<td>** du **<thead>**

<tbody> regroupe les cellules du corps du tableau.

<tfoot> correspond au pied du tableau, regroupe les cellules du pied du tableau.

<caption> placé juste après **<table>** indique la légende associée au tableau.

<col> délimite la ou les colonnes d'un groupe de colonnes

<colgroup> contient le ou les **<col>** (colonnes du tableau), il se place après **<caption>** et avant **<thead>**, **<tbody>**, **<tfoot>** et **<tr>**. Ces balises permettent d'appliquer des styles sur une colonne.

```
<table>
  <caption>Liste des joueurs de ligue 1</caption>
  <thead>
    <tr>
      <th>ID</th>
      <th>Nom</th>
      <th>Prénom</th>
      <th>Club</th>
      <th>Pays</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Mbappé</td>
      <td>Kylian</td>
      <td>PSG</td>
      <td>France</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Payet</td>
      <td>Dimitri</td>
      <td>OM</td>
      <td>France</td>
    </tr>
  </tbody>
```



```

<table>
  <caption>Liste des joueurs de ligue 1</caption>
  <thead>
    <tr>
      <th>ID</th>
      <th>Nom</th>
      <th>Prénom</th>
      <th>Club</th>
      <th>Pays</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Mbappé</td>
      <td>Kylian</td>
      <td>PSG</td>
      <td>France</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Payet</td>
      <td>Dimitri</td>
      <td>OM</td>
      <td>France</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="4">Nombre de joueurs</td>
      <td>2</td>
    </tr>
  </tfoot>
</table>

```

Les attributs **colspan** et **rowspan** permettent de fusionner plusieurs cases du tableau :

- **colspan** horizontalement.
- **rowspan** verticalement

Liste joueurs football / Rugby

ID	Nom	Prénom	Ville	Pays
1	Mbappe	Kylian	Paris	France
2	Dupont	Antoine	Toulouse	France
3	Depay	Memphis	Barcelone	Espagne
4	Benzema	Karim	Madrid	Espagne
5	Griezman	Antoine	Madrid	Espagne

Mon premier tableau

ID	Nom	Prénom	Ville	Pays
1	Thuram	Marcus	Paris	France
2	Dupont	Antoine	Toulouse	
3	Depay	Memphis	Barcelone	Espagne
4	Benzema	Karim	Madrid	
5	Griezman	Antoine	Madrid	
Nombre de joueurs				5

Il n'est possible de ne fusionner que les cases d'une même section (thead, tbody, tfoot).

Aussi, vous tentez de fusionner des cases appartenant à des sections différentes, cela ne fonctionnera pas.

```
<table>
  <thead>
    <tr>
      <th>Entête de colonne 1</th>
      <th>Entête de colonne 2</th>
      <th>Entête de colonne 3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="2">Fusion de 2 cases verticalement</td>
      <td colspan="2">Fusion de 2 cases horizontalement</td>
    </tr>
    <tr>
      <td>Ligne 3, colonne 2</td>
      <td>Ligne 3, colonne 3</td>
    </tr>
  </tbody>
</table>
```

Ce qui donne :

Entête de colonne 1	Entête de colonne 2	Entête de colonne 3
Fusion de 2 cases verticalement	Fusion de 2 cases horizontalement	
	Ligne 3, colonne 2	Ligne 3, colonne 3

Résumé

- Les liens permettent de changer de page et sont, par défaut, écrits en bleu et soulignés.
- Pour insérer un lien, on utilise la balise `<a>` avec l'attribut `href` pour indiquer l'adresse de la page cible. Exemple : ``.
- On peut faire un lien vers une page interne d'un site en utilisant simplement: ``.
- Les liens permettent aussi d'amener vers d'autres endroits sur la même page. Il faut créer une ancre avec l'attribut `id` pour « marquer » un endroit dans la page, puis faire un lien vers l'ancre comme ceci : ``.