


WIM 1.1

LANGAGE HTML

Denis Monnerat


monnerat@u-pec.fr 

Updated: 2017/11/06

IUT de Fontainebleau

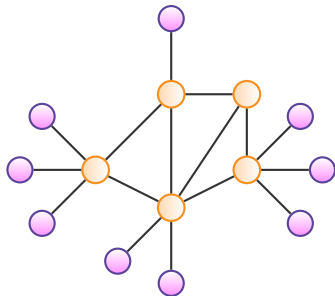


1. Introduction
2. HTML
3. Principaux éléments
4. Index des balises html(5)

- Réseau de réseaux
- Applications offrant des services + protocoles associés, basées sur TCP/IP.
- Mode Client/Serveur
- Comprend, entre autre :
 - Couche réseau (transport)
 - Gestion des noms et adresses
 - Outils et protocoles dédiés
 - Le(s) langage(s) HTML
- Beaucoup de technologies.
- Normalisation par le World Wide Web Consortium 

Pourquoi faire ?

- Recherche d'informations !
- Communication entre les gens.
- Commerce électronique, vente aux enchères.
- Gestion de comptes en banques.
- Démarches administratives (impôts, etc...)
- Peer-to-peer
- Téléphonie, visiophonie, radio, vidéos, télévision, ...
- Enseignement, travail à distance.



○ Permet d'accéder à des documents liés entre eux, sur des machines différentes.

○ Architecture basée sur :

La localisation ~> URL

Le protocole ~> HTTP

Le langage ~> HTML

○ Succès grâce :

○ Interfaces graphiques conviviales.

○ Grande diversité et quantité d'information.

○ Multi-platerformes.

○ Écosystème naturel de plate-formes logicielles comme **TIZEN** ou



Interconnection d'applications de toutes sortes, sur toutes platerformes et tournant sur tout type de matériel.

Comment ça marche ?

1. Le client envoie une requête HTTP au serveur web.



Comment ça marche ?

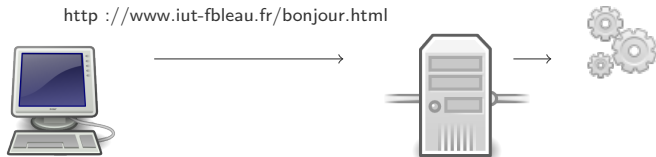
1. Le client envoie une requête HTTP au serveur web.
2. Le serveur vérifie la demande, les autorisations et renvoie la réponse.

`http ://www.iut-fbleau.fr/bonjour.html`



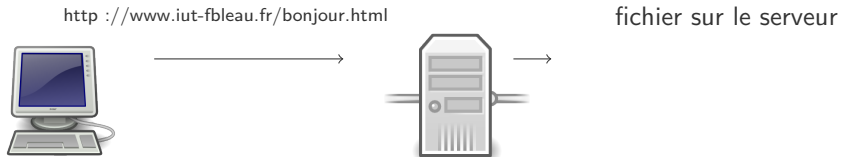
Comment ça marche ?

1. Le client envoie une requête HTTP au serveur web.
2. Le serveur vérifie la demande, les autorisations et renvoie la réponse.



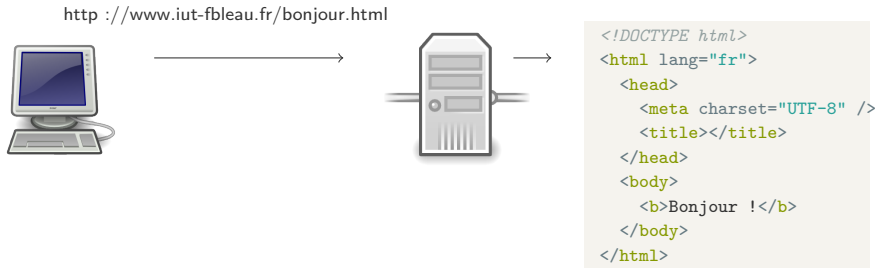
Comment ça marche ?

1. Le client envoie une requête HTTP au serveur web.
2. Le serveur vérifie la demande, les autorisations et renvoie la réponse.



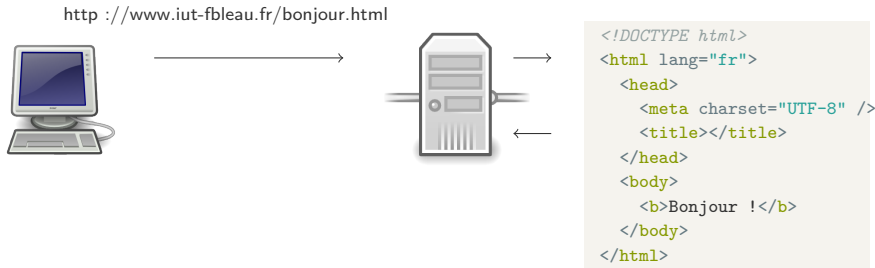
Comment ça marche ?

1. Le client envoie une requête HTTP au serveur web.
2. Le serveur vérifie la demande, les autorisations et renvoie la réponse.



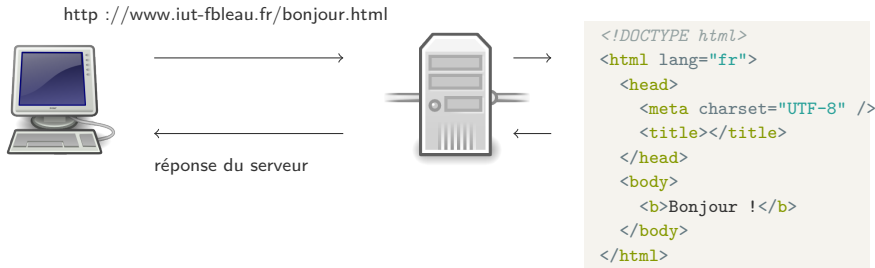
Comment ça marche ?

1. Le client envoie une requête HTTP au serveur web.
2. Le serveur vérifie la demande, les autorisations et renvoie la réponse.



Comment ça marche ?

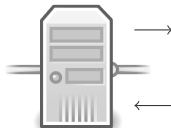
1. Le client envoie une requête HTTP au serveur web.
2. Le serveur vérifie la demande, les autorisations et renvoie la réponse.



Comment ça marche ?

1. Le client envoie une requête HTTP au serveur web.
2. Le serveur vérifie la demande, les autorisations et renvoie la réponse.
3. Le navigateur interprète la réponse reçue et affiche le fichier html.

http ://www.iut-fbleau.fr/bonjour.html

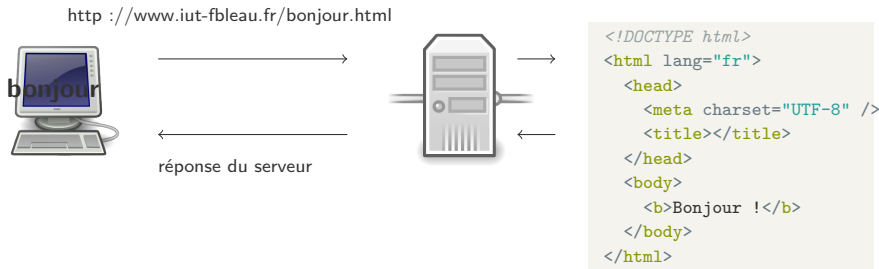


réponse du serveur

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <title></title>
  </head>
  <body>
    <b>Bonjour !</b>
  </body>
</html>
```

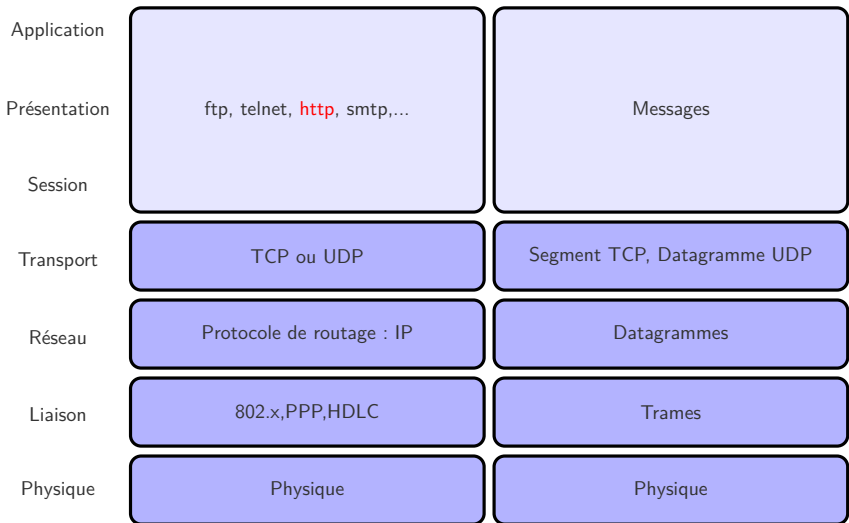
Comment ça marche ?

1. Le client envoie une requête HTTP au serveur web.
2. Le serveur vérifie la demande, les autorisations et renvoie la réponse.
3. Le navigateur interprète la réponse reçue et affiche le fichier html.



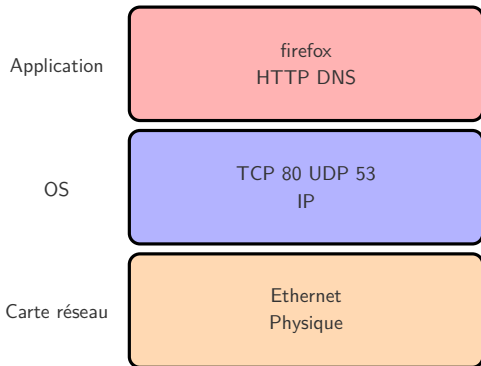
Il peut y avoir en plus :

- Des contrôles par compte, domaine, adresse ip,...
- Des exécutions de codes coté serveur et/ou client.



Application		
Présentation	ftp, telnet, http, smtp,...	Messages
Session		
Transport	TCP ou UDP	Segment TCP, Datagramme UDP
Réseau	Protocole de routage : IP	Datagrammes
Liaison	802.x, PPP, HDLC	Trames
Physique	Physique	Physique

Requête HTTP



File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help



Filter:

Expression...

Clear

Apply

Enregistrer

No.	Time	Source	Destination	Protocol	Length	Info
6	2.307594000	10.11.80.33	37.58.131.227	TCP	66	32985
7	2.307668000	10.11.80.33	37.58.131.227	HTTP	521	GET
8	2.314780000	37.58.131.227	10.11.80.33	TCP	66	http
9	2.315365000	37.58.131.227	10.11.80.33	TCP	1388	[TCP
10	2.315380000	10.11.80.33	37.58.131.227	TCP	66	32985

▶ Frame 7: 521 bytes on wire (4168 bits), 521 bytes captured (4168 bits) on interface 0
▶ Ethernet II, Src: Sony_67:a0:9c (00:1d:ba:67:a0:9c), Dst: Cisco_93:08:00 (00:1d:e5:93:08:00)
▶ Internet Protocol Version 4, Src: 10.11.80.33 (10.11.80.33), Dst: 37.58.131.227 (37.58.131.227)
▶ Transmission Control Protocol, Src Port: 32985 (32985), Dst Port: http (80), Seq: 1, Ack: 1, Len: 455
▶ Hypertext Transfer Protocol
▶ GET / HTTP/1.1\r\n
Host: www.iut-fbleau.fr\r\nUser-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:32.0) Gecko/20100101 Firefox/32.0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept-Language: fr,en-us;q=0.7,en;q=0.3\r\nAccept-Encoding: gzip, deflate\r\nDNT: 1\r\nCookie: txp_remember=0; storedtemplate=calm; style_cookie=null; phpbb3_h7zjb_u=55; phpbb3_h7zjb_k=; phpbb3_h7zjb_l=;\r\nConnection: keep-alive\r\n\r\n[Full request URI: http://www.iut-fbleau.fr/]

```
0000 00 1d e5 93 08 00 00 1d ba 67 a0 9c 08 00 45 00 .....g...E.  
0010 01 fb eb b4 40 00 40 06 c9 ff 0a 0b 50 21 25 3a .....k.@. ...P!%:  
0020 83 e3 80 d9 00 50 cf a3 44 3d b5 8c 99 c0 80 18 .....P.. D=.....  
0030 00 e5 05 37 00 00 01 01 08 0a 00 57 2f 90 61 92 ...7.... .W/.a.  
0040 32 9c 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 2.e.G E / HTTP/1.1  
0050 8d 0e 49 6f 72 74 2a 20 77 77 77 2e 69 75 74 2d .....Host: www.iut-
```

Frame (frame), 521 bytes

Packets: 685 · Displayed: 685 (100,0%)


Profile: Default

Exemple

```
[denis@portable_denis ~]$ telnet www.iut-fbleau.fr 80
Trying 37.58.131.227...
Connected to www.iut-fbleau.fr.
Escape character is '^]'.
GET http://www.iut-fbleau.fr/exemple.html HTTP/1.0
```

```
HTTP/1.1 200 OK
Date: Sun, 21 Sep 2014 19:07:22 GMT
Server: Apache/2.2.23 (Unix) DAV/2 SVN/1.7.6 PHP/5.4.8
Last-Modified: Sun, 21 Sep 2014 18:51:50 GMT
ETag: "e2822-65-50397d3be9ef2"
Accept-Ranges: bytes
Content-Length: 101
Connection: close
Content-Type: text/html
```

```
<html>
<head>
<title>Un exemple</title>
</head>
<body>
<h1>Ceci est un exemple !</p>
</body>
</html>
Connection closed by foreign host.
[denis@portable_denis ~]$
```

Permet la résolution des noms en adresse ip. Par exemple, pour la machine `www.iut-fbleau.fr` , le navigateur interroge un serveur DNS du client (free, etc) qui découpe l'adresse autour des points :

- `fr` : le TLD. Il faut trouver un dns responsable de `.fr`, à savoir l'afnic.
- `iut-fbleau` : un serveur dns de l'afnic donnera l'adresse d'un dns responsable du sous domaine `iut-fbleau`.
- `www` : celui-ci lui donnera l'adresse ip de `www`.

`www.iut-fbleau.fr` est résolue en **37.58.131.227**

Adressage des documents

Solution

- **URL** : *Uniform Resource Locator* = adressage universelle de ressources.
- 3 parties : le protocole (comment), le nom (où) et le nom du document (quoi).
- **URL** \subset **URI** *Universal Resource Identifier*.

Exemples

- `http://www.iut-fbleau.fr/sitebp`
- `https://www.google.fr:8888/img/plan.php?x=12&y=20`
- `ftp://user:password@www.iut-fbleau.fr/account/`

URL

protocole://hostname:port/path/extrapath?arguments

- La racine / de path est définie par la configuration du serveur Web. (rien à voir a priori avec la **racine du système de fichier**)
- path peut contenir un point d'ancrage.
- extrapath et arguments permettent de passer des informations à un programme qui s'exécute sur le serveur (cgi).

Un peu d'histoire


L'aventure a commencé au début des années 1990. Quelques acteurs et inventeurs :

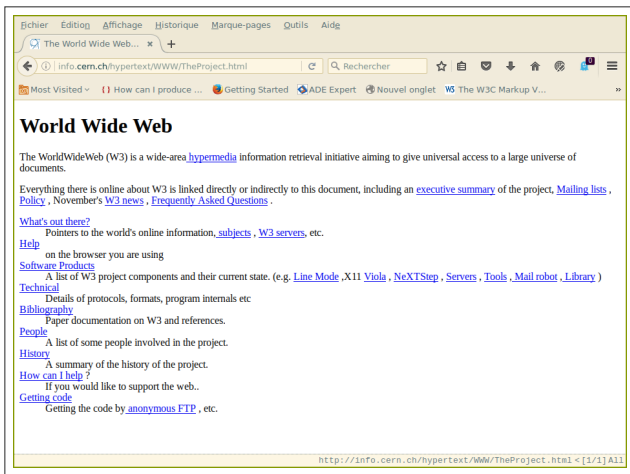
- Tim Berners-Lee
- Le C.E.R.N.
- Mosaic, Netscape



Le premier serveur Web.

Le premier site web

<http://info.cern.ch/hypertext/WWW/TheProject.html> 



The screenshot shows a web browser window with the title "The World Wide Web...". The address bar contains the URL "http://info.cern.ch/hypertext/WWW/TheProject.html". The browser interface includes a menu bar with options like "Fichier", "Edition", "Affichage", "Historique", "Marque-pages", "Outils", and "Aide". The main content area displays the title "World Wide Web" in a large, bold font. Below the title, there is a paragraph of text describing the WorldWideWeb (W3) project. The text includes several hyperlinks in blue, underlined font. At the bottom of the browser window, the status bar shows the current URL and page information: "http://info.cern.ch/hypertext/WWW/TheProject.html <[1/1] All".

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)
Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)
on the browser you are using

[Software Products](#)
A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11 Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#))

[Technical](#)
Details of protocols, formats, program internals etc

[Bibliography](#)
Paper documentation on W3 and references.

[People](#)
A list of some people involved in the project.

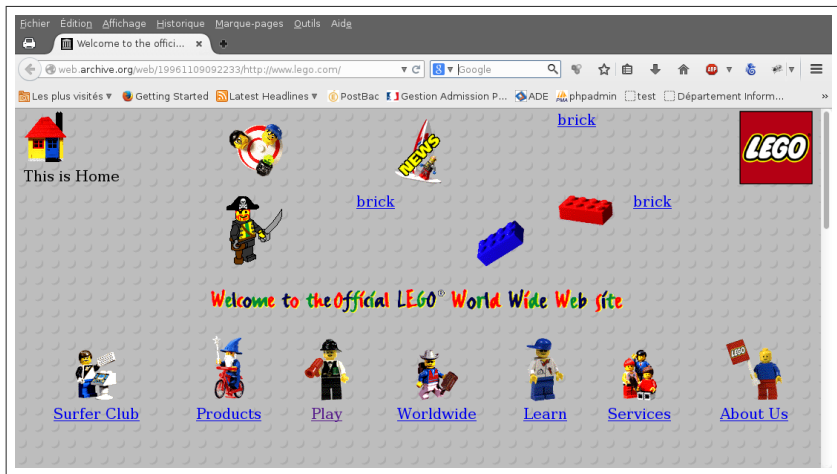
[History](#)
A summary of the history of the project.

[How can I help?](#)
If you would like to support the web..

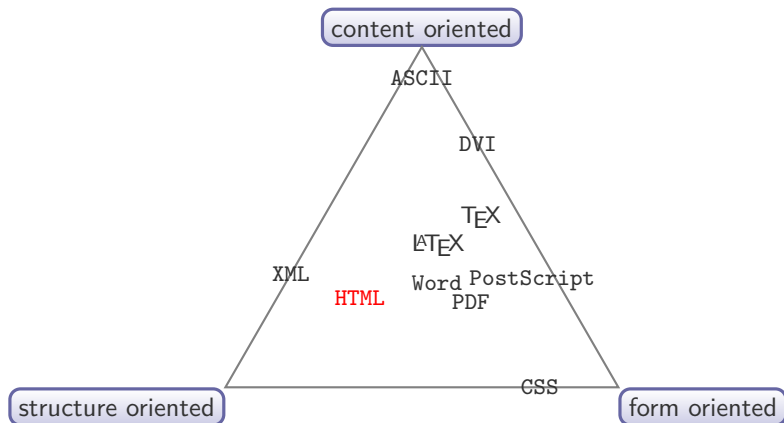
[Getting code](#)
Getting the code by [anonymous FTP](#), etc.

http://info.cern.ch/hypertext/WWW/TheProject.html <[1/1] All

Le site (pré)historique de Lego en 1996



HTML : langage de structuration de documents



Langage à balises

- HTML (Hyper Text Markup Language) est un langage à balises (tag).
- Décrit les pages WEB.
- Version 4.01 en 1999

<http://www.w3.org/TR/html401/> ↗

- Version 5

<http://www.w3.org/TR/html5/> ↗

- Intégration de la vidéo et du son, support pour le dessin.
- Ajout d'attributs, de contrôles et de balises structurantes.
- Adaptation aux périphériques.

Balise : mot clé entre "<" et ">" (case insensitive)

- Généralement, une balise de début (ouvrante) et de fin (fermante).
- Certaines balises sont seules : `
` (elles n'ont pas de contenus)

Attributs : données additionnelles d'une balise

- Une paire `nom="valeur"` (minuscule, souvent case sensitive)
- Placée dans la balise ouvrante.
- Il peut y en avoir plusieurs : `width`, `id`, `class`, `onclick`, ...

Un exemple

Balise et attributs

```
<a href="http://www.iut-fbleau.fr"> iut </a>
```

Balise et attributs

```
<a href="http://www.iut-fbleau.fr"> iut </a>
```

- Balise ouvrante <a>

Balise et attributs

```
<a href="http://www.iut-fbleau.fr"> iut </a>
```

- Balise ouvrante <a>
- Attribut href de la balise

Balise et attributs

```
<a href="http://www.iut-fbleau.fr"> iut </a>
```

- Balise ouvrante <a>
- Attribut href de la balise
- Contenu texte de la balise

Balise et attributs

```
<a href="http://www.iut-fbleau.fr"> iut </a>
```

- Balise ouvrante `<a>`
- Attribut `href` de la balise
- Contenu texte de la balise
- Balise fermante ``

Balise et attributs

```
<a href="http://www.iut-fbleau.fr"> iut </a>
```

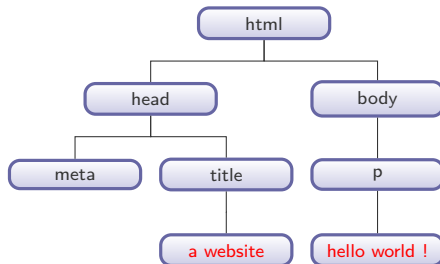
- Balise ouvrante <a>
- Attribut href de la balise
- Contenu texte de la balise
- Balise fermante

Certaines balises sont autofermantes (elles n'ont pas de contenu) :
, <embed>, <hr>, , <input>, <link>, <meta>, <param>, <source>, <wbr>

Imbrication des balises

L'écriture d'un document html syntaxiquement correct obéit à des règles précises :

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>un site</title>
  </head>
  <body>
    <p>hello world !</p>
  </body>
</html>
```



Représentation arborescente

Structure minimale d'un document html

Structure du document html

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <title>un site</title>  
  </head>  
  <body>  
    <p>hello world !</p>  
  </body>  
</html>
```

Structure minimale d'un document html

Structure du document html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>un site</title>
  </head>
  <body>
    <p>hello world !</p>
  </body>
</html>
```

Un préambule **DOCTYPE** qui indique la syntaxe utilisée dans le document.

Structure minimale d'un document html

Structure du document html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>un site</title>
  </head>
  <body>
    <p>hello world !</p>
  </body>
</html>
```

La balise **html** est la racine du document. Elle contient :

- la balise **head**, qui contient des métadonnées.
- la balise **body**, qui contient le contenu.

Structure minimale d'un document html

Structure du document html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>un site</title>
  </head>
  <body>
    <p>hello world !</p>
  </body>
</html>
```

La balise **head**, qui contient ici :

- la balise **meta** qui permet de rajouter des mots clés, le type du contenu, une description. Utilisée par les navigateurs et robots (référencement).
- un titre avec la balise **title**

On peut rajouter :

- Des ressources utilisées par la page avec la balise **link**.
- Des références aux fichiers javascript avec la balise **script**.


Structure minimale d'un document html

Structure du document html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>un site</title>
  </head>
  <body>
    <p>hello world !</p>
  </body>
</html>
```

La balise **body** contient les balises affichées dans le navigateur.

Validation de la syntaxe

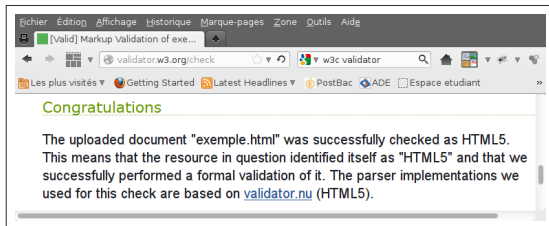
<http://validator.w3.org> 

Correction de la syntaxe

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>un site</title>
  </head>
  <body>
    <p>hello world !</p>
  </body>
</html>
```

Validation

<http://validator.w3.org>

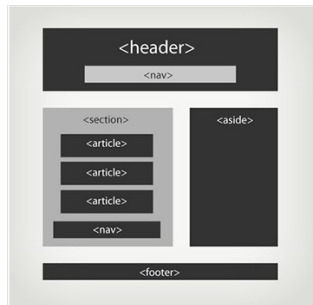


The screenshot shows a web browser window with the URL `validator.w3.org/check`. The page content displays a green checkmark icon and the text "Valid" in a green box, indicating a successful validation. Below this, the heading "Congratulations" is shown in green. The main text reads: "The uploaded document 'exemple.html' was successfully checked as HTML5. This means that the resource in question identified itself as 'HTML5' and that we successfully performed a formal validation of it. The parser implementations we used for this check are based on [validator.nu](#) (HTML5)." The browser's address bar shows the URL, and the page title is "[Valid] Markup Validation of exe...". The browser's menu bar includes "Fichier", "Édition", "Affichage", "Historique", "Marque-pages", "Zone", "Outils", and "Aide". The browser's toolbar shows various navigation and utility icons, including "Les plus visités", "Getting Started", "Latest Headlines", "PostBac", "ADE", and "Espace etudiant".

Éléments de sections

Les éléments de sections, introduits avec html5 (`section`, `article`, `nav`, `aside`, `header`, `footer`) définissent des portions du document avec une sémantique particulière.

- section** Section générique regroupant un même sujet, généralement avec un titre
- article** Section de contenu indépendante
- nav** Section possédant des liens de navigation principaux (au sein du document ou vers d'autres pages)
- aside** Section dont le contenu est un complément par rapport à ce qui l'entoure, qui n'est pas forcément en lien direct avec le contenu mais qui peut apporter des informations supplémentaires.
- header** Section d'introduction d'un article, d'une autre section ou du document entier (en-tête de page).
- footer** Section de conclusion d'une section ou d'un article, voire du document entier (pied de page).



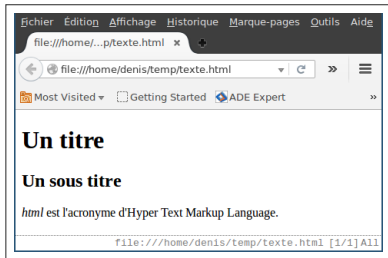
Commentaires

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
  <!--
  Ceci est un commentaire
  sur plusieurs lignes !
  -->
</body>
</html>
```

Il s'agit de balises qui ont un contenu texte avec une sémantique précise :

- `<p>paragraphe</p>`
- `<h1> ... <h6>` : titre
- `emphase`
- `renforcement`
- `<pre>affiche le contenu tel quel</pre>`
- `<blockquote>mise en valeur d'un paragraphe</blockquote>`
- `<acronym></acronym>` et `<abbr></abbr>`.
- etc.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <h1>Un titre</h1>
    <h2>Un sous titre</h2>
    <p>
      <em>html</em> est l'acronyme
      d'Hyper Text Markup Language.
    </p>
  </body>
</html>
```



Textes - mise en forme

- `texte`
- `
` passage à la ligne
- `<i>italique</i>`
- `gras`
- `<u>souligné</u>`
- `<strike>texte barré</strike>`
- `<u>souligné</u>`
- `<big>`, `<small>`
- `<sub>`, `<sup>`
- etc

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
  <p>Du <b>gras</b>, de <i>l'italique</i>,<br>
  du texte <del>barré</del>, du texte
  <u>souligné</u>, en <sub>bas</sub>, en <sup>haut</sup>,
  etc.
  </p>
</body>
</html>
```



Listes

- Liste ordonnée , liste à puces
- Élément de chaque liste

```
<ul>  
  <li>item 1</li>  
  <li>item 2</li>  
</ul>
```

- Liste de définition <dl> (définition list)
- rajouter un terme <dt> et sa définition <dd>

```
<dl>  
  <dt>html</dt>  
  <dd>langage utilisé pour affiché du contenu sur le web</dd>  
  <dt>balise</dt>  
  <dd>élément présentant certaines fonctionnalités</dd>  
  <dt>attribut</dt>  
  <dd>élément spécifique à une balise pouvant prendre certaines valeurs</dd>  
</dl>
```



Liste non ordonnée

- item1
- item2
- item3

Liste de définitions

html

langage utilisé pour affiché du contenu sur le web

balise

élément présentant certaines fonctionnalités

attribut

élément spécifique à une balise pouvant prendre certaines valeurs

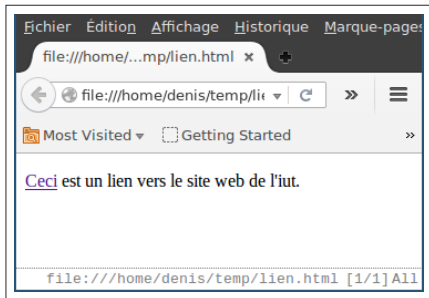
```
<a href="url ou #nom d'une ancre" target="ou">texte ou image</a>
```

- target représente où le lien doit être ouvert :
 - s'il manque, la même page
 - `_blank` un nouvel onglet ou nouvelle page
 - autres valeurs : `_self`, `_parent`, `_top`
- url est l'url du lien. il peut être :
 - absolu : `http://www.iut-fbleau.fr`
 - relatif par rapport
 - à l'adresse de base si document contient un élément base,
 - à l'adresse de la page en cours sinon.

Pour créer une ancre :

Ancre

```
<a name="ancre1">mon ancre</a>
```



```
<h1>Sports</h1>
```

```
<p>
```

```
  <a href="../sport.html#foot">Le football </a>
```

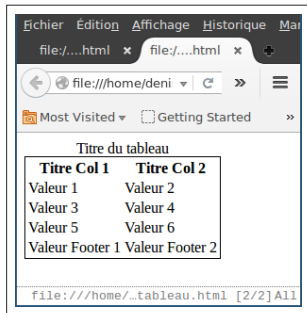
```
  <a href="#arc">Le tir à l'arc</a>
```

```
  <a href="http://www.ff-handball.org/">handball</a>
```

```
</p>
```

Tableaux

```
<table style="border:solid 1px black">
  <caption>Titre du tableau</caption>
  <thead>
    <tr>
      <th>Titre Col 1</th>
      <th>Titre Col 2</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td>Valeur Footer 1</td>
      <td>Valeur Footer 2</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Valeur 1</td>
      <td>Valeur 2</td>
    </tr>
  </tbody>
</table>
```



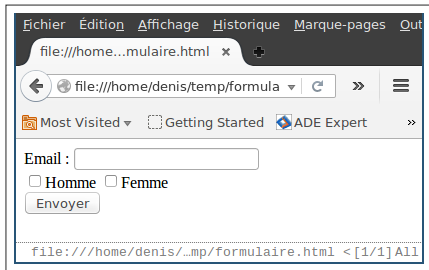
Formulaire

Un formulaire permet la saisie et l'envoi de données au serveur :

```
<form action="url" method="post ou get">
  <label for="mail">Mail</label>
  <input type="text" id="mail" required>
  <input type="submit" name="soumission" value="envoyer" />
</form>
```

- `url` : définit l'url du script à qui sont envoyées les données sur le serveur.
- `method` : les données sont envoyées soit en `get` (dans l'url), soit en `post` (dans le corps de la requête http).

Un formulaire contient un ou plusieurs éléments de type `input`, `select`, `textarea`, ... qui permettent à l'utilisateur de saisir des données.



- Il est conseillé d'ajouter à chaque champ un `label` (étiquette décrivant le champ)
- L'attribut `for` du `label` doit se référer à l'attribut `id` du champ correspondant
- L'attribut `placeholder` contient le texte affiché par défaut (et peut, à l'inverse, contenir tout type de caractères)
- La balise `<fieldset>` permet un rendu agréable par défaut

Nombreux types de champs possibles :

- Cases à cocher (checkbox), boutons radio (radio), zones de texte (textarea), sélections (select), barres de boutons (toolbar), fichiers à téléverser (file), etc.
- Variantes de text : tel , email , url , date , color , etc.

Attention, certains nouveaux types de champ (tel, date, color, etc.) ne s'affichent pas toujours bien sur tous les navigateurs.

On peut rajouter des contraintes de validations avec l'attribut `required` et `pattern`.

Multimedia

Image : balise

```

```

Video : balise <video>.

```
<video controls="controls" poster="exemples/exemplechatbg.jpg">  
  <source src="exemples/exemplechat.webm" type="video/webm" />  
  <source src="exemples/exemplechat.mp4" type="video/mp4" />  
  <source src="exemples/exemplechat.ogv" type="video/ogg" />  
  Votre navigateur ne supporte pas le tag &lt;video&gt;.  
</video>
```

Son : balise <audio>

Question : que représente < et >, et pourquoi l'utiliser ?

- `<div>` : ne représente rien de particulier. Cette balise (une division) permet néanmoins de regrouper du contenu pour lui donner un sens particulier. A utiliser avec l'attribut `class` par exemple.
- `` : comme la balise `div`, mais avec un affichage en mode ligne (et non bloc).

Structures

```
<!--...-->  
<!DOCTYPE>  
<html>  
<head>  
<title>  
<meta />  
<body>
```

Sections

```
<div>  
<span>  
<section>  
<header>  
<hgroup>  
<nav>  
<article>  
<details>  
<summary>  
<figure>
```

```
<figcaption>  
<aside>  
<footer>
```

Références

```
<base />  
<link />  
<style>  
<script>  
<noscript>
```

Cadres

```
<frameset>  
<frame />  
<noframes>  
<iframe>
```

Listes

```
<dir>  
<ol>  
<ul>  
<li>  
<dd>  
<dl>  
<dt>
```

Liens

```
<a>  
<map>  
<area>
```

Multimédia






```
<img />  
<video>  
<track>  
<audio>
```

```
<source>  
<embed>  
<applet>  
<object>  
<param />  
<canvas>  
<svg>
```

Tableaux

```
<table>  
<caption>  
<colgroup>  
<col />  
<thead>  
<tbody>  
<tfoot>  
<tr>  
<th>  
<td>
```

Formulaires

```
<form>  
<fieldset>  
<legend>  
<label>  
<button>  
<input />  
<textarea>  
<select>  
<optgroup>  
<option>  
<isindex>  
<menu>   
<command>   
<datalist>   
<output>   
<keygen> 
```

Rendus visuels

```
<center>  
<hr />  
<br />  
<wbr />   
<meter>   
<progress> 
```

Textes - mise en forme

```
<b>  
<basefont />  
<bdi>   
<bdo>  
<big>  
<font>  
<i>  
<mark>   
<strike>  
<sub>  
<tt>
```

```
<u>  
<s>  
<small>
```

Texte - sémantique

```
<abbr>  
<acronym>  
<address>  
<blockquote>  
<cite>  
<code>  
<del>  
<dfn>  
<em>  
<h1> ...<h6>  
<ins>  
<kbd>  
<p>  
<pre>  
<q>
```

```
<rp>   
<rt>  <ruby>   
<samp>  
<strong>  
<time>   
<var>
```