

Technologies Web

Enseignante : Samia SAIDI OUERGHI

Email : samiasaidi@gmail.com

- Selon vous, une Application Web c'est quoi ?



Proposition de définition

- C'est une application qui a pour particularité :
 - Accessible via un navigateur (Chrome, Safari, Firefox, etc.)
 - Elle s'adresse à une population hétérogène :
 - Des simples utilisateur(humains)
 - D'autres applications (Mobile / Desktop)
- Son contenu est hétérogène
 - Information structurée dans des SGBDs
 - Information non structurés (hypermédia)
- Fortement basée sur une structure

Exemples d'applications Web

- Exemples :
 - Orientée E-business, E-commerce
 - Catalogues online (Amazon, ...)
 - Auction, vente aux enchères (Ebay, ...)
 - Orientée service:
 - Online order tracking, Réservation en ligne (trip Advisor, Expedia,...)
 - Orientée communauté :
 - Portails web (yahoo, ...)
 - Social networks (Facebook, Flickr,Google+...)
 - Forum,...
 - Orientée contenu :
 - Digital librairies, ...
 - E-learning platforms (UVT)

Les technologies du Web

- Elles englobent les technologies de développement d'applications Web qui sont:
 - XHTML, CSS, Javascript, DOM, JQuery, Php, Perl, JSP, Ruby, ASP.NET, XML, Xquery, Ajax, JSON, Etc.
- Ainsi que les divers Frameworks et outils de gestion de contenu (CMS) tels que :
 - IBM WebSphere Portal,
 - ZendFramework
 - CodeIgniter,
 - phpNuke,
 - Symfony,
 - Ruby on Rail,
 - Etc.

Avant tout

- Avant de faire le tour de ces technologies,
 - À travers les cours,
 - Les Tps à rendre,
 - Les projets à élaborer,
- Quelques généralités sur la programmation en environnement WEB.

Web ≠ Internet

- Le **Web** est un système de fichiers présent sur des machines (serveurs) transitant par un protocole particulier **HTTP**, consultable grâce à des navigateurs web et fonctionnant **SUR** Internet !



- **Internet** est un assemblage de multiples réseaux, tous connectés entre eux en utilisant le protocole **TCP/IP**. Cet amas de câbles, de fibres optiques... de matériels, pour faire simple, constitue Internet, aussi appelé « **le réseau des réseaux** ».
- **Internet est hardware; le Web est software**
 - *Il y a plusieurs autres applications basées sur Internet:*

- *e.g. email telnet ftp usenet Instant Messenger etc*

- Tout périphérique est accessible sur un réseau par son adresse Internet Protocol (IP) par exemple : 74.125.19.147
- Lorsque vous demandez une page web à votre navigateur, vous tapez une adresse **Uniform Resource Locator (URL)**
http://www.google.fr
http://www.monsite.com/dossier/fichier.html
- Les serveurs Domain Name System (**DNS**) font le lien entre l'URL tapée et son adresse IP

Protocole
de
communication

Adresse
du
serveur

Arborescence
sur
le serveur

Internet : Historique

- 1967 : Arpanet (réseau militaire américain robuste aux pannes)
- Les années 70s: ARPANET doublait de taille chaque année.
- 1984: ~1000 d'ordinateurs militaires et académiques connectés.
- 1992:
 - ~1,000,000 d'ordinateurs connectés.
 - Internet society a été créée et le contrôle lui a été transféré :
 - Internet Engineering Task Force (IETF).
 - Internet Architecture Board .
 - Internet Assigned Number Authority .
 - **World-Wide-Web Consortium (W3C).**

Historique (bref) Web

- L'idée des liens hypertexte: a été proposée au début dans les années 40 par Vannevar Bush.
- En 1989: Tim Berners-Lee du European Particle Physics Laboratory (CERN) a conçu un système d'hypertexte pour connecter des document sur le Net.
- Il a conçu un langage pour spécifier le contenu des documents. Devenu par la suite : HyperText Markup Language (HTML).
- Il a conçu un protocole pour télécharger et interpréter le contenu des documents devenu par la suite Hypertext Transfer Protocol (HTTP).
- Il a implémenté le premier navigateur Mozaïc : seulement du texte, PAS d'objet multimédia.



Tim Berners-Lee

Pour accéder au Web

- Pour accéder au Web (client),
- Il vous suffit d'avoir : un **navigateur**



- Pour programmer du Web ?
- Développement Web (web application) vs. Développement application monoposte

Le serveur HTTP

- Pour programmer, il vous faut **un serveur HTTP**:
 - Par exemple :
 - **Microsoft IIS** (Internet Information Services)
 - Le serveur **Apache** (apache Foundation)
- Les environnements de développement:
 - (open-source) WAMP, LAMP, XAMPP permettent de simuler un environnement de développement Web.
 - (Propriétaire) Microsoft Web Platform : IIS+SQL SERVER + Visual studio +.NET FRamework.
 - etc.

Notion client/serveur

- Quand un visiteur veut aller sur un site Web. Il tape l'adresse URL, ok, mais ensuite?
- La page s'affiche, d'accord, mais entre-temps que s'est-il passé ?
 - Lorsque l'internaute tape l'adresse d'un site dans son browser (**client**) (www.google.fr par exemple), celui-ci envoie une requête au **serveur** qui héberge ce site. Le serveur transmet alors la page demandée au browser qui l'affiche.
 - Le **client**, c'est le navigateur Internet de l'internaute comme, Mozilla Firefox, Microsoft IE, Apple Safari, Google Chrome, Opera etc.

Static vs. Dynamic pages

- « A static web page is a web page that always comprises the same information in response to all download requests from all users. »
 - Le contenu (texte, multimedia, etc.) est toujours le même.
- ***Avantages :***
 - Rapide et facile à mettre en place même pour les non connaisseurs.
 - Un outil idéal pour donner un avant gout/maquette d'un site Web à construire.
- ***Inconvénients :***
 - N'offrir qu'une et une seule présentation, sans aucune possibilité de personnalisation et avec une interactivité limitée au strict minimum
 - La maintenance des sites Web de taille est couteuse.
 - Difficile de les garder à jour et surtout cohérent.

Static vs. Dynamic pages (cont.)

- Orientations du Web :
 - Services en ligne, E-business, les blog, les forums, etc.
 - Offrir un contenu personnalisé / dynamique qui s'adapte aux besoins de chaque visiteur en fonction de ses actions, + d'interactivité.
- Un page web dynamique est construite à la demande (à la volée) par le **serveur** (côté serveur), en fonction de critères spécifiques.
- La présentation et le contenu affichés peuvent ainsi être personnalisés de manière interactive, en fonction des produits, des internautes, des langues, etc.

Static vs. Dynamic pages (cont.)

- On reconnaît facilement une page dynamique grâce à l'[URL](#) qui s'affiche dans le navigateur web de l'utilisateur:
 - **Page statique**
`http://www.monsite.fr/accueil.html`: affiche la page ***accueil.html***, stockée telle quelle sur le serveur
 - **page dynamique**
`http://www.monsite.fr/accueil.php?langue=en`: affiche la page ***accueil.php*** en demandant au serveur d'afficher le contenu de cette page en anglais.
- *NB: Certaines pages ayant l'extension **.html** sont dites **dynamiques coté client** si elle comporte un code javascript(.js).*

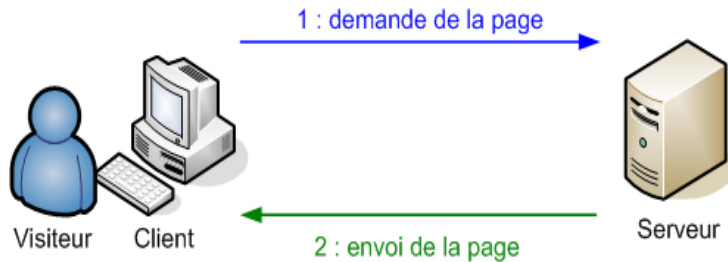
Static vs. Dynamic pages (cont.)

- Alors que les pages statiques font appel seulement à HTML, les pages dynamiques sont mises en œuvre grâce à des langages de programmation WEB dynamique.
 - But : disposer d'instructions conditionnelles, de boucles et de fonctions de traitement complexes. Le langage de programmation variera en fonction de la technologie retenue (PHP, ASP.NET, JSP, JSF, Perl, etc.).
- Le langage de programmation Web ne remplace pas le HTML, mais il en produit.
- Comment ?

Statique Vs Dynamique

Page Statique

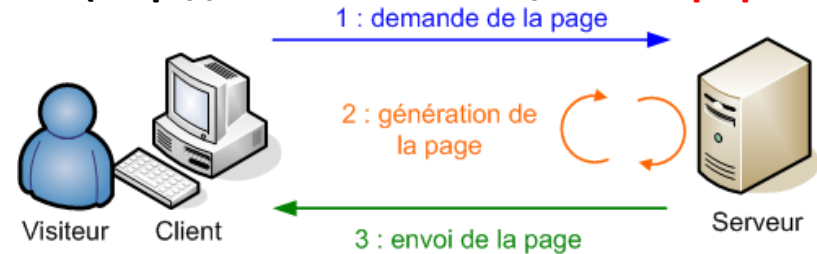
<http://www.monsite.fr/accueil.html>



- Demander au DNS l'adresse IP correspondant à www.monsite.fr
- Connecter à cette adresse IP
- Demander au serveur la page [accueil.html](http://www.monsite.fr/accueil.html)

Page Dynamique

(<http://www.monsite.fr/accueil.php>)



- Le client: "S'il te plaît, envoie-moi la page **accueil.php**".
- Le serveur n'envoie PAS de suite la page au client. Il la **génère** à partir du code php. En effet, le client n'est pas capable d'exécuter une page PHP (seul le serveur sait le faire).
- Une fois la page est générée, le serveur l'envoie au client.

Programmation de pages Web dynamiques : 2 types

- Plusieurs langages sont utilisés pour amener des pages personnalisées aux utilisateurs.
- *Type 1 : Programmation coté client (client-side scripting)*
 - Changer l'interface de la même page Web, en réponse à un événement bien particulier (entre dans le contexte de la programmation événementielle).
 - Le script (programme) est téléchargé avec la page Web et s'exécute sur la machine du client.
 - Exp.: **Javascript**, **VBScript** (seulement sur IE).

Programmation Web : 2 types

- *Type 2 : Programmation coté serveur
(server-side scripting)*

- Changer le contenu d'une page source et l'ajuster selon la réponse du serveur reçu par le navigateur. La réponse du serveur dépend:
 - Des données postées dans la requête : HTML forms, paramètres dans l'adresse URL, etc.
 - Du contenu de la BD interrogée via les scripts Php, asp, jsp etc.

L'évolution du Web

- Depuis son apparition le Web est passé par plusieurs versions:
 - **Web 0.0** : Web 0.0 – Developing the internet
 - **Web 1.0**: 1990~2000 – Web 1.0 – The shopping carts and static web
 - La version « read-only » du Web
 - Permet essentiellement la recherche et la lecture d'information
 - Très peu de communication entre les utilisateurs ou de contribution au contenu
 - But des utilisateurs: être présent sur le Net.
 - **Web 2.0** : le terme a été proposé en 2003 par Mr. Dale Dougherty, vice-president à O'Reilly Media

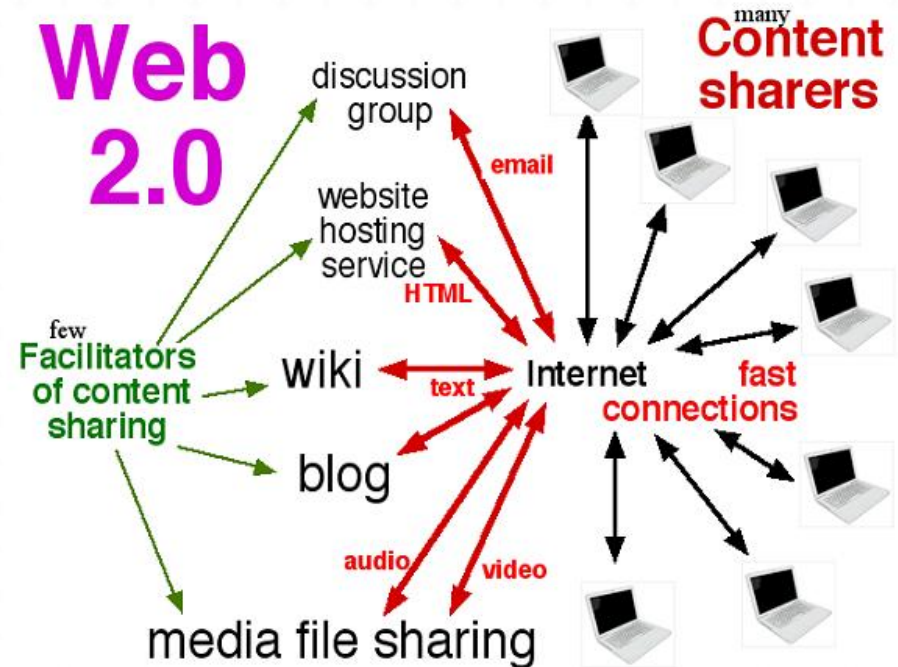
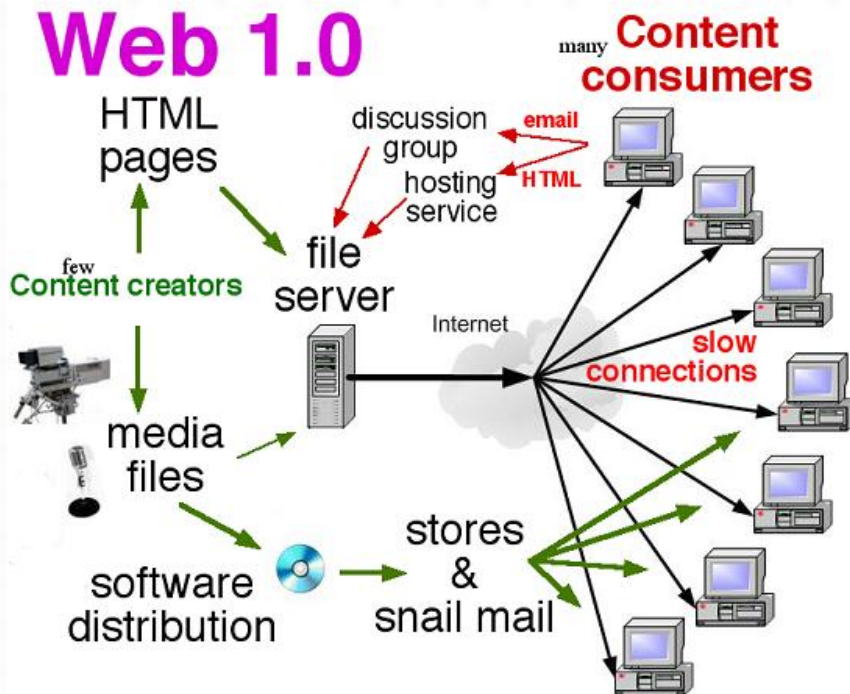


L'évolution du Web

- **Web 2.0**: 2000
- Web 2.0 – The writing and participating web
 - La version « read-write du Web »
 - Une interaction et une contribution accrue des utilisateurs au contenu du Web
 - Le shift avec l'apparition de Youtube, Myspace
 - Le grand boom avec l'apparition des social networks,



Web 1.0 vs Web 2.0



Des exemples de service du Web 2.0

- Blogs, social networks, podcasts, wikis, media sharing sites, etc.



L'évolution du Web

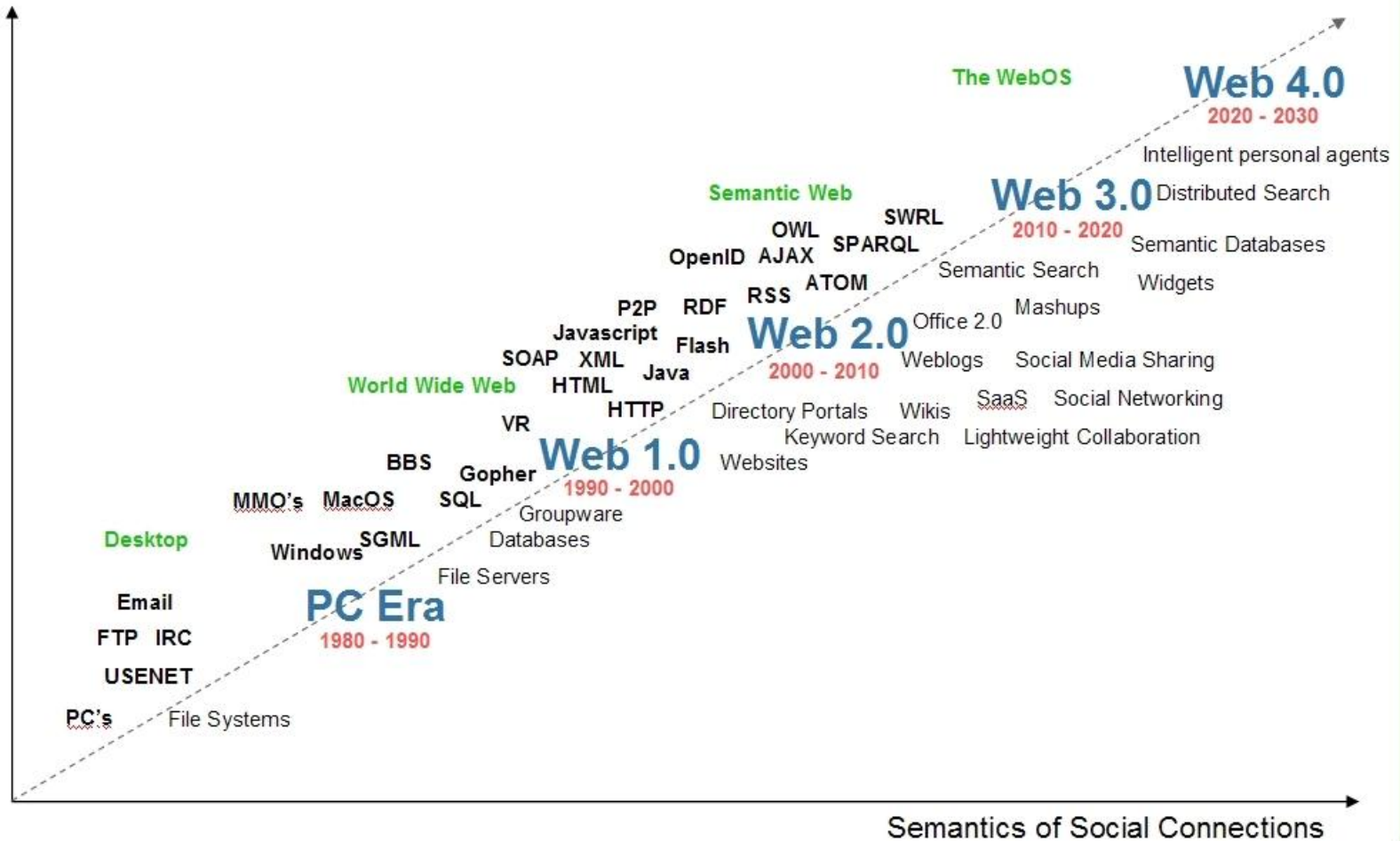
- **Web 3.0** : Web 3.0 –semantic markup and web services.
 - ***Web intelligent***
 - Convergence avec le domaine de l'IA
 - ***Web sémantique***
 - Les ordinateurs comprendraient l'information qui circulent sur le Web tout autant que les humains
 - ***Web services***
 - Read-write-execute" web, machine to machine communication



L'évolution du Web

- ***Web 4.0 – “Mobile Web”***
 - Ce n'est pas une nouvelle version mais plutôt une adaptation
 - “The next step is not really a new version, but is a alternate version of what we already have. Web needed to adapt to it's mobile surroundings.
 - Web 4.0 connects all devices in the real and virtual world in real-time.”

Semantics of Information Connections



“The next web”

- **Web 5.0** ou encore WebOS : Apparition d'un OS dédié Web
- **Open, Linked and Intelligent Web = Emotional Web**
- (emotional) interaction between humans and computers.
- The will perceive the users feel and emotions.
- Face recognition etc.

Les grands axes du cours

- HTML4.01 (Rappel ??), HTML5.1
- CSS1, CSS2, CSS3 (avant gout)
- Client-side scripting:
 - Javascript (bibliothèques), DOM, les APIs de HTML5
- Server-side scripting
 - Php7/Accès BD
- La Technologie AJAX, XML
- SimpleXML
- Nouvelles orientations:
 - programmable web Les Mashup

Références

- W3C : <http://www.w3schools.com/>
- **Internet & World Wide Web: How to Program (5th Edition)**, Harvey M. Deitel and Paul J. Deitel, 2012,
- **HTML5: Up and Running, Dive into the Future of Web Development**, Mark Pilgrim, O'Reilly Media, 222pp, 2010.
- **AJAX, Rich Internet Applications, and Web Development for Programmers**, Paul J. Deitel, Harvey M. Deitel, Prentice Hall, 1040pp, 2008.
- **PHP6 and MySQL Bible**, Steve Suehring, Tim Converse, Joyce Park, Wiley, Jan 20, 2009

(Versions électroniques disponibles)

Outils de travail

- éditeurs:
 - Brackets (<http://brackets.io/>)
 - Atom (<https://atom.io/>)
 - Notepad++ (<http://notepad-plus-plus.org/>)
 - Sublime Text 2 <http://www.sublimetext.com/2>
 - Eclipse Web Tools Platform (WTP) Project Development
 - Les lives IDE (Google app Engine, JSFiddle, Thimble, JS Bin, CodePen, Etc.)
- Validator
 - HTML validator : <http://validator.w3.org/>
 - CSS validator : <http://csslint.net/>
 - JS Checker: <http://jshint.com/>

- HTML4.01



Le HTML : Définition de l'acronyme

➔ **HTML : HyperText Markup Language** : est un langage de balisage de **description** du **contenu** de pages Web.



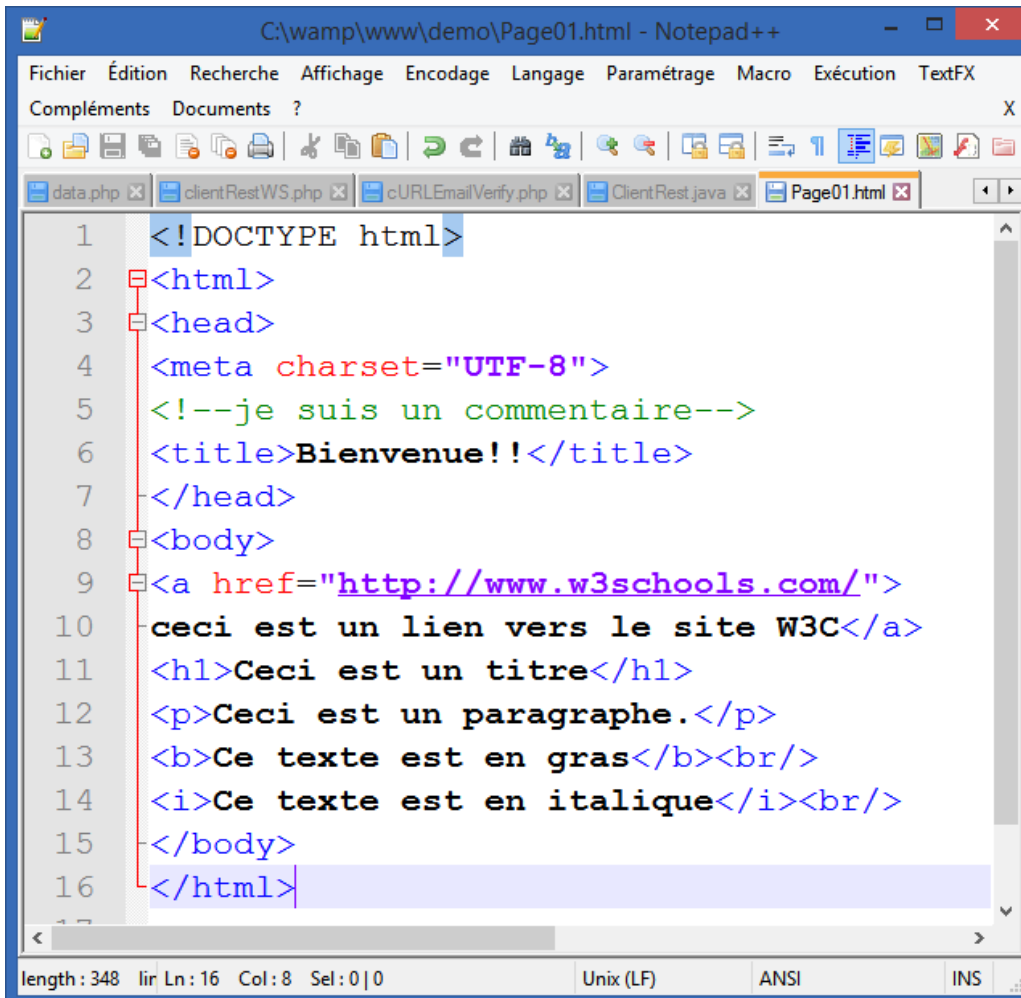
- **Hypertext** réfère au fait que les pages contiennent bien plus que du texte seulement:
 - Peut contenir des **images**, des **objets multimédia**, des **liens** référant à d'autres partie de la même page/d'autres pages.
- **Markup** réfère ici que, autre le texte à afficher, une page comporte des **balises** (**tags**) pour définir la **structure** et le **contenu** d'une page.

Les versions du HTML en quelques dates

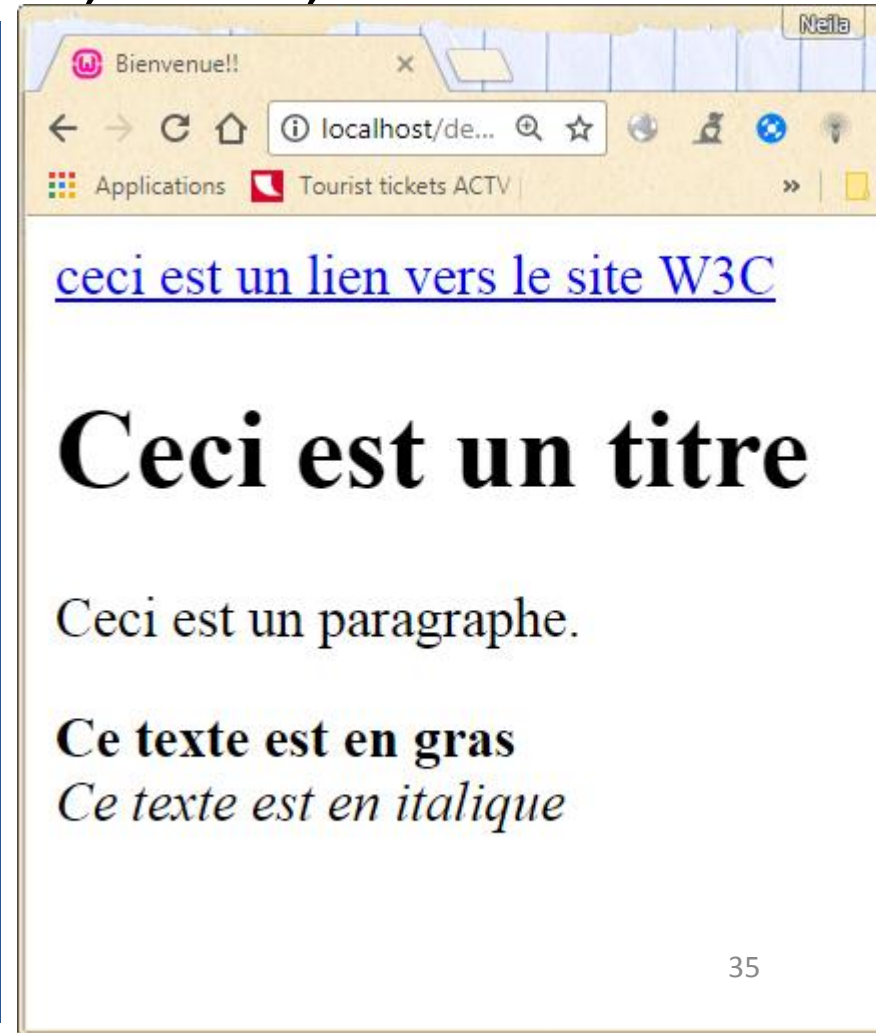
- **HTML 1.0 (Berners-Lee, 1989)**: très basique, intégration limitée d'objet multimédia.
- **HTML 2.0 (IETF, 1994)**: ont essayé de standardiser cette version, mais plus tard vers 1994-96, Netscape & IE ont ajouté +sieurs dispositifs divergents.
- **HTML 3.2 (W3C, 1996)**: ont essayé d'unifier le tout en un unique standard, technologies comme Java applets & streaming video n'ont pas été prise en considération.
- **HTML 4.0 (W3C, 1997)**: L'utilisation de frames, des tableaux plus complexes, des améliorations sur les formulaires etc...Mais surtout, cette version permet pour la première fois l'utilisation de feuilles de style CSS.
- **XHTML1.0 (W3C, 2000)**: **HTML 4.01** modifié pour être conforme au standard XML.
- **XHTML 2.0 (W3C, 2006)**: Sortie d'une version de travail de la recommandation XHTML 2.0.
- **HTML5 (2009)**: Sortie d'une première version DRAFT.
- **HTML5 (2014)** : W3C Recommendation.
- **HTML 5.1 : W3C Candidate Recommendation: HTML 5.1**

Un premier exemple : page01.html

- Un document écrit en langage HTML est un fichier texte avec l'extension **.htm**, **.html**, **.shtm**, **.shtml**, **.xhtml**.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <!--je suis un commentaire-->
6   <title>Bienvenue!!</title>
7 </head>
8 <body>
9   <a href="http://www.w3schools.com/">
10  ceci est un lien vers le site W3C</a>
11  <h1>Ceci est un titre</h1>
12  <p>Ceci est un paragraphe.</p>
13  <b>Ce texte est en gras</b><br/>
14  <i>Ce texte est en italique</i><br/>
15 </body>
16 </html>
```

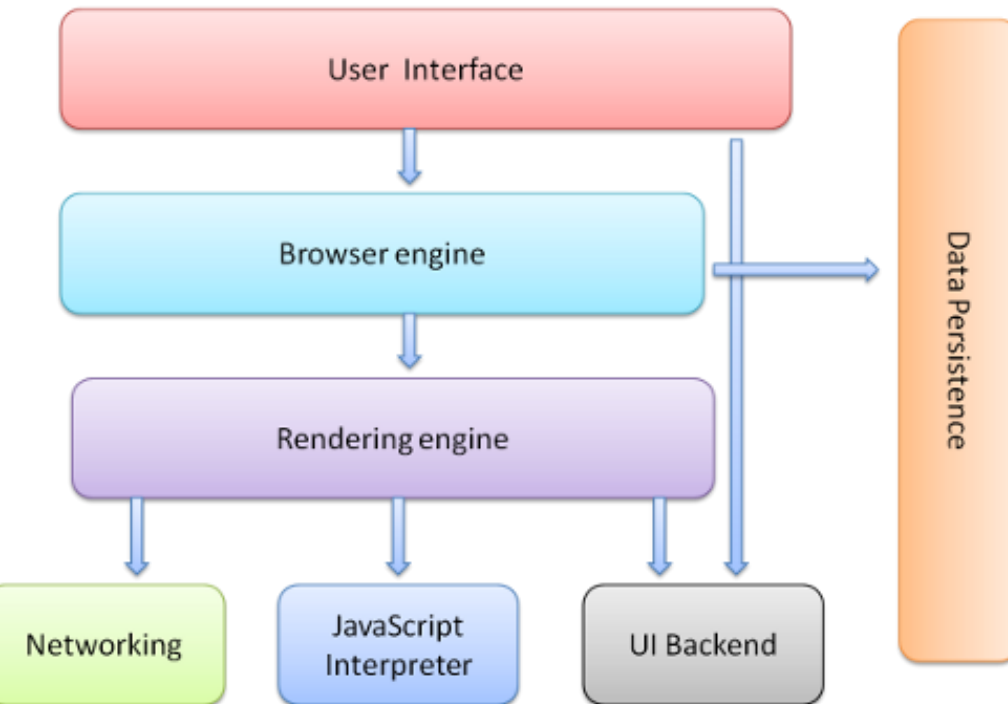


HTML est un langage interprété

- Les balises sont **invisibles** pour l'internaute :
 - Le navigateur va lire le document HTML et l'afficher sous forme d'une page Web.
 - Le navigateur **n'affiche pas les balises HTML** mais il utilise les balises pour **interpréter** le contenu de la page.
 - HTML **n'est pas** un langage de programmation mais de balisage qui est **interprété** ligne par ligne par le navigateur pour produire le contenu structuré selon les balises utilisées.

Structure de haut niveau d'un navigateur

- Le fonctionnement d'un navigateur:



- ➡ **JavaScript Interpreter** : utilisé pour parser et exécuter le code JS

- **User interface (Interface utilisateur)**: bar d'adresse, boutons suivant|précédent, le menu des favoris, etc.
- **Browser engine (Moteur du navigateur)** : l'interface d'envoi des requête au moteur de traitement (Rendering Engine)
- **Moteur de traitement (Rendering Engine)** : responsable de parser le contenu demandé de la page (HTML/CSS) et de l'afficher
- ➡ **Networking** : utilisé pour les traitements réseaux (requêtes HTTP par exemple)
- ➡ **Data Storage** : utilisé par le navigateur pour stocker toute sorte d'information persistante comme les cookies etc.
- ➡ **UI Backend** : utilisé pour dessiner à l'écran les fenêtres, les boutons de formulaire etc. (fonctionnement détaillé : voir

<http://taligarsiel.com/Projects/howbrowserswork1.htm>

De quoi est-il constitué page01.html?

- ➡ Son contenu est *structuré* à l'aide de *renères annelés*

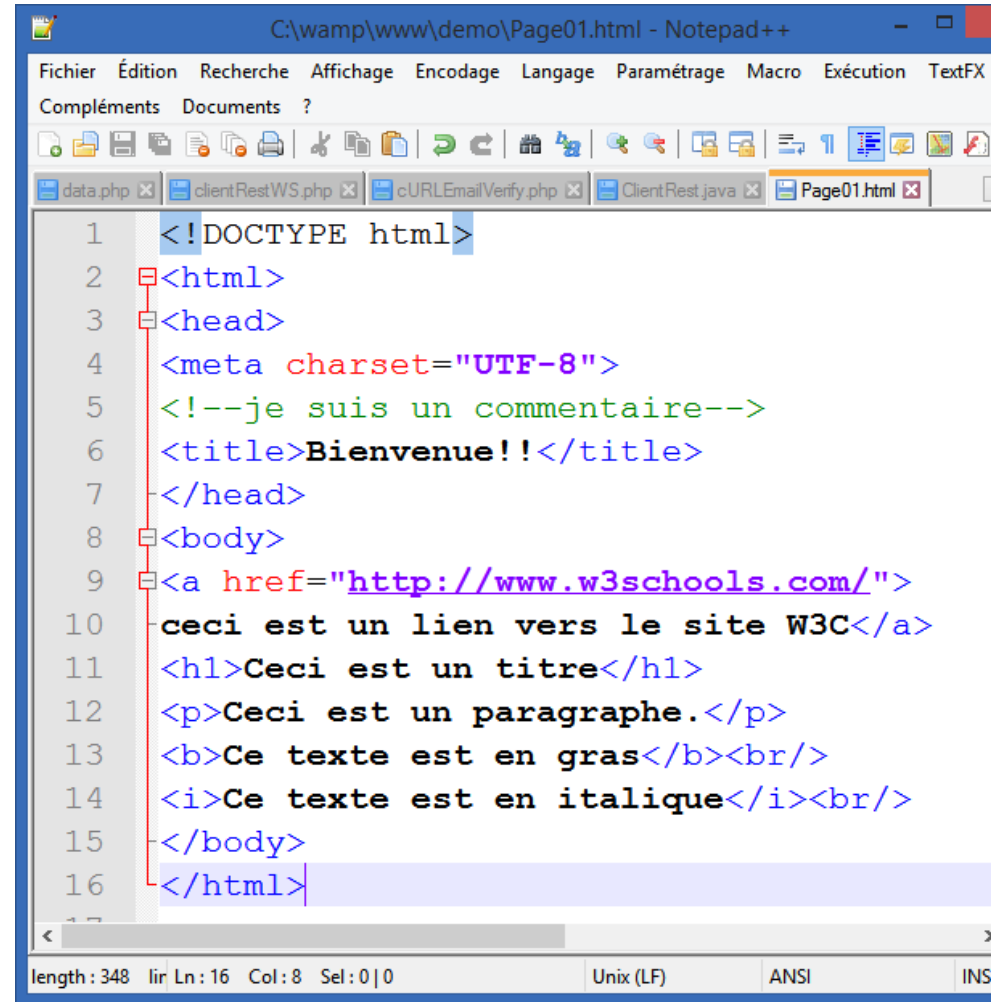
`<title>Bienvenue!!</title>`
des *éléments*

- Chaque élément est constitué de *balises* et d'*attributs* qui permettent d'apporter des informations sur son contenu

- **Balise** : mot-clé du langage HTML entouré par "<" et ">"

`<head>`, `<body>`, ...

- **Attribut** : un moyen de donner `href`, `title`, `src`, ... des précisions sur une balise.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <!--je suis un commentaire-->
6   <title>Bienvenue!!</title>
7 </head>
8 <body>
9   <a href="http://www.w3schools.com/">
10    ceci est un lien vers le site W3C</a>
11   <h1>Ceci est un titre</h1>
12   <p>Ceci est un paragraphe.</p>
13   <b>Ce texte est en gras</b><br/>
14   <i>Ce texte est en italique</i><br/>
15 </body>
16 </html>
```

Une balise ça sert à quoi?

- En parcourant un document HTML, les balises servent comme **marqueur** au navigateur pour indiquer :

```
<a href=" http://www.w3schools.com/ ">ceci est  
un lien vers le site W3C</a>  
<h1>ceci est un titre</h1>
```

```
<b>ce texte est en gras</b>  
<i>ce texte est en italique</i>
```

Les parties d'un document

HTML 1/2

- La balise **<html>**: C'est la balise principale qui englobera toute votre page HTML . On ne la ferme qu'en dernier avec **</html>** .
- Les deux parties fondamentales d'un document HTML sont l'en-tête (**<head>**) puis ensuite dans l'ordre le corps (**<body>**)
 - **l'en-tête (<head>)** : sert à enregistrer des informations complémentaires (mots-clefs, feuilles de styles applicables, des scripts à utiliser, des chargements annexes, etc.)
 - **Le corps du texte (<body>)** : Il contient tout ce qui **apparaîtra** dans la fenêtre du navigateur.

Les parties d'un document HTML 2/2

Le corps : <body>

- Paragraphes
- Titres
- Images
- Liens hypertextes
- Balises de mise en forme
- Listes
- Tableaux

- Formulaire
- Objets multimédia

L'en-tête : <head>

- Title
- Link
- Script
- style
- Meta

Syntaxe d'un élément HTML

- Un élément commence par une **balise d'ouverture** et se termine par une **balise de fermeture**

```
<balise> contenu </balise>
```

Syntaxe :

```
<p>ceci est un  
paragraphe.</p>
```

Exemple :

- Certains éléments HTML ont **un contenu vide**
- Les éléments à contenu vide ont une **balise de début**

```
<balise/>
```

```
<br/>
```

Syntaxe :

Exemple :

Attributs HTML

- Les éléments HTML peuvent avoir **des attributs** : fournissent **des informations supplémentaires** sur un élément
- Les attributs sont toujours spécifiés dans la balise de début et sont définis en paire "nom / valeur"
cc `<balise attribut1="val"> contenu </balise>`
er `<balise attribut='valeur' />`

Syntaxe :

```
<a href="http://site.fr/page.html"> ceci est  
un lien</a>
```

Vue Arborescente & Représentation imbriquée

- Les éléments d'un documents HTML sont imbriqués les uns dans les autres pour définir la

```
<racine>  
<premier_niveau>  
  <second_niveau> <niveau_3 /></second_niveau>  
</premier_niveau>  
<autre_balise_au_niveau_1 />  
</racine>
```

Par exem

```
<html>  
  <body>  
    <p>ceci est un paragraphe.</p>  
  </body>  
</html>
```

```

<!DOCTYPE html>
<html>
<head><title>Text
Layout</title></head>
<body>
  <p>
    This is a paragraph of text<br/>
    made up of two lines.
  </p>

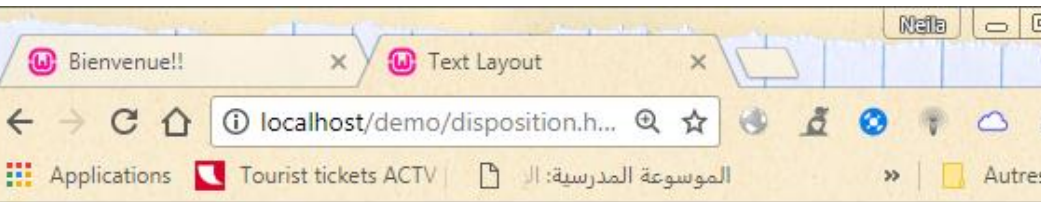
  <p>
    This is another paragraph with a
    &nbsp; GAP &nbsp; between
    some of the words.
  </p>

  <pre>
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both      spaces and
line breaks
  </pre>
  <p>

```

du texte

- Généralement, la disposition du texte doit être laissée à la charge du navigateur:
 - Plusieurs espaces sont interprétés comme un seul espace. Les retours à la ligne sont faits automatiquement par le navigateur.
- HTML donne la possibilité de forcer la disposition du texte:
 - Un retour à la ligne: la balise **
**
 - Un nouveau paragraphe (laisser une ligne vide et retour à la ligne) en utilisant **<p>...</p>**
 - Ajouter un espace dans un emplacement supposé être sans espace: le symbole ** **
 - Utiliser la balise **<pre>** pour préserver les espaces et les retours à la ligne.



This is a paragraph of text
made up of two lines.

This is another paragraph with a GAP between
some of the words.

Text in a pre element
is displayed in a fixed-width
font, and it preserves
both spaces and
line breaks

This paragraph is
indented on the first line
but not on subsequent lines.

```
<!DOCTYPE html>
<html>
<head><title>Text
Layout</title></head>
<body>
  <p>
    This is a paragraph of text<br/>
    made up of two lines.
  </p>
```

```
<p>
  This is another paragraph with a
  &nbsp; GAP &nbsp; between
  some of the words.
</p>
```

```
<pre>
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both spaces and
line breaks
</pre>
<p>
  &nbsp;&nbsp; This paragraph
is<br/>
  indented on the first line<br/>
  but not on subsequent lines.</p>
```

L'élément paragraphe

- L'élément **<p>** définit un paragraphe dans le document HTML.
 - Il a une balise de début **<p>** et une balise de fin **</p>**.
 - **<p>** provoque un saut de ligne avant le paragraphe.
 - Un paragraphe ne peut en contenir un autre.

- Attribut principal de **<p>**

Attribut	Valeur(s)
align	left (par défaut) right center justify

```
<html><head></head>
<body>
  <p>This is a paragraph of
text<br/>made up of two
lines.</p>
  <p align="center">This is a
centered paragraph.</p>
</body>
```

Titre et sous-titre

```
<html>
<body>
<h1>Titre super important</h1>

<h2>Titre important</h2>

<h3>Titre un peu moins
important</h3>

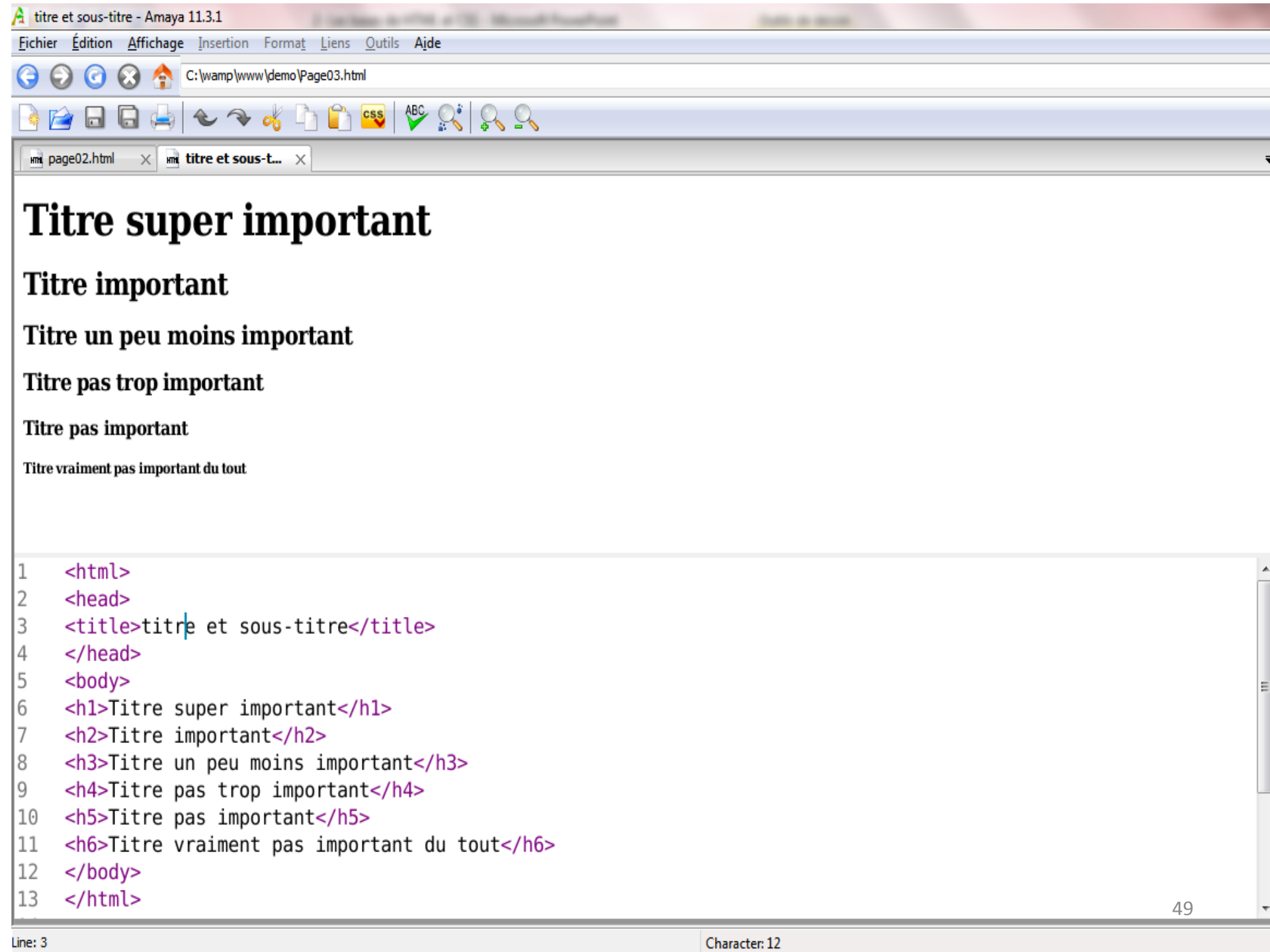
<h4>Titre pas trop
important</h4>

<h5>Titre pas important</h5>

<h6>Titre vraiment pas important
du tout</h6>

</body>
</html>
```

- Il s'agit ici des titres qui s'afficheront dans la page : ce sont les en-têtes de sections ou paragraphes.
- Il existe 6 niveaux de titres d'importance décroissante :
 - **<h1> </h1>** : signifie "titre très important". En général, on s'en sert pour afficher le titre de la page en haut.
 - **<h2> </h2>** : signifie "titre important".
 - ...
 - **<h6> </h6>** : titre vraiment pas important du tout.



Titre super important

Titre important

Titre un peu moins important

Titre pas trop important

Titre pas important

Titre vraiment pas important du tout

```
1  <html>
2  <head>
3  <title>titre et sous-titre</title>
4  </head>
5  <body>
6  <h1>Titre super important</h1>
7  <h2>Titre important</h2>
8  <h3>Titre un peu moins important</h3>
9  <h4>Titre pas trop important</h4>
10 <h5>Titre pas important</h5>
11 <h6>Titre vraiment pas important du tout</h6>
12 </body>
13 </html>
```

Lien hypertexte 1/4

- Un hyperlien (ou lien) est un mot, un groupe de mots, ou une image que vous pouvez cliquer dessus pour passer à une autre page Web ou pour passer à une nouvelle section de la même page.
- Un lien est défini en utilisant la balise `<a>` qui peut être utilisé pour créer :
 - **Lien externe**: vers un autre document, en utilisant l'attribut `href`
 - **Lien interne/signet/ancree** dans un même document, en utilisant l'attribut `id`.

Lien hypertexte 2/4

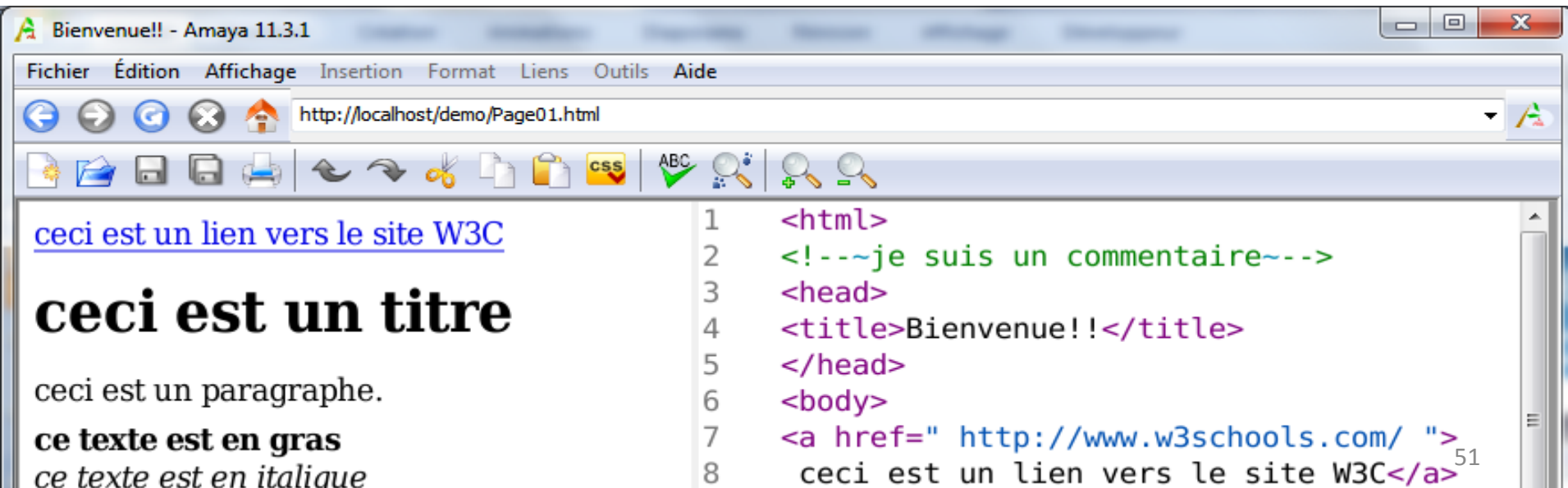
Syntaxe :

```
<a href="url" target="_blank">lien externe</a>
```

- L'attribut **href** spécifie la destination d'un lien.
- L'attribut **target** spécifie où ouvrir le document destination

Exemple :

```
<a href="http://www.w3schools.com/"> ceci est  
un lien vers le site W3C</a>
```



Lien hypertexte 3/4

- Un **lien interne** pointe vers une ancre, c'est à dire un endroit à l'intérieur d'un document défini par un nom. Il permet de naviguer dans les longs documents.
`ancre ici`

Syntaxe :

- L'attribut `href` spécifie le nom d'une ancre
`lien vers ancre`
- Le lien proprement dit se construit comme suit :

Exemple :

- On peut créer un lien vers la partie 1 d'une autre page
`Première Partie
...<h1>Partie 1</h1>`

```
<a href="http://www.site.rnu.tn/liens.html#partie1">  
Première Partie</a>
```



Run »

Result Size: 487 x 49

```
<!DOCTYPE html>
<html>
<body>

<p><a href="#C4">Jump to Chapter 4</a></p>

<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2 id="C4">Chapter 4</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>
```

[Jump to Chapter 4](#)

Chapter 1

This chapter explains ba bla bla

Chapter 2

This chapter explains ba bla bla

Chapter 3

This chapter explains ba bla bla

Chapter 4

This chapter explains ba bla bla



Run »

Result Size: 487 x

```
<!DOCTYPE html>
<html>
<body>

<p><a href="#C4">Jump to Chapter 4</a></p>

<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2 id="C4">Chapter 4</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>
```

Chapter 4

This chapter explains ba bla bla

Chapter 5

This chapter explains ba bla bla

Chapter 6

This chapter explains ba bla bla

Chapter 7

This chapter explains ba bla bla

Chapter 8

This chapter explains ba bla bla

Lien hypertexte 4/4

- Exemple : `contact`
d'autres protocoles:

Lien e-mail : `upload
link`

Lien externe vers un serveur FTP :

Lien vers d'autres objets :

Le système d'exploitation recherchera de lui-même, selon

l'exemple : `Téléchargement du
site`
téléchargement...

Images 1/3

- La balise `` est le moyen le plus simple d'insérer une image dans un document HTML.

```

```

Syntaxe :

- La balise `` est vide, c.à.d. elle contient uniquement des attributs et pas de balise de fermeture.
- **Src** : l'URL de l'image que vous souhaitez afficher.
- **Alt** : est un texte alternatif à afficher si l'image ne se charge pas.
- **Title** : affiche un texte dans une info-bulle.

```


```



```
C:\wamp\www\demo\Page02.html - Notepad++
Fichier Edition Recherche Affichage Encodage Langage Paramétrage Macro Exécution TextFX Compléments
Documents ? X
time.html Page02.html liste.html mapage.php doc.xml disposition.html

2 <html>
3 <head>
4   <title>Images</title>
5 </head>
6 <body>
7   
8   <br/>
9
10  
12
13  <p>
14    
16    un paragraphe avec un image. L'attribut align de l'image a
17    pour valeur "left". L'image est sur la gauche de ce paragraphe.
18  </p>
19
20  <a href="http://www.w3schools.com">
21    
22  </a>
23  </body></html>
24
25

length : 529 lines : 25 Ln: 3 Col: 1 Sel: 0 UNIX ANSI INS
```

Images 2/3



Images 3/3

- Le navigateur affiche l'image où la balise `` se trouve.
- Si vous placez la balise `` entre 2 paragraphes, le navigateur affiche l'image entre le 1^{er} et le 2^{ème} paragraphe.
- Utilisez l'attribut `align` pour laisser une image flottante à gauche ou à droite d'un paragraphe.

```
<p>  
un paragraphe  
avec un image. L'attribut align de l'image a pour  
valeur "left". L'image est sur la gauche de ce  
paragraphe.</p>
```

```
<a href="http://www.w3schools.com">  
</a>
```

Les listes

La notion de liste permet de structurer un ensemble de données :

- ***Type 1 : Listes non ordonnées (Unordred List)***
 - Une liste non ordonnée commence par la balise ****.
 - Chaque élément de la liste commence avec la balise ****.
 - Il y aura autant de **** que d'élément dans la liste.
- ***Type 2 : Les listes numérotées (Ordered List)***
 - Une liste non ordonnée commence par la balise ****.
 - Chaque élément de la liste commence avec la balise ****.
- ***Type 3 : Les listes de définition (Definition List)***
 - La balise **<dl>** définit une liste de définitions.
 - La balise **<dt>** définit l'élément de la liste et
 - **<dd>** décrit l'élément de la liste



ordered list

1. Je me leve
2. Je mange et je bois
3. Je retourne me coucher

unordered list

- Fraises
- Framboises
- Cerises

list of definition

Hacker	a clever programmer
Nerd	technically bright but socially inept person

```
<html> <head> <title>Les listes</title>
</head><body>
<h1> ordered list</h1>
<ol>
<li>Je me leve</li>
<li>Je mange et je bois</li>
<li>Je retourne me coucher</li>
</ol>
<h1> unordered list</h1>
<ul>
<li>Fraises</li>
<li>Framboises</li>
<li>Cerises</li>
</ul>
<h1>list of definition</h1>
<dl>
<dt> Hacker </dt>
<dd> a clever programmer </dd>
<dt> Nerd </dt>
<dd> technically bright but socially inept
person </dd>
</dl>
</body>
</html>
```

ordered list

1. Je me leve
2. Je mange et je bois
3. Je retourne me coucher

unordered list

- Fraises
- Framboises
- Cerises

list of definition

Hacker
a clever programmer

Nerd
technically bright but socially
inept person

```
1 <html>
2 <head><title>Les listes</title></head>
3 <body>
4 <h1>ordered list</h1>
5 <ol>
6 <li>Je me leve</li>
7 <li>Je mange et je bois</li>
8 <li>Je retourne me coucher</li>
9 </ol>
10 <h1>unordered list</h1>
11 <ul>
12 <li>Fraises</li>
13 <li>Framboises</li>
14 <li>Cerises</li>
15 </ul>
16 <h1>list of definition</h1>
17 <dl>
18 <dt>Hacker</dt>
19 <dd>a clever programmer</dd>
20 <dt>Nerd</dt>
21 <dd>technically bright but socially inept person</dd>
22 </dl>
23 </body>
24 </html>
```

Les tableaux

- En HTML, un tableau est un conteneur définie par la balise **<table>** et constitué de lignes **<tr>** (pour *Table Row*) et de cellules **<td>** (pour *Table Data*)
- Un tableau est divisé en lignes (avec les **<tr>**), et chaque rangée est divisée en cellules de données (avec les **<td>**).
 - **<td>** peut contenir du texte, liens, images, listes, formulaires, d'autres tables, etc.
- Par default, le contenu des colonne est **justifié a gauche**, et **sans bordure** pour cela il faut définir un alignement et les bordures **explicitement**.

Les tableaux: exemple

```
<html>
<head><title>Tables</title>
</head>
<body>
<h2>A Simple Table</h2>
  <table>
    <tr>
      <td> Left Column </td>
      <td> Right Column </td>
    </tr>
    <tr>
      <td> Some data </td>
      <td> Some data </td>
    </tr>
  </table>
</body>
</html>
```

A Simple Table

Left Column	Right Column
Some data	Some data

Les tableaux

- **Bordure :**

- Si vous ne spécifiez pas un attribut de bordure (***border***), le tableau sera affiché sans bordure.

Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

```
1
2 <table border="1">
3   <tr>
4     <th>Header 1</th>
5     <th>Header 2</th>
6   </tr>
7   <tr>
8     <td>row 1, cell 1</td>
9     <td>row 1, cell 2</td>
10  </tr>
11  <tr>
12    <td>row 2, cell 1</td>
13    <td>row 2, cell 2</td>
14  </tr>
15 </table>
```

- **En-tête:**

- Définis avec la balise ***<th>***.
- Le texte dans un élément sera en gras et centré.

Cellule qui relie 2 colonnes :

Name	Telephone	
Bill Gates	555 77 854	555 77 855

Cellule qui relie 2 lignes :

First Name:	Bill Gates
Telephone:	555 77 854
	555 77 855

Tableaux : les attributs rowspan et colspan

Cellule qui relie 2 colonnes :

Name	Telephone
Bill Gates	555 77 854 555 77 855

Cellule qui relie 2 lignes :

First Name:	Bill Gates
Telephone:	555 77 854 555 77 855

```
1 <html><body>
2 <h4>Cellule qui relie 2 colonnes :</h4>
3 <table border="1">
4 <tr>
5   <th>Name</th>
6   <th colspan="2">Telephone</th>
7 </tr>
8 <tr>
9   <td>Bill Gates</td>
10  <td>555 77 854</td>
11  <td>555 77 855</td>
12 </tr>
13 </table>
14 <h4>Cellule qui relie 2 lignes : </h4>
15 <table border="1">
16 <tr>
17   <th>First Name:</th>
18   <td>Bill Gates</td>
19 </tr>
20 <tr>
21   <th rowspan="2">Telephone:</th>
22   <td>555 77 854</td>
23 </tr>
24 <tr>
25   <td>555 77 855</td>
26 </tr>
27 </table>
```

- *rowspan* : Relier des cellules dans une rangée sur plusieurs colonnes
- *colspan* : Relier des cellules dans une colonne sur plusieurs rangées

Mis en en forme

- HTML4.01 défini plusieurs balises pour

Balise	Description
<u></u>	texte en gras
<u><big></u>	taille supérieure à la taille courante
<u></u>	accentuation
<u><i></u>	texte en italique
<u><small></u>	taille inférieur à la taille courante
<u></u>	Plus d'accentuation
<u><sub></u>	texte en indice
<u><sup></u>	texte en exposant
<u><ins></u>	Définit le texte inséré
<u></u>	Définit le texte supprimé

Mis en en forme : exemple

Text Variations

We can use **simple** tags to *change* the appearance of **text** within Web pages. Even super^{script} and sub_{scripts} are *supported*.

```
1  <html>
2  <head>
3  <title>Text Variations and Escape Sequences</title>
4  </head>
5
6  <body>
7  <h1>Text Variations</h1>
8
9  <p>We can use <b>simple</b> tags to <i>change</i>
10 the appearance of<strong>text</strong> within <tt>Web pages</tt>. Even super<sup>script</sup>
11 and sub<sub>scripts</sub> are <em>supported</em>.</p>
```

Mise en forme : autres balises (html 4.01)

- Autres balise de mise en forme !
 - Computer Tags : <tt>,<pre>, <code>,
<samp> ...
 - Citation, définition, etc : <abbr>,
<blockquote>,...
 - Direction du texte,
 - Etc..

Les entités HTML

- Elles sont utilisées pour écrire les caractères spéciaux du langage, les caractères accentués etc.

caractère(s) entité

< >

< >

é è

é è

™ ©

™ ©

π δ Δ

π δ Δ

" &

" &

- Pour une liste complète :
http://www.w3schools.com/tags/ref_entities.asp

Pourquoi écrire du code HTML Valide et respectant les standards du Web ?

- Dans le développement Web, il est essentiel que votre code HTML soit correct qui respecte minutieusement la syntaxe du Language :
 - Pour garantir la bonne interprétation de votre code sur différent navigateurs
 - Pour augmenter la chance que votre code va être correctement affiché avec les versions futurs des navigateurs
 - Pour garantir que votre code peut être interchangeable et/ou imbriqué avec d'autre Langages
 - Exp. XML, MathML, JS, PhP, etc.

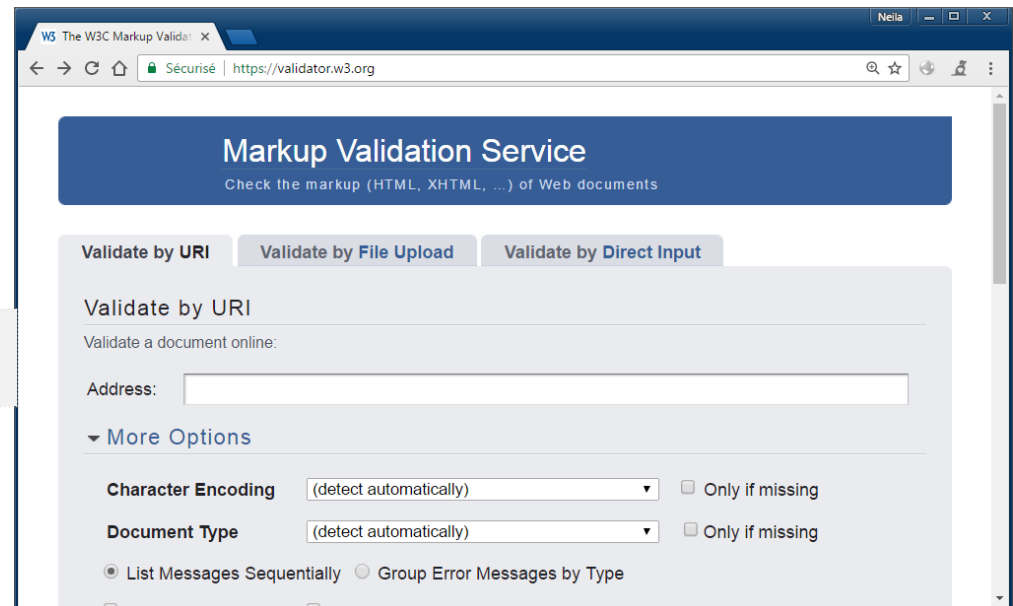
Importance du balisage

- Un document HTML est '**bien formé**' au sens du W3C si
 - Toute balise ouverte est fermée
 - L'ordre de fermeture respecte l'ordre inverse d'ouverture
 - Les balises sont écrites en minuscule. Les versions futures de HTML ne vont plus acceptées les balises en MAJ.
- Comment savoir si une page est bien formée ?

W3C Validator

- **HTML Validator** permet de corriger et de valider le code HTML d'une page Web selon les critères émis par le W3C.
 - Après analyse de la syntaxe d'un fichier HTML, il corrige et valide celui-ci. Il est capable de corriger plusieurs erreurs, mais lorsque la correction lui semble trop difficile à déterminer, il envoie un message d'avertissement.

<https://validator.w3.org/>




The screenshot shows the W3C Markup Validation Service interface in a web browser. The browser's address bar displays "https://validator.w3.org". The page has a blue header with the text "Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below the header, there are three tabs: "Validate by URI" (selected), "Validate by File Upload", and "Validate by Direct Input". Under the "Validate by URI" tab, there is a section titled "Validate by URI" with the instruction "Validate a document online:". Below this, there is a text input field labeled "Address:". Further down, there is a section titled "More Options" with a dropdown menu for "Character Encoding" set to "(detect automatically)", a checkbox for "Only if missing", a dropdown menu for "Document Type" set to "(detect automatically)", another checkbox for "Only if missing", and two radio buttons: "List Messages Sequentially" (selected) and "Group Error Messages by Type".

W3C online validator

← → ↺ 🏠 validator.w3.org/#validate_by_uri+with_options

📄 📄 ★ Web 2.0 Icons - Dow... 📖 ...: calibre library :... ... 📄 COMPT1 🌿 Flash Jigsaw 📄 0 Notifications 📺 MOSAIQUE FM :: LIVE 📺 Télécharger - Sams ... 📖 Ebooks Livres - Ergor » 📁 その他のブックマーク

**Markup Validation Service**
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI | Validate by File Upload | Validate by Direct Input

Validate by URI

Validate a document online:

Address:

▼ More Options

Character Encoding ☐ Only if missing

Document Type ☐ Only if missing

☒ List Messages Sequentially ☐ Group Error Messages by Type

☐ Show Source ☒ Clean up Markup with HTML-Tidy

☐ Show Outline ☐ Validate error pages ☐ Verbose Output

Les formulaires HTML (4.01)

Les Formulaires

- Dans des formulaires, l'utilisateur peut compléter des champs de saisie, faire des choix dans des listes et cliquer sur des boutons etc.
- Il faut associer un traitement selon le besoin :
 - Un traitement sur le client, avec Javascript par exemple.
 - Un traitement sur le serveur, avec PHP par exemple.
- Dans quel but ?
 - Récolter de l'utilisateur des renseignements.
 - Permettre la recherche dans des BDs.
 - Offrir à l'utilisateur la possibilité d'une interaction individuelle par exemple en commandant un produit déterminé dans un assortiment de produits.
 - Etc.

Syntaxe

- Balise placée dans la balise <body>:

Syntaxe: `<form method="" action="" name="" >...</form>`

- Les champs de type **input**, **select**, ou **textarea** sont placées à l'intérieur de l'élément **form**

Attributs:

- **method**: valeurs **GET** ou **POST** qui indique la façon dont les données du formulaire sont transmises au serveur.
- **action (obligatoire)**: l'url d'un script sur le web serveur appelé pour générer la page. Nom de la page qui sera exécuté quand l'utilisateur clique sur un bouton de soumission.
- **name**: le nom du formulaire.

Exemple :

```
<form action="form_action.php"
method="get" name="f1">...</form>
```

Les éléments d'un formulaire

- 3 catégories :
 - **Input** : Champs de saisie de texte + les boutons
 - type = "text" – zone de texte (type par défaut)
 - type = "password" – zone de texte caché
 - type = "radio" – minimum 2, un seul sélectionnable
 - type = "checkbox" – cases à cocher
 - type = "submit" – soumission de formulaire
 - type = "reset" – bouton de remise à zéro des champs
 - type = "button" – bouton associé à du code Javascript
 - type = "hidden" – bouton caché
 - **Select**: menus déroulants, listes à choix
 - Size="1"- liste simple, 1 seul élément sélectionnable
 - Size="1"- liste à choix multiple
 - **Textarea**: zone de saisie d'un texte "long"

Form Test

- Simple inputs :
 - Text :
 - Password :
 - Radio : Choice 1 ☐ Choice 2 ☐ Choice 3 ☐
 - Checkbox : Box 1 ☐ Box 2 ☐ Box 3 ☐
 - Hidden :
- Selection Fields :
 - Simple :
 - Disabled :
 - Preselected :
 - Multiple :
- Text area :
- File upload :
- Validation button :
 - Submit :
 - Reset :

Zone de saisie texte

Syntaxe :

```
<input type="text" size=".."
maxlength=".." name="nom"
```

Attributs :

```
value="valeur par défaut"/>
```

- **size**: taille d'affichage de la zone en caractère, 20 par défaut.
- **disabled="disabled"** (non utilisable et non cliquable)
- **readonly="readonly"** (non modifiable)
- **Maxlength**—le nombre maximal de caractère permis

Exemple :

```
<html>
<body>
<form><fieldset><legend>Personal
information:</legend>
<label for="Name">Name:</label>
<input type="text" size="30" /><br/>
<label for="e-mail">E-mail:</label>
<input type="text" size="30" />
</fieldset></form>
</body></html>
```


Zone de saisie texte : exemple

```
1  <html>
2  <body>
3  <form><fieldset>
4  <legend>Personal information:</legend>
5      <label for="Name">Name:</label>
6      <input type="text" size="30" maxlength="30" name="fname"/><br/>
7      <label for="e-mail">E-mail:</label>
8      <input type="text" size="30" readonly="readonly" value="e-mail"/><br/>
9      <label for="Date of birth">Date of birth:</label>
10     <input type="text" size="10" disabled="disabled" />
11 </fieldset>
12 </form>
13 </body>
14 </html>|
```

Personal information:

Name:

E-mail:

Date of birth:

Zone de saisie de mot de passe

Syntaxe

```
<input type="password" size="."
maxlength=".." name="nom" value="valeur
par défaut"/>
```

Attributs

- **size**: taille d'affichage de la zone en caractère, 20 par défaut.
- **disabled="disabled"** (non utilisable et non cliquable)
- **readonly="readonly"** (non modifiable)
- **Maxlength**=le nombre maximal de caractère permis.

Général

```
<html>
<body><form><fieldset>
<legend>Personal information:</legend>
  <label for="Password">Password:</label>
  <input type="password" size="8" name="fpass"/>
</fieldset>
</form>
</body></html>
```

Exemple :

Zone de saisie de mot de passe: exemple

```
1 <html>
2 <body>
3 <form action=""><fieldset>
4 <legend>Personal information:</legend>
5   <label for="Name">Name:</label>
6   <input type="text" size="30" maxlength="30" name="fname"/><br/>
7   <label for="Password">Password:</label>
8   <input type="password" maxlength="8" value="pass" name="fpass"/><br/>
9   </fieldset>
10 </form>
11 </body>
12 </html>
```

Personal information:

Name:

Password:

Définir une liste de choix

- Cases à cocher permettant un choix multiple :

Syntaxe

```
<input type="checkbox" name="nom" value="val"/>
```

Attributs

- Name, value, disabled="disabled" (non sélectionnable)
- **checked="checked"** (coché)

Exemple

```
<html><body><form action="">
<input type="checkbox" name="langage" value="js" />
<label for>Javascript</label><br />
<input type="checkbox" name="langage" value="php" checked="checked" />
<label for>php</label></form></body></html>
```

- NB: l'attribut **name** a la même valeur pour relier +sieurs choix!!!

Définir une liste de choix : exemple

```
1  <html>
2  <body>
3  <form action="">
4  <input type="checkbox" name="langage" value="js" />
5  <label for>Javascript<label><br />
6  <input type="checkbox" name="langage" value="php" checked="checked"/>
7  <label for>php<label><br />
8  <input type="checkbox" name="langage" value="java"/>
9  <label for>java<label><br />
10 </form>
11 </body>
12 </html>
```

Définir des cases d'option

- Choix d'une et une seule option parmi n (appelé aussi boutons radios)

Syntaxe :

```
<input type="radio" name="nom" value="val" />
```

Attributs :

- Name, value, disabled="disabled" (non sélectionnable)

Ex

```
<html><body><form action="">
<input type="radio" name="langage" value="js" />
<label for>Javascript</label><br />
<input type="radio" name="langage" value="php"
checked="checked" />
<label for>php</label>
</form></body></html>
```

Définir des cases d'option: exemple

```
1 <html>
2 <body>
3 <form action="">
4 <input type="radio" name="langage" value="js" />
5 <label for>Javascript</label><br />
6 <input type="radio" name="langage" value="php" checked="checked" />
7 <label for>php</label><br />
8 <input type="radio" name="langage" value="java" />
9 <label for>java</label><br />
10 <input type="radio" name="langage" value="perl" disabled="disabled" />
11 <label for>Perl</label><br />
12 </form>
13 </body>
14 </html>
```


Bouton de validation

- Envoi des données et exécution du programme PHP spécifié par l'attribut **action** de **<form>**

Syntaxe:

```
<input type="submit" name="nom"  
value="ok"/>
```

Attributs :

- Name, type, size, disabled="disabled" (non sélectionnable)
- **value**: permet de différencier le traitement à effectuer s'il y a plusieurs boutons.

```
<form>...  
<input type="submit" value="Envoyez" size="40"/>  
<input type="submit" value="Envoyez (désactivé)"  
disabled="disabled"/>  
</form>
```

```

3 <form name="input" action="html_form_action.php">
4 <input type="radio" name="langage" value="js" />
5 <label for>Javascript<label><br />
6 <input type="radio" name="langage" value="php" checked="checked"/>
7 <label for>php<label><br />
8 <input type="radio" name="langage" value="java"/>
9 <label for>java<label><br />
10 <input type="radio" name="langage" value="perl" disabled="disabled"/>
11 <label for>Perl<label><br />
12 <input type="submit" value="Envoyez" size="40"/>
13 <input type="submit" value="Envoyez (d&#xe9;sactiv&#xe9;)" disabled="disabled"/>
14 </form>
15 <p>Si vous cliquez le bouton "Envoyez", le contenu du formulaire va etre envoy&#xe9;
&#xe0; une page appel&#xe9;e "html_form_action.php". Remarquez que l'autre bouton est
d&#xe9;sactiv&#xe9;;</p>
16 </body>
17 </html>

```

Le bouton reset

- Recharge tous les champs du formulaire à leur valeur par défaut **ET** ne provoque **PAS** l'envoi du contenu à la page

PHP a `<input type="reset" name="nom" value="annuler"/>`

Syntaxe:

Attributs :

- **name, type, size, disabled="disabled"** (non sélectionnable)
- **value:** `<form>...`

plusie `<input type="reset" value="Annuler" size="40"/>`

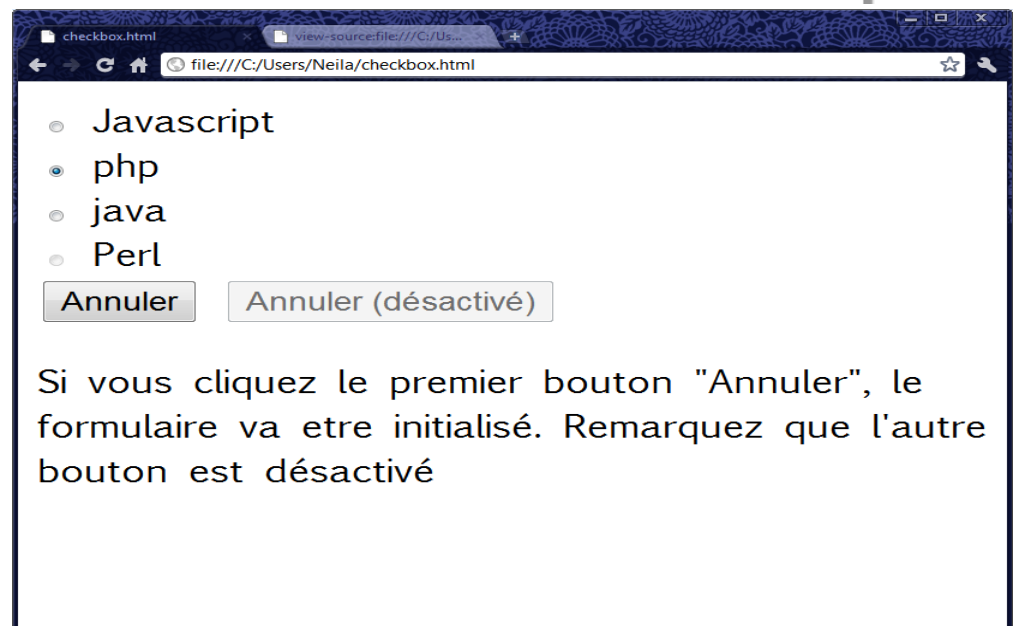
Exemple :

```
<input type="reset" value="Annuler
(désactivé)" disabled="disabled"/>
</form>
```

```

1 <html>
2 <body>
3 <form name="input" action="html_form_action.php">
4 <input type="radio" name="langage" value="js" />
5 <label for>Javascript</label><br />
6 <input type="radio" name="langage" value="php" checked="checked" />
7 <label for>php</label><br />
8 <input type="radio" name="langage" value="java" />
9 <label for>java</label><br />
10 <input type="radio" name="langage" value="perl" disabled="disabled" />
11 <label for>Perl</label><br />
12 <input type="reset" value="Annuler" size="40" />
13 <input type="reset" value="Annuler (désactivé)" disabled="disabled" />
14 </form>
15 <p>Si vous cliquez le premier bouton "Annuler", le formulaire va être initialisé.
16 Remarquez que l'autre bouton est désactivé;</p>
17 </body>
18 </html>

```



<input type = "button">

- Il N'a de sens que dans un contexte JavaScript
 - Pas de comportement préprogrammé
 - Ne | `<input type="button" name="nom"`

Syntaxe: `value="bouton"/>`

Attributs :

- Name, value, size, disabled="disabled"

```
<form>...
```

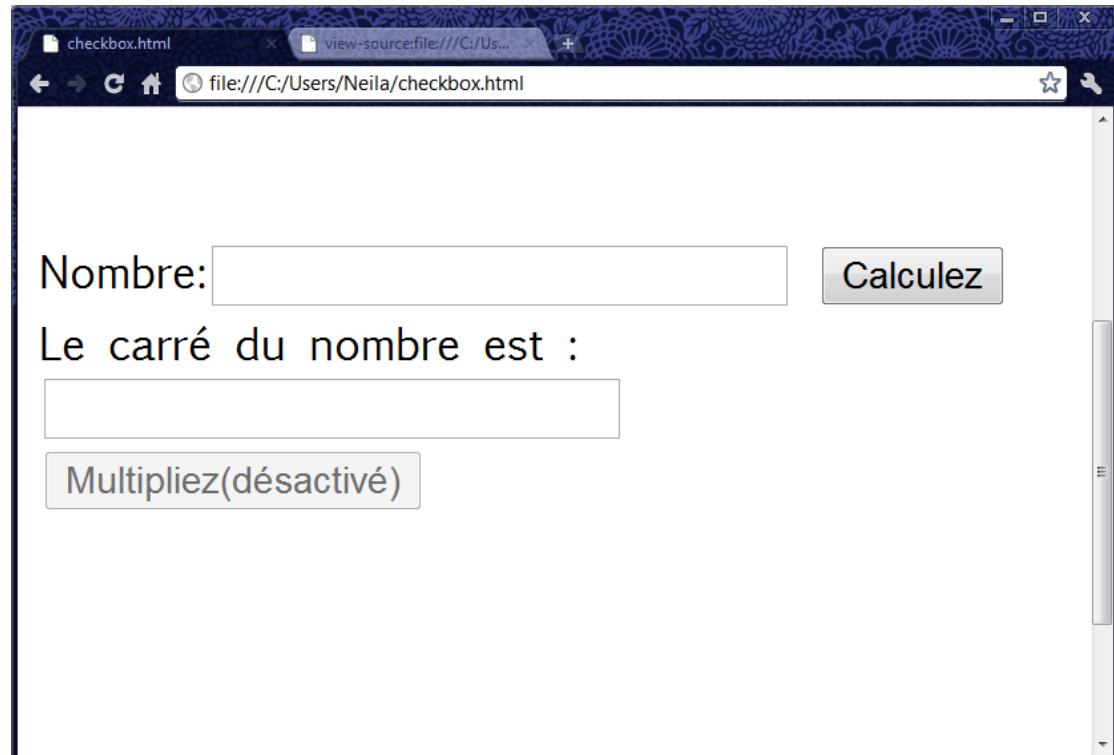
```
<input type="button" value="Calculez" name="somme"  
size="40"/>
```

```
<input type="button" value="Multipliez (désactivé)"  
name="produit" disabled="disabled"/>
```

```
</form>
```

<input type = "button"> : example

```
1 <html>
2 <body>
3 <form name="input" action="html_form_action.php">
4 <label for="nombre">Nombre:</label><input type="text" size="30" />
5 <input type="button" value="Calculez" name="somme" size="40"/><br/>
6 <label for="carre">Le carré du nombre est :</label>
7 <input type="text" size="30" /><br/>
8 <input type="button" value="Multipliez(désactivé)" name="produit"
9 disabled="disabled"/>
</form>
```



Zone de texte libre

- Zone de saisie de texte libre qui permet de saisir du texte sur plusieurs lignes et colonnes

Syntaxe :

```
<textarea name="nom" rows=".." cols="..">
```

Contenu qui sera affiché par défaut

```
</textarea>
```

Attributes :

- name, readonly, disabled="disabled"
- rows,cols

Exemple :

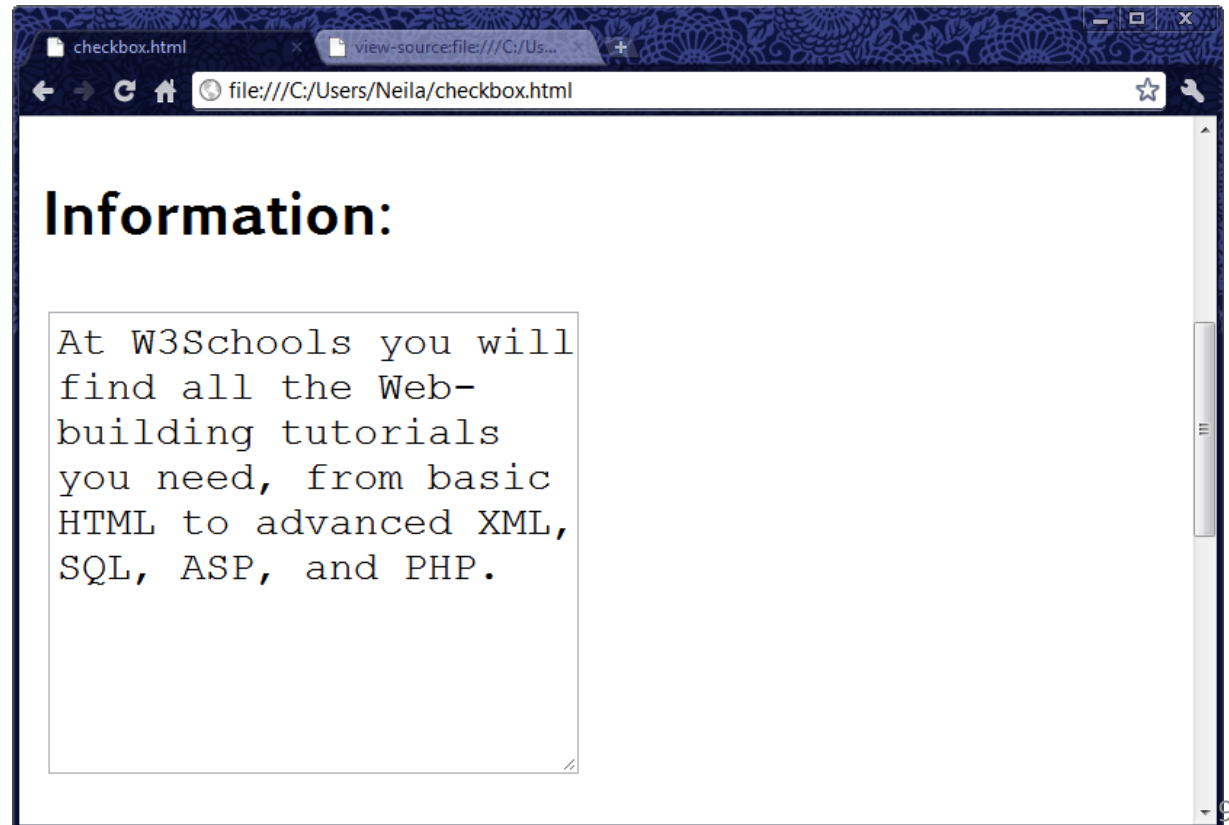
```
<textarea rows="10" cols="20" name="ftexte" >
```

At W3Schools you will find all the Web-building tutorials you need, from basic HTML to advanced XML, SQL, ASP, and PHP.

```
</textarea>
```

Zone de texte libre : exemple

```
1 <html>
2 <body>
3 <form name="input" action="html_form_action.php">
4 <h3> Information: </h3>
5 <textarea rows="10" cols="20" name="ftexte" >
6 At W3Schools you will find all the Web-building tutorials you need, from basic HTML to
  advanced XML, SQL, ASP, and PHP.
7 </textarea>
```



Zone de sélection/liste déroulante

- La balise **<select>** permet de définir une liste de déroulante (drop-down list). Chaque choix dans la liste est défini par une balise **<option>**

Syntaxe :

```
<select>
  <option value="value1">Valeur 1</option>
  <option value="value2">Valeur2
  </option>  ...
</select>
```

Attributs de <s

- **Size**: spécifie le nombre d'option visible dans la liste
- **Multiple**: spécifie que plusieurs options peuvent être sélectionnées à la fois
- **Name, disabled**

Attributs de <option>

- **Value**: spécifie la valeur à envoyer au serveur si cette option est sélectionnée
- **Selected**: spécifie que cette option est sélectionnée par défaut
- **Label**: définit un label
- **Disabled**

Zone de sélection : exemple

...

```
<form name="f1" action="html_form_action.php">  
  <h4>une liste simple avec un choix par défaut:
```

```
</h4>
```

```
  <select name="langagedefault">
```

```
    <option value="s1">Javascript</option>
```

```
    <option selected="selected"
```

```
value="s2">PhP</option>
```

```
    <option value="s3">Java</option>
```

```
    <option value="s4">Perl</option>
```

```
  </select>
```

```
  <h4>une liste avec choix multiple avec une taille égale  
à 4: </h4>
```

```
    <select name="langagemulti" multiple="multiple"  
size="4">
```

```
      <option value="javascript">Javascript</option>
```

```
      <option value="php">PhP</option>
```

```
      <option value="java">Java</option>
```

```
      <option value="perl">Perl</option>
```

```
    </select>
```

```
</form>
```

Web - my first web site/index.html - Aptana Studio

Navigate Search Project Run Scripts Window Help

index.html My Studio

```
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>New Web Project</title>
</head>
<body>
  <form name="f1" action="html_form_action.php">
    <h4>une liste simple avec un choix par défaut: </h4>
    <select name="langagedefault">
      <option value="s1">Javascript</option>
      <option selected="selected" value="s2">Php</option>
      <option value="s3">Java</option>
      <option value="s4">Perl</option>
    </select>
    <h4>une liste simple avec une taille égale à 2: </h4>
    <select name="langagemulti" size="2">
      <option value="javascript">Javascript</option>
      <option selected="selected" value="php">Php</option>
      <option value="java">Java</option>
      <option value="perl">Perl</option>
    </select>
    <h4>une liste avec choix multiple avec une taille égale à 4: </h4>
    <select name="langagemulti" multiple="multiple" size="4">
      <option value="javascript">Javascript</option>
      <option value="php">Php</option>
      <option value="java">Java</option>
      <option value="perl">Perl</option>
    </select>
  </form>
</body>
</html>
```

My Studio index.html

une liste simple avec un choix par défaut:

Php

une liste simple avec une taille égale à 2:

Php
Java

une liste avec choix multiple avec une taille égale à 4:

Javascript
Php
Java
Perl

Source IE Writable Insert 1:3 Sign In 99

