

# **Internet et la composition de pages Web**

---

**Virginie Sans**

**Virginie.sans@univ-rennes1.fr**

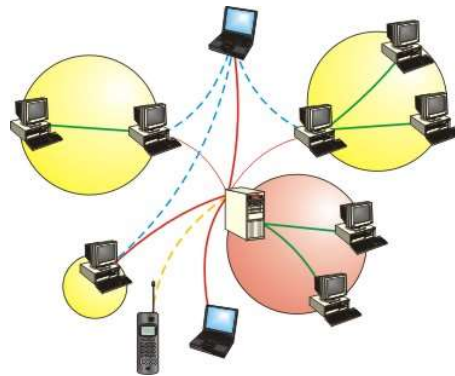
## **Introduction**

---

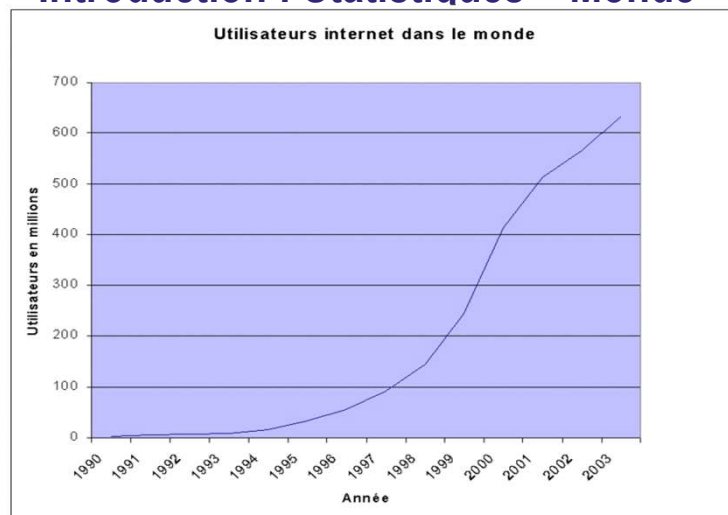
1. Introduction
2. Le principe d'Internet
3. Les langages du web

## Introduction : Historique

- 1967 : Arpanet : réseau militaire américain robuste aux pannes
- 1973 : Apparition du TCP/IP
- 1983 : Internet : Interconnexion d'Arpanet et d'autres réseaux

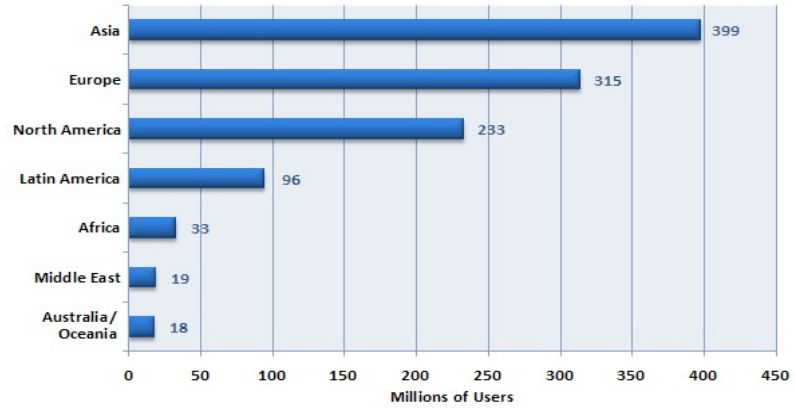


## Introduction : Statistiques – Monde



## Introduction : Statistiques - Monde

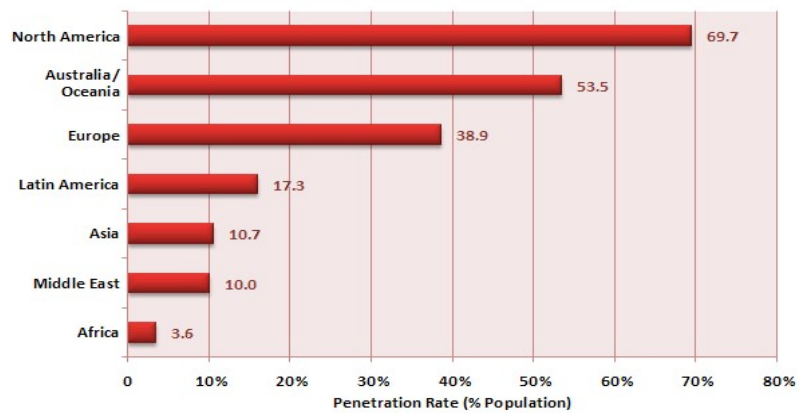
Internet Users by World Region



Copyright © www.internetworldstats.com - Mar 19, 2007

## Introduction : Statistiques - Monde

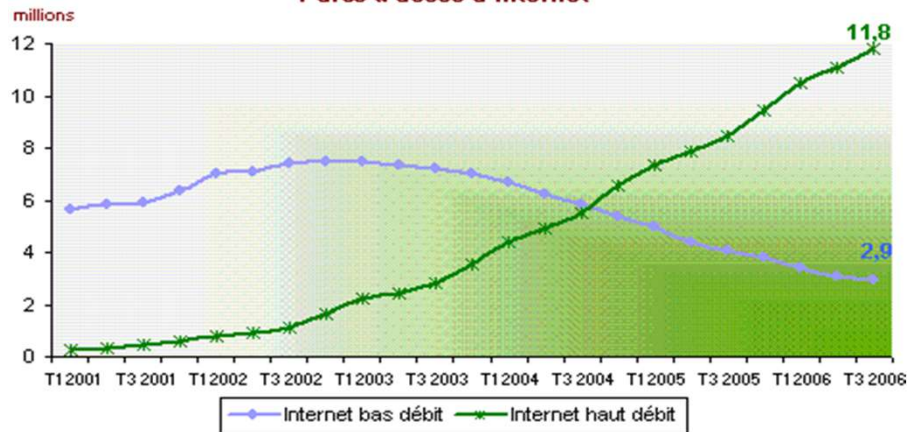
Internet Penetration by World Region



Copyright © www.internetworldstats.com - Mar 19, 2007

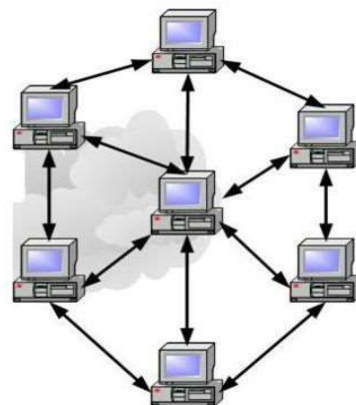
## Introduction : Statistiques - France

### Parcs d'accès à Internet



## Principe d'Internet

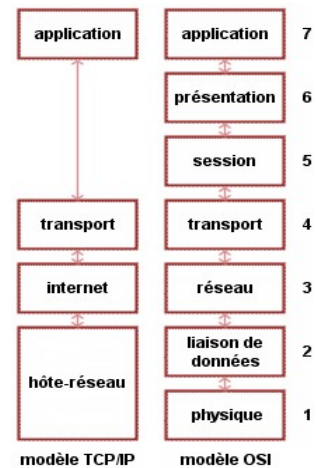
- Le réseau Internet met en contact les utilisateurs par le biais de leur matériel informatique respectif.



## Principe d'Internet : modèle TCP-IP

---

- Hôte-réseau : liaison physique et de données.
- Internet : interconnexion des réseaux / routage.
- Transport : conversation.
- Application : Telnet, TFTP, SMTP, HTTP.



## Principes d'Internet : Les couches applicatives

---

- HTTP : web
- NNTP : newsgroup
- POP / SMTP : mail
- FTP : transfert de fichiers
- DNS : correspondance entre noms et adresses IP
- SSH : connexion à distance sécurisée

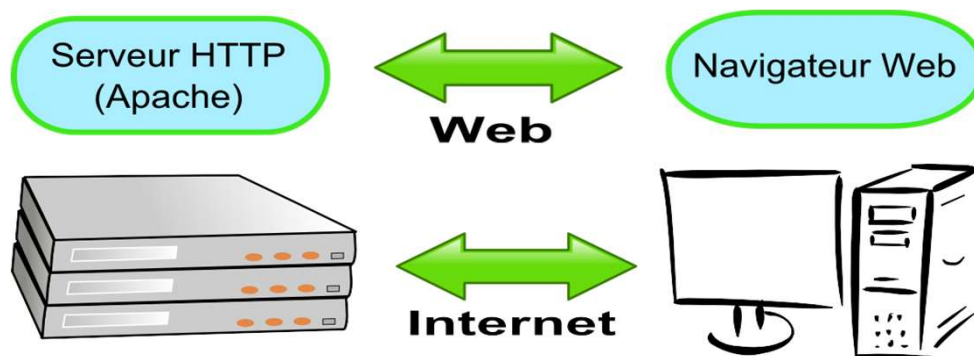
## Principe d'Internet : Le web

---

- Système hypertexte public fonctionnant sur Internet et qui permet de consulter, avec un navigateur, des pages mises en ligne dans des sites.  
L'image de la toile vient des hyperliens qui lient les pages Web entre elles.
- Communication entre un serveur (HTTP) et un client (navigateur)

## Principe d'Internet

---



## Principe d'Internet : HTTP

---

- Trame HTTP

<p>Ligne de commande (Commande, URL, Version de protocole) En-tête de requête [Ligne vide] Corps de requête</p>
---

- Méthodes : GET / POST
- HTTP 1.0 ou 1.1

## Principe d'Internet : URL

---

- Une URL est une chaîne de caractères utilisée pour adresser les ressources dans le Web
- Exemple :  
<http://www.example.com/chemin/page.html?q=req>
  - http : protocole
  - www.example.com : hôte
  - /chemin/ : chemin absolu sur le service
  - page.html : nom de la page Web
  - q=req : chaîne de requête, transmise à la page

## Les langages du web

---

- Le web fonctionne sur le protocole HTTP
- Plusieurs langages sont utilisés pour amener des pages personnelles aux utilisateurs

## Comment fonctionne le Web ?

---

- ⊕ C'est un mécanisme client-serveur.
- ⊕ Le client demande un fichier, le serveur lui donne tel qu'il est stocké – processus statique
- ⊕ Le serveur peut aussi générer un fichier en fonction de la demande du client – processus dynamique



## Ecrire pour le Web

---

- ⊕ Ce n'est pas uniquement écrire des pages en HTML, il faut penser en terme de **projet**:
  - ⊕ Définir le contenu
  - ⊕ Trouver une arborescence ergonomique
  - ⊕ Appliquer / Respecter la charte graphique
  - ⊕ Produire les pages
  - ⊕ Installer le site sur le serveur
  - ⊕ Maintenance, politique de mise à jour

## HTML, l'origine

---

- ⊕ HTML **Hyper Text Markup Language** est né en 1989 sous l'impulsion de Tim Berners Lee, " inventeur " du Web.
- ⊕ HTML est basé sur **SGML (Structured Markup Language)**, qui est une vieille norme utilisée pour la description de documents.Elle est conçue pour les grosses documentations techniques.
- ⊕ HTML est une **instance** de SGML.

## HTML, les principes

---

- ⊕ Il contient des **commandes**, implémentées par des **balises** pour marquer les différents types de texte (titres, paragraphe, listes ...), pour inclure des images, des formulaires, des liens ...
- ⊕ C'est un **langage à balisage** qui décrit la structure logique d'un document **hypertexte**. Il a volontairement été conçu pour être simple.
- ⊕ Il a évolué vers un langage de description de pages offrant des possibilités plus proches de la P.A.O.

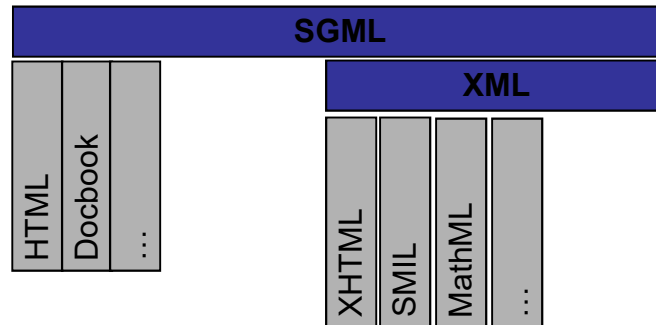
## L'hypertexte

---

- ⊕ Le langage **HTML** permet de créer des documents interactifs grâce à des **liens hypertextes**, qui relient votre document à d'autres documents.
- ⊕ En **cliquant** sur une zone de texte (ou une image, un logo) mise en évidence, on peut accéder à un **nouveau document** situé sur un autre ordinateur en n'importe quel point du globe.

## Arbre généalogique

---



## Introduction au marquage - 1

---

- ⊕ Pour décrire un fichier hypertexte, le langage HTML insère des **balises** dans le texte du document :

```
<marqueur> ici votre texte </marqueur>
```

↑  
Début de mise en forme

↑  
Fin de mise en forme

\* Synonymes: marqueur, élément, tag.

## Introduction au marquage - 2

---

- ⊕ Ces balises peuvent être insérées n'importe où dans le texte, entre 2 phrases, mots, lettres ...

```
<gras>Le <italique> cours </italique> HTML</gras>
```



**Le *cours* HTML**

## Introduction au marquage - 3

---

- ⊕ Il faut respecter une logique d'imbrication:

**Bon:**

```
<gras><italique> Le cours HTML </italique> </gras>
```

**Mauvais:**

```
<gras><italique> Le cours HTML</gras></italique>
```

## Introduction au marquage - 4

---

- ⊕ Le langage HTML est **sensible à la casse**, toujours écrire en **minuscules**.

Bon:

```
<gras><italique> Le cours HTML </italique> </gras>
```

Mauvais:

```
<GRAS><italique> Le cours HTML </italique> </GRAS>
```

```
<Gras><ITALIQUE> Le cours HTML </ITALIQUE> </Gras>
```

```
<GRAS><ITALIQUE> Le cours HTML </italique> </GRAS>
```

## Les attributs

---

- ⊕ Les balises peuvent posséder un ou plusieurs attributs qui permettent de spécifier l'action de la balise. Toujours mettre la valeur de l'attribut entre **guillemets**.

```
<marqueur attribut="argument">texte</marqueur>
```

```
<marqueur attribut1="argument" attribut2="argument">texte</marqueur>
```

## Les commentaires

---

```
<!-- Voici un commentaire HTML -->
```

```
<!--
```


```
Voici un commentaire HTML qui  
peut se placer sur plusieurs lignes
```

```
-->
```

## Que choisir pour écrire de l' XHTML?

---

- ⊕ **A la main, avec un éditeur de texte**
  - ⊕ Bolc Notes, Simple Text, Bbedit, Emacs, WordPad ...
- ⊕ **Avec un logiciel « assistant » au code HTML**
  - ⊕ PageSpinner sur MacIntosh, Amaya sur Unix, HTML-Kit sur PC ...
- ⊕ **A l'aide d'un programme dit "WYSIWYG"**
  - ⊕ Dreamweaver, Golive, Netscape composer, FrontPage, ...
- ⊕ **A l'aide d'un filtre**
  - ⊕ Les commandes enregistrer sous html, que l'on trouve dans les suites bureautiques (Word, OpenOffice, ...), dans certains logiciels de P.A.O.
- ⊕ **Conversion HTML vers XHTML avec HTML Tidy**
  - ⊕ Disponible dans de nombreuses versions sur le site du W3C

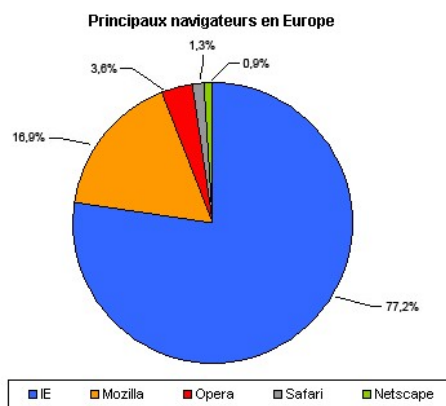


Attention aux mauvaises surprises de compatibilité !

## Que choisir pour lire HTML?

⊕ Le client doit pouvoir interpréter HTML et afficher le résultat. Ils sont divers, tournant sur des systèmes différents:

- Netscape Navigator, Mozilla, Internet explorer, Safari, Opéra, iCab, Emacs mode www, Amaya, Lynx, links, w3m ...



## Note sur les caractères accentués

⊕ Vous pouvez rencontrer le codage ASCII sur 7 bits spécifié par la norme pour afficher les caractères accentués ou spéciaux. Ceux ci devront faire l'objet d'un codage spécial au sein du fichier HTML.

é s'écrit `&eacute;`; être s'écrit `&eacirc;`tre

⊕ Les serveurs Web accepte les caractères accentués de la norme iso-8859-1. On spécifie l'encodage dans le fichier HTML.

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

## Structure de base d'un fichier HTML

```
<html>
  <head>
    ... <title>Mon premier document html</title>
  </head>
  <body>
    ... Bonjour tout le monde
  </body>
</html>
```

Une balise `<html>` contenant une seule balise `<head>` et une seule balise `<body>`.

Les balises que l'on va trouver dans le corps divisent le contenu en sections logiques, sous forme de blocs ( paragraphes, tableaux ...). On parle d'éléments de niveau bloc.

Les éléments qui représentent les propriétés du texte (strong, i) qui figurent dans un bloc sont dits "éléments de ligne".

## Analyse des balises

`<html> .. </html>`

Délimite le début et la fin du document

`<head> .. </head>`

Entête du document, contient des méta-  
informations

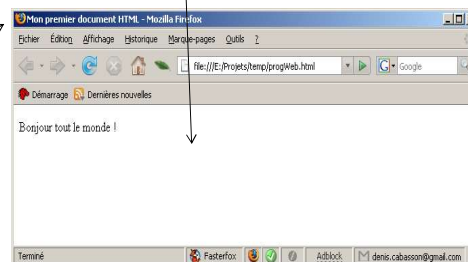
`<body> . . </body>`

Corps du document

`<title>.. </title>` titre du document

`<meta> . . </meta>` méta-informations

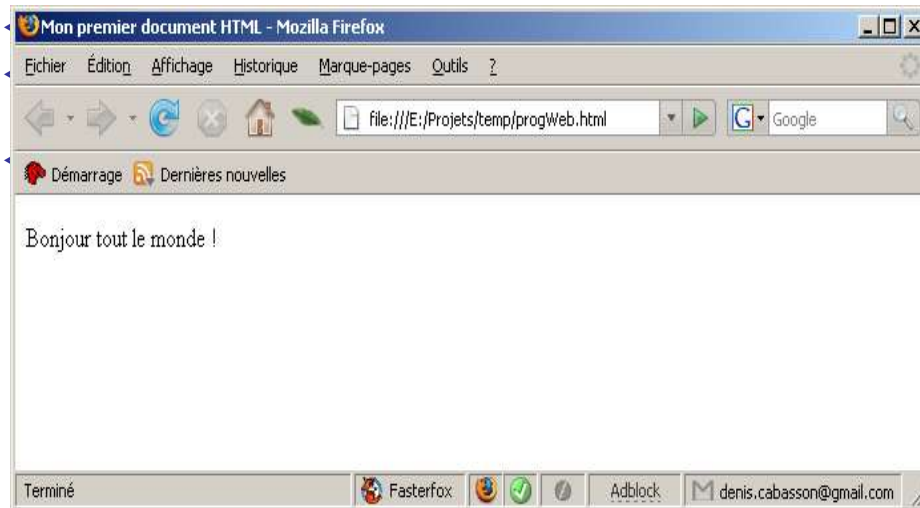
`<script> . . </script>` script ou référence





## Analyse des balises

---



## Rappels

---



- ⊕ Les noms d'éléments sont sensibles à la casse et sont écrits en **minuscules**.
- ⊕ Les noms d'attributs sont sensibles à la casse, doivent être écrits en **minuscules** et encadrés par des **guillemets**. Tous les attributs doivent recevoir **une valeur**.
- ⊕ Les balises fermantes sont obligatoires.
- ⊕ Les éléments vides sont signalés par une balise spéciale. (ex: `<br />`)

## Un peu plus loin dans HTML

---

```
<html>
<head>
<title>Mon deuxieme fichier</title>
</head>
<body>
<h2>Hello World</h2>
<p>
L'emplacement de votre nouveau mat&eacute;riel est-il &agrave; proximit&eacute; (moins de 5m) d'une prise r&eacute;seau
libre ? Si oui, relevez le num&eacute;ro de la prise (il est &eacute;crit sur une plaque de c&eacute;ramique bleue, il faut
respecter la distinction entre lettres majuscules et minuscules). Sinon, il faut faire installer une nouvelle prise par le
Service Travaux, auquel vous devrez adresser un bon de cession interne. La suite des op&eacute;rations aura lieu quand vous
aurez une prise disponible. Dans le doute, interrogez par courrier &eacute;lectronique netadm@pasteur.fr. </p>
<p>Votre demande doit nous parvenir par l'interm&eacute;diaire du correspondant informatique de votre Unit&eacute;. Elle se
composer de deux parties : un bon de cession interne envoy&eacute; au Service Informatique Scientifique et un courrier
&eacute;lectronique adress&eacute; &agrave; netadm@pasteur.fr qui contiendra toutes les informations utiles (dont le
num&eacute;ro du bon de cession). </p>
<ol>
<li> le premier &eacute;l&eacute;ment</li>
<li> le deuxi&egrave;me &eacute;l&eacute;ment</li>
<li> le troisi&egrave;me &eacute;l&eacute;ment</li>
</ol>
<p><a href="http://www.pasteur.fr/infosci/utilinfo/FAQ.html#Q1">Vous trouverez ici la suite du texte !! </a></p>
</body>
</html>
```

## Eléments de niveau bloc

---

**<h1> . . . </h1>**                      **Titre de niveau n, de 1 à 6**

**<p> . . . </p>**                      **Paragraphe**

**Et aussi: address, blockquote, div, hr, pre sans oublier body !**

## Éléments de listes

---

`<ul> . . </ul>`

Liste non triée, liste à puces

`<ol> . . </ol>`

Liste triée, liste à numéros

`<li> . . </li>`

Élément de la liste

Et aussi: dl, dt, dd

## Les liens sur un texte ...ou sur autre chose !

---



`<a> . . </a>`

Création d'un lien hypertexte, ou vers un point d'ancrage du document

Principaux attributs:

`href = url`

`name = chaîne de caractères`

`<a href = "http://www.u-cergy.fr">Université de Cergy Pontoise</a>`

Je m'appelle  
une ancre

## Les liens intra-pages



```
<a name = "chats" >Texte sur les chats</A>
<p>
.
<p>
.
<p>
.
<a href = "#chats"> En savoir plus sur les chats </a>
<a href = "autrepage.html#refautrepage">Vers la
référence d'une autre page</a>
```

## Les adresses URL

- ⊕ Les adresses du Web ou URL ( Uniform Resource Locator ) sont du type:

```
http://depinfo.u-cergy.fr/licence/l1/introinfo.html#cours
```

- ⊕ Le protocole: **http**
- ⊕ Le serveur (sous domaine + **nom de domaine**): **depinfo.u-cergy.fr**
- ⊕ Le fichier avec son chemin: **licence/l1/introinfo.html**
- ⊕ Un ancrage: **#cours**

## Les adresses URL

---

- ⊕ L'adresse indiquée dans le lien ( *URL* ) peut être **absolue**, elle inclut tout le chemin en commençant par le protocole:

```
http://www.u-cergy.fr/monfichier.html
```

- ⊕ Ou **relative**, elle n'inclut qu'une partie du chemin:

```
Pages_chats/monfichier.html
```

## Éléments de ligne

---

```
<b>texte gras</b>
```

**texte gras**

```
<i>texte italique</i>
```

*texte italique*

```
<big>texte gros</big>
```

texte gros

```
<small>texte petit</small>
```

texte petit

Et aussi: **br** (pour sauter des lignes), **code**, **sub**, **sup**, **span**, **u** (pour underline/soulignement), **strike** ...

## Encore plus loin dans l' HTML

---

```
<div style="text-align: center;">
<table border="1" cellspacing="0" cellpadding="10">
<tr>
<th>Pour Mac</th>
<th>Pour PC</th>
<th>Pour Linux</th>
</tr>
<tr>
<td>
<ul>
<li><a href="http://proxad.mac.tucows.com/blueberry/htmltextmac.html">Mode texte </a></li>
<li> <a href="http://proxad.mac.tucows.com/blueberry/beginnarmac.html">Simple </a></li>
<li><a href="http://proxad.mac.tucows.com/blueberry/htmleditmac.html">Avancé</a></li>
</ul>
</td>
<td>
<ul>
<li><a href="http://proxad.tucows.com/htmltext95.html">Mode texte </a></li>
<li> <a href="http://proxad.tucows.com/htmlbeginner95.html">Simple </a></li>
<li><a href="http://proxad.tucows.com/htmledit95.html">Avancé</a></li>
</ul>
</td>
<td>
<ul>
<li><a href="http://proxad.linux.tucows.com/xlhtml/off_html.html">Linuxberg</a></li>
</ul>
</td>
</tr>
</table>
</div>
```

## Les tableaux -1

---

`<table> . . </table>`

**Définit un tableau**

**Principaux attributs:**

```
align = position
bgcolor = color
border = n
cellpadding = n
cellspacing = n
width = n
```

## Les tableaux -2

---

```
<tr> . . </tr>
```

Définit une ligne d'un tableau

Principaux attributs :

```
align = left, center, right  
valign = top, middle, bottom  
bgcolor = color  
border = n
```

## Les tableaux -3

---

```
<td> . . </td>
```

Définit une cellule de données

Principaux attributs :

```
align = type  
valign = type  
bgcolor = color  
colspan, rowspan = n  
height, width = n
```

## Les tableaux -4

---

```
<table>
<tr>
<td> A</td>
<td> B </td>
<td> C </td>
</tr>
<tr>
<td> D</td>
<td> E </td>
<td> F </td>
</tr>
</table>
```

A	B	C
D	E	F

## Les images -1

---

```
<img> . . </img>
```

Insère une image

Principaux attributs:

```
align = left, bottom, middle, top, right
alt = text
border = n
height, width = n
src = url
```

```
<img src='monimage.jpg' />
<img src='monimage.jpg' width='5' width='10%' />
```



## Les images - 2

---

- ⊕ Attention au poids des images, il est important d'optimiser son fichier image.
- ⊕ Deux formats sont lus par les navigateurs, GIF ( Graphics Interchange Format ) et JFIF ( JPEG File Interchange Format). On utilise le GIF pour les illustrations, le JPEG pour les photos.



Attention à l'agrandissement de vos images....une image de 10x10 affichée en 100x100 grâce à width et height devient pixellisée !

## Les droits d'auteurs

- ⊕ Attention aux droits sur les images. La seule image qui vous appartient est celle que vous avez prise avec votre matériel photo. Il faut aussi l'autorisation des personnes figurant sur la photo.
- ⊕ Attention à la portée des droits d'auteurs, surtout pour une utilisation web (Pensez à faire une déclaration à la CNIL si besoin)
- ⊕ Attention aux images "libres de droits"
- ⊕ Lire les recommandations juridiques sur l'intranet.
- ⊕ Attention aux images externes...
- ⊕ `<img src=« http://www.autresite.com/image.jpg »/>`



## Exercices

- ✦ Ecrire le code d'une texte possédant un lien sur le mot Cergy Pontoise et qui pointe vers le site de l'université
- ✦ Donnez le code pour afficher une image de nom pixel.gif et de l'afficher à 50% de sa taille
- ✦ Donnez le code pour écrire un lien sur une image
- ✦ Créez un tableau à 3 colonnes , 2 lignes.
- ✦ Faire en sorte que ce tableau possède du texte en rouge dans ces cases
- ✦ Créez un tableau à 1 ligne , 2 colonnes contenant une image et du texte.

## Un fichier différent ...

```
<html>
<head>
<title>des cadres</title>
</head>
<frameset cols="25%,75%">
<frame name="menu" src="menu.html"
scrolling="yes">
<frame name="cible" src="cible.html"
scrolling="no">
</frameset>
<body>
message pour les navigateurs ne supportant
pas les cadres
</body>
</frameset>
</html>
```



## Les cadres

---

- ⊕ Ils permettent de diviser la fenêtre principale du navigateur en plusieurs sous fenêtres de taille réduite, chacune d'entre elles affichant un document différent.
- ⊕ Certains navigateurs ne les supportent pas, il est possible de les désactiver, il faut donc prévoir un fichier de substitution inscrit entre les balises:

```
<noframes> .. </noframes>
```

## Internet et la composition de pages Web

---

### Partie 2: HTML Dynamique

## XHTML

---

- ⊕ **XHTML 1.0**, est une reformulation de **HTML** en une application **XML** et trois DTDs (Document Type Definition) correspondant à celles définies par **HTML 4**.
- ⊕ La compatibilité avec les agents utilisateurs (les "clients") HTML actuels est possible en suivant un ensemble raisonnable de règles.
- ⊕ La sémantique des éléments et de leurs attributs sont définis dans la **Recommandation W3C pour le HTML**.
- ⊕ La norme actuelle est **XHTML 1.1**
- ⊕ Le site du W3C: <http://www.w3c.org>.

## Conclusion: XHTML compatible HTML

---

- ⊕ Écrire en minuscules.
- ⊕ Mettre les attributs entre guillemets.
- ⊕ Ajouter un caractère d'espacement avant la barre oblique d'une balise d'élément vide ( `<br />` ).
- ⊕ Spécifier l'encodage de caractères de deux façons ( entête xml et balise méta)

```
<?xml version="1.0" encoding=" iso-8859-1"?>
<meta http-equiv="Content-type" content="text/html; charset=iso-8859-1">
```
- ⊕ Spécifier les cibles d'hyperliens (balise a) en utilisant à la fois les attributs id et name.
- ⊕ Assigner une valeur aux attributs HTML booléens.

```
<input type="radio" checked="checked">
```

## Les limites d' HTML / XHTML

---

- ⊕ Langage de base du Web
- ⊕ Langage simple, limité et statique
- ⊕ D'autres langages, extensions, programmes sont venus se greffer à XHTML pour en augmenter les possibilités.
- ⊕ XHTML offre les aspects XML mais ce n'est pas suffisant pour toutes les applications que l'on souhaite mettre en œuvre sur le web.

## Les formulaires

---

L'élément `<form>` permet de créer des formulaires, qui peuvent contenir des cases à cocher, des boutons radio, des listes déroulantes ... les données recueillies sont transmises à un programme qui s'exécutera sur le serveur web et vous retournera le résultat.

```
<form action="mon_programme.php">
```

```
... éléments du formulaire
```

```
</form>
```

## Éléments de formulaire

---

```
<form name="mon_formulaire" action="mon_programme.php">
<input type="text" name="nom" value="" size="60" />

<input type="radio" name="rad1" value="" size="60" />choix1
<input type="radio" name="rad2" value="" size="60" />choix2

<input type="checkbox" name="chk1" value="" size="60" />choix1
<input type="checkbox" name="chk2" value="" size="60" />choix2

<select name="liste">
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
</select>

<input type="submit" name="envoyer" value="envoyer" />
</form>
```

## Éléments de formulaire

---

```
<form name="my_form" action="mon_programme.php"> ... </form>
```

### Principaux attributs

**name = "chaîne"** spécifie le nom sous lequel le formulaire peut-être identifié, par exemple en Javascript , je peux y faire référence avec l'expression: document.my\_form

**action = "URL"** spécifie l'URL à laquelle le contenu est envoyé

**method= "GET" ou "POST"**, spécifie la méthode HTTP utilisée pour transmettre les données. Avec GET elles sont ajoutées à l'URL, avec POST elles sont envoyées au serveur dans le corps du message.

## Éléments de formulaire

---

```
<input name="my_choice" type="radio"> ... </input>
```

Définit les entrées de données dans le formulaire

### Principaux attributs

`name = "chaine"`, associe un nom aux données entrées dans cet élément `input` ( ex: `civilite` pour un choix Mme, Mlle, Mr de type `radio`, on affectera dans le script PHP la valeur cochée à la variable `civilite`, ce qui donne en PHP:

```
$civilite=$_POST[civilite]
```

**Types possibles = "button, checkbox, file, hidden, image, radio, reset, submit, text ..."**

## Éléments de formulaire

---

```
<select name="my_list">
<option value="10">10</option>
</select>
```

Liste d'options sélectionnables

### Principaux attributs

`name = "chaine"`, associe un nom aux données entrées dans cet élément `select`

`value = "valeur"`, spécifie la valeur de l'élément `option`. Si cet élément est omis, c'est le contenu de l'élément `option` qui est envoyé au CGI.

## Les langages du web : CSS

---

- CSS : Cascading Style Sheets
- 1996 : CSS 1.0, actuellement la seule implémentée
- Langage de présentation d'un document HTML ou XHTML
- Issu de la séparation entre le contenu et la présentation
- Il est encapsulé dans la page HTML (ou dans un fichier lié) pour être interprété par le client.

## Les feuilles de styles CSS

---

- ⊕ Elles permettent de changer la mise en forme d'une page HTML ou XHTML sans en modifier son contenu.
- ⊕ Le langage CSS spécifie l'apparence des blocs de texte ou image, mais il peut contrôler d'une manière très précise le positionnement des objets, les bordures, les marges, le recouvrement ...
- ⊕ La norme est consultable sur le site du consortium Web. Les navigateurs n'implémentent qu'une petite partie du langage.



## Les langages du web : CSS

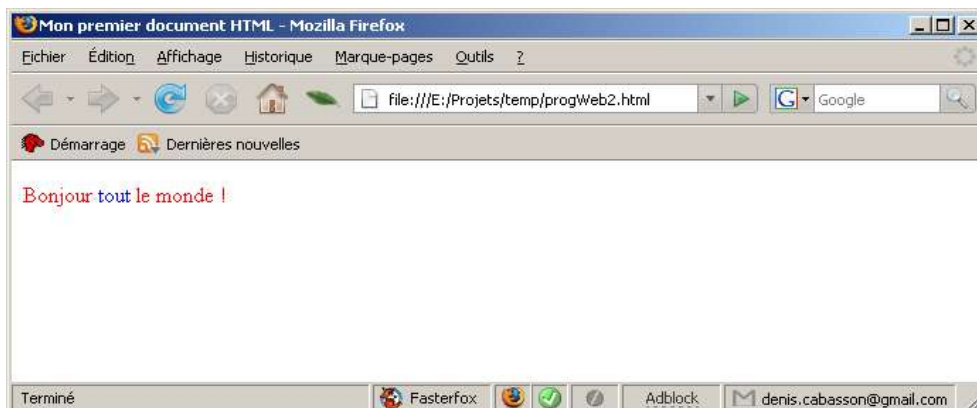
---

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="fr">
  <head>
    <style type="text/css">
      p { color : red; }
      span { color : blue; }
    </style>
    <title>Mon premier document HTML</title>
  </head>

  <body>
    <p>Bonjour <span>tout</span> le monde&nbsp;<!--</p>
  </body>
</html>
```

## Les langages du web : CSS

---



## L'HTML Dynamique

### CGI - SSI - ePerl – PHP – JSP – ASP - Javascript

#### Coté serveur

- Common Gateway Interface, est un programme qui s'exécute sur le serveur.
- SSI sont des fonctions spécifiques du serveur Apache.
- Java Server Pages, en Java
- ⊕ ePerl, PHP sont des modules que l'on installe sur le serveur.
- ⊕ Le code est interprété par le serveur
  - ⊕ ex: affichage personnalisé pour un client, filtrage, connexion à des bases de données ..

#### Coté client

- ⊕ On insère des "morceaux de code informatique" dans la page HTML, qui la rendent "dynamique". Le code est interprété par le client.
- ⊕ Javascript (Netscape), Jscript (Microsoft)

## Les langages du web : JavaScript

---

- 1995 : invention du JavaScript
- Langage de script exécuté dans le navigateur (client) et permettant un dynamisme des pages
- Il est encapsulé dans la page HTML (ou dans un fichier lié) pour être exécuté sur le client.

## JavaScript

---

```
<script language="javascript"> ... </script>
```

Inclusion des scripts avec l'élément script. Cet élément peut être imbriqué dans l'en-tête (<head>) ou dans le corps du document (<body>).

Note: lorsqu'un navigateur examine les composants d'un script, il commence par le début du fichier, il est important de les définir avant de les utiliser.

## Les langages du web : JavaScript

---

```
<html>
<head>
  <script type='text/javascript'>
    window.onload= setTimeout( fonction()
{ alert( 'chargement de la page terminé il y a une seconde et demi' ) }, 1500 );
  </script>
</head>
<body>
  <p>Une page HTML tout simple</p>
</body>
</html>
```

## **Les langages du web : JavaScript**

### **Avantages et Inconvénients**

---

- **Avantages :**
  - Plus grande convivialité coté client
- **Inconvénients :**
  - Compatibilité entre les navigateurs
  - Problème de sécurité
  - Désactivation du JavaScript (~10% des navigateurs)

## **Les langages du web : Autres langages coté client**

---

Il existe d'autres langages interprétés sur le client (et insérés dans une page HTML) :

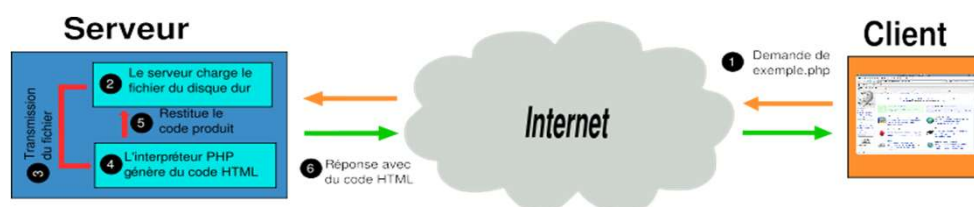
- Les applets Java
- Les animations flash
- Les ActiveX (uniquement sous IE)

## Les langages du web : PHP



- PHP : PHP Hypertext Preprocessor
- 1994 : création par Rasmus Lerdorf
- Langage interprété sur un serveur HTTP (Apache ou IIS)
- Produit en sortie un flux HTML envoyé au client.

## Les langages du web : PHP



## Les langages du web : PHP

---



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="fr">
  <head>
    <title>Mon premier document PHP</title>
  </head>
  <body>
    <?php
    $pseudo = "Robert";
    echo "<p>Bienvenue " . $pseudo. "</p>";
    ?>
  </body>
</html>
```

## Les langages du web : PHP

---



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="fr">
  <head>
    <title>Mon premier document PHP</title>
  </head>
  <body>
    <p>Bienvenue Robert</p>
  </body>
</html>
```