

World Wide Web

M1 Architecture de l'Information
Remise à niveau en informatique
Lionel Médini



Objectifs des cours et des TP (partie Web)

- ❑ Découvrir ce qu'est le Web et comment ça marche
 - environnement réseau
 - mécanismes de base du Web
- ❑ Être capable de
 - créer une page Web simple
 - langage XHTML
 - mettre en forme une page ou un ensemble de pages
 - langage CSS niveaux 1 et 2
 - rajouter des contenus dynamiques dans une page
 - scripting côté client
 - créer un site Web basique
 - installation d'un serveur
 - publication de contenus
 - scripting côté serveur

World Wide Web

❑ Principe original : accéder à des documents textuels

- situés sur des machines accessibles par Internet
- reliés entre eux par un mécanisme de lien « hypertexte »

❑ Actuellement : servir des ressources

- De différentes natures : texte, image, son, vidéo, contenu applicatif...
- Hypermédia
- Interactives
- Permettant à l'utilisateur d'accéder à un service donné : rechercher de l'information, acheter un objet, accéder à ses mails, consulter ses comptes en banque...

➔ Nombreuses évolutions techniques

Genèse du Web : la notion d'hypertexte

□ Principe

- S'abstraire de l'aspect linéaire du document textuel
- Mécanisme intellectuel permettant le cheminement d'une information à une autre → navigation, butinage, transclusion

□ Historique

- 1945 : invention de la notion d'hypertexte
 - Vannevar Bush, *As We may think*, *Atlantic Monthly*, 1945
- 1965 : invention du terme d'hypertexte
 - Ted Nelson, projet Xanadu
- Années 1960 : premier système hypertexte fonctionnel
 - NLS (oNLine System), Douglas Englebart
- 1987-2004 : diffusion du logiciel HyperCard
 - Programme et environnement graphique de programmation, créé par Bill Atkinson pour Mac OS, livré avec les Mac
- 1987 : première conférence HyperText
 - Sponsorisée par l'ACM

Naissance du Web (1989-1991)

- ❑ Mars 89 : projet de création d'un hypertexte documentaire distribué sur le réseau du CERN
 - Origine : Tim Berners-Lee, puis Robert Cailliau (1990)
 - Choix des technologies TCP/IP et ouverture de la première connexion du CERN avec Internet
 - Mise au point des **3 technologies de base du Web : URL, HTML et HTTP**
- ❑ Septembre 90 : 1er site Web fonctionnel (mode texte)
 - 1^{er} serveur Web : nxoc01.cern.ch
 - 1^{er} navigateur Web : WorldWideWeb (rebaptisé plus tard Nexus), développé en Objective C
- ❑ Août 1991 : publication du projet WorldWideWeb dans un message sur UseNet
- ❑ Décembre 91 : démonstration publique à la conférence Hypertext'91 (San Antonio)



Le premier serveur Web, un NeXT Cube (source : Wikipédia)

Historique du Web

- ❑ **1993 : Mosaic : premier navigateur « grand public »**
 - Marc Andreessen, NCSA : plateformes X, puis Mac et Windows
 - affichage d'images (GIF et XBM) dans les pages Web
 - prise en charge de formulaires interactifs
- ❑ **01/10/1994 : création du W3C**
 - à l'initiative du CERN (Genève) et du MIT (Boston)
 - président : Tim Berners-Lee
 - but : standardisation et développement du Web
- ❑ **1994 : Apparition des navigateurs privés**
 - M. Andreessen crée Netscape Communications Corp.
- ❑ **1995 : Microsoft lance la « guerre des navigateurs »**
 - Apparition d'Internet Explorer pour Windows 95
- ❑ **1995 : Altavista : premier « gros » moteur de recherche**
- ❑ **1996 : Navigateur Opera**
- ❑ **1998 : Apparition de Google**
- ❑ **2003 : Apple lance Safari**
- ❑ **2004 : Première version de Mozilla Firefox**
- ❑ **2004 : première conférence « Web 2.0 »**
- ❑ **2008 : Google lance Google Chrome**
- ❑ **Actuellement : explosion du Web mobile**
- ❑ **En cours de développement : Web sémantique, de données, des objets...**

Nombre de sites référencés

1990 : 1 (CERN)

1991 : premier site hors d'Europe (SLAC, Stanford)

1992 : 26

Juin 1993 : 130

Juin 1994 : 2738

Juin 1995 : 23 500

Janvier 1996 : ~ 100 000

Avril 1997 : > 1 million

Février 2000 : > 11 millions

Février 2007 : > 100 millions

Février 2009 : ~ 216 millions

Le Web : première définition

□ World Wide Web (WWW, Toile)

□ Définition

- ensemble de données disponibles sur les serveurs accessibles par Internet
- pouvant être visualisées et/ou utilisées avec un navigateur Web
- regroupées sous forme de pages et de sites

□ Rappel : Web \neq Internet

- Internet = support de communication
- Web = une partie des contenus circulant sur l'Internet

Page Web

□ Une page Web, c'est

- un document multimédia
 - textes, images, vidéo...
 - possibilité d'interaction
 - liens hypertextes
- décrit élément par élément
 - titre, morceaux de texte, images...
- avec un langage de description
 - HTML (1992) ou XHTML (2002)
- stocké dans un fichier
 - .html (ou .htm, .xhtml)

□ Une page Web

- est **calculée** et affichée par un navigateur
- est localisée sur Internet à l'aide d'une adresse (URL)
- permet d'accéder à d'autres pages en suivant des liens

Site Web

❑ Regroupement de pages Web

- autour d'une thématique commune
- reliées entre elles par des liens hypertextes
- émanant d'une même entité (organisation, entreprise, particulier...)
- accessibles à partir d'une page d'accueil commune
- accessibles à partir d'une URL de base commune
- en général, localisées sur un même serveur

❑ Organisation hiérarchique

- structure sous forme d'arborescence de dossiers et de fichiers (comme un disque local)
- permet de définir des chemins relatifs
- **exceptions** (de plus en plus fréquentes) : pages générées dynamiquement

❑ Liens vers et à partir d'autres sites

- permettent de naviguer d'un site à un autre
- composent la « toile » mondiale (World Wide Web)

Le « Web 2.0 »

❑ Principe du Web « classique »

- ce sont les organisations qui détiennent des sites qui décident de l'information qui y figure
- ➔ Communication de type « one-to-many » (= diffusion)

❑ Principe du « Web 2.0 »

- donner le contrôle de l'information aux utilisateurs
- faire émerger des « réseaux sociaux »
- ➔ chacun peut déposer des contenus
 - pour donner son avis sur un sujet donné (blogs)
 - pour partager ses documents (images, vidéos...)
 - pour étiqueter (« tagger ») des contenus existants

❑ Aspects techniques

- outils (protocoles, clients, serveurs) identiques à ceux du Web classique
- nécessite plus de puissance de calcul (pages dynamiques)
- nécessite plus d'espace de stockage (contenus envoyés par les utilisateurs)

❑ Exemples de sites « Web 2.0 »

- Wikipédia, Del.icio.us, Technorati, Flickr, Picasa Web album, Dailymotion, YouTube, Kartoo...

Popularité du Web

□ Pour l'internaute

- Accessibilité « world-wide »
- Interfaces graphiques conviviales
- Interactivité et richesse des services
- Grande quantité d'informations disponibles (Web « 1.0 »)
- Possibilité de contribuer en tant qu'utilisateur (Web « 2.0 »)

□ Pour le développeur

- Simplicité des développements (technologies de base...)
- Liens avec les outils applicatifs installés sur le serveur (shell, Perl, Java...)
- Indépendance par rapport aux plateformes des clients

Usages du Web (2008)

- ❑ Consultation simple (Web 1.0)
 - Navigation
 - Recherche d'informations
- ❑ Divertissement
 - TV, radio, musique, vidéo en ligne
 - Information
 - Jeux
- ❑ Communication
 - Asynchrone (Webmail)
 - Synchrone (Web chat, Webconférence)
- ❑ Web 2.0
 - Travail collaboratif : partage / édition de documents sur des intranets
 - Autres sites participatif (blogs...)
 - Réseaux sociaux
- ❑ Consommation de services
 - Sites marchands, enchères
 - Autres services en ligne : banque, administration...

Sources : Journal du Net (http://www.journaldunet.com/cc/01_internautes/inter_usage_fr.shtml), Ipsos (<http://www.ipsos.fr>), Carrefour Numérique (<http://carrefour-numerique.cite-sciences.fr>)

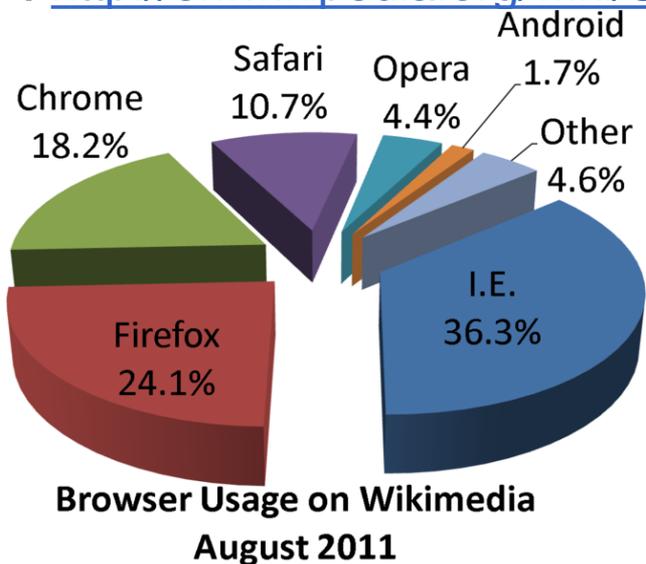
Répartition des clients (2011)

☒ **Fixes**

Source	Internet Explorer	Firefox	Chrome	Safari	Opera *	Mobile	Unknown
Net Applications	52.7%	21.5%	13.5%	8.1%	2.9%	5.5%	
StatCounter	42.5%	28.0%	22.1%	5.2%	1.7%	7.0%	
W3Counter	35.6%	27.7%	19.6%	6.4%	2.3%		8.4%
Wikimedia *	36.8%	25.0%	16.8%	10.8%	4.2%	9.8%	

(*) certaines sources comptent deux fois les navigateurs mobiles

- Source : http://en.wikipedia.org/wiki/Usage_share_of_web_browsers



- Source : http://en.wikipedia.org/wiki/Web_browser

Évolution des clients (2012)

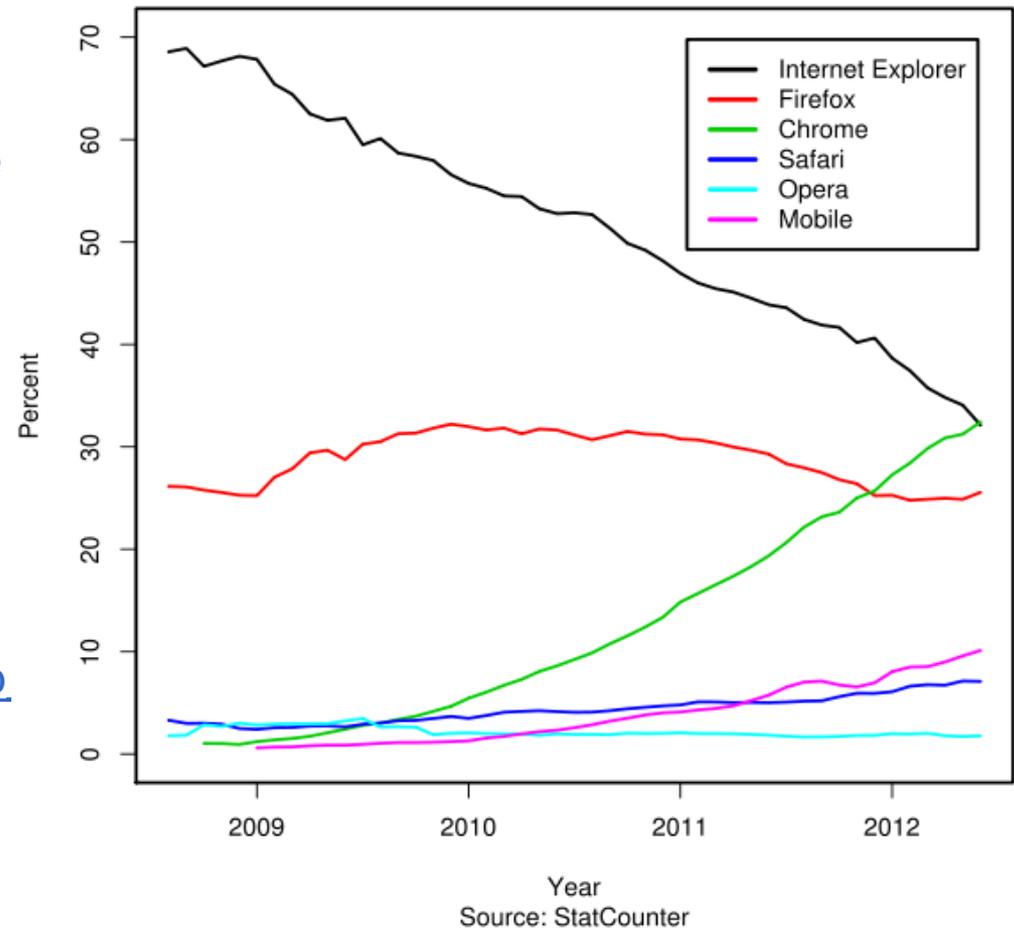


- Google Chrome rattrape Microsoft IE et double Mozilla Firefox

Source :

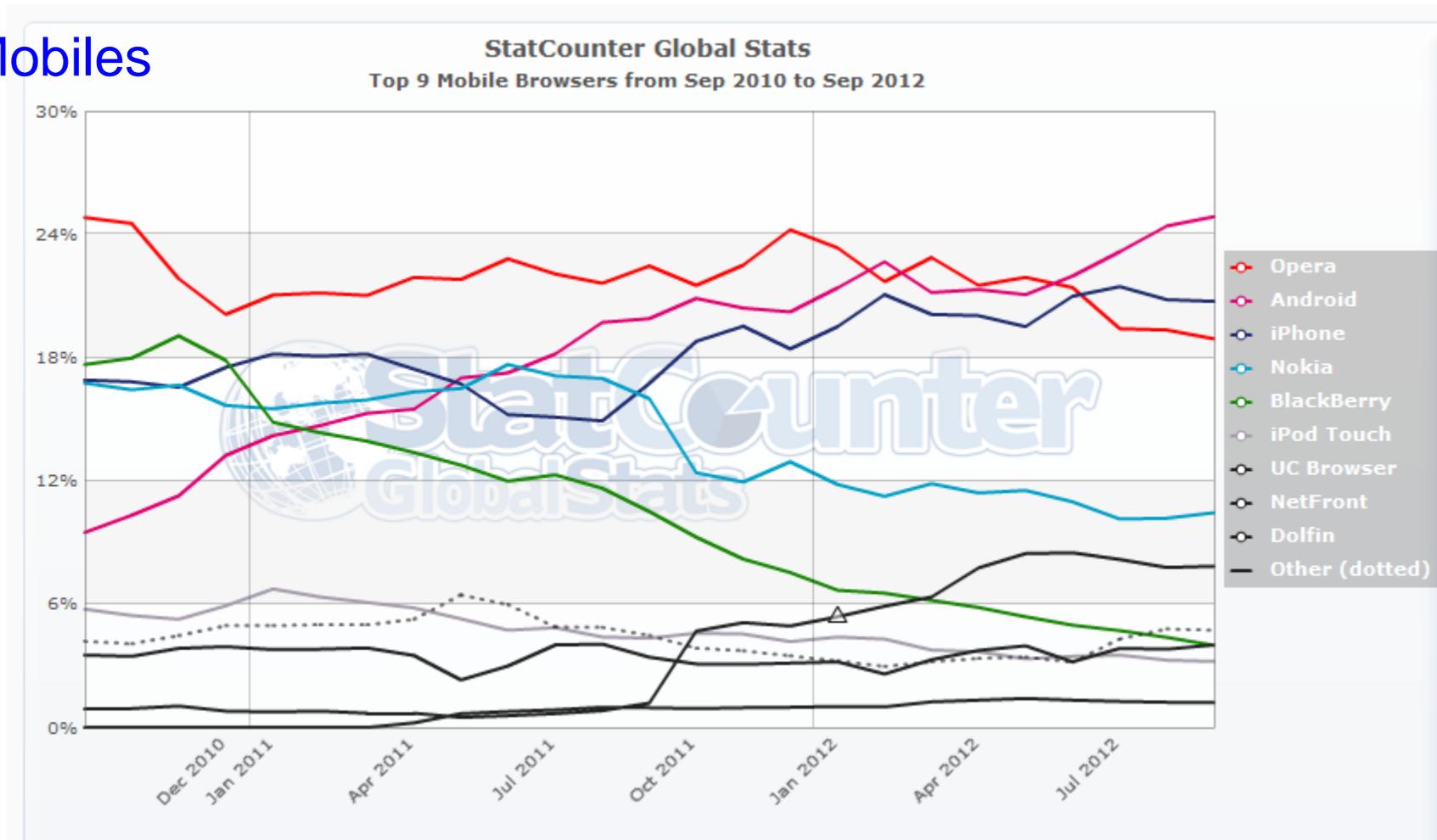
http://en.wikipedia.org/wiki/Web_browser

Usage share of web browsers



Répartition des clients (2012)

☑ Mobiles



- Source : http://gs.statcounter.com/#mobile_browser-ww-monthly-201009-201209

Les 3 mécanismes de base du Web

□ URL

- Le Web permet d'accéder à un ensemble de ressources
- Le mécanisme de localisation peut faire appel au protocole DNS

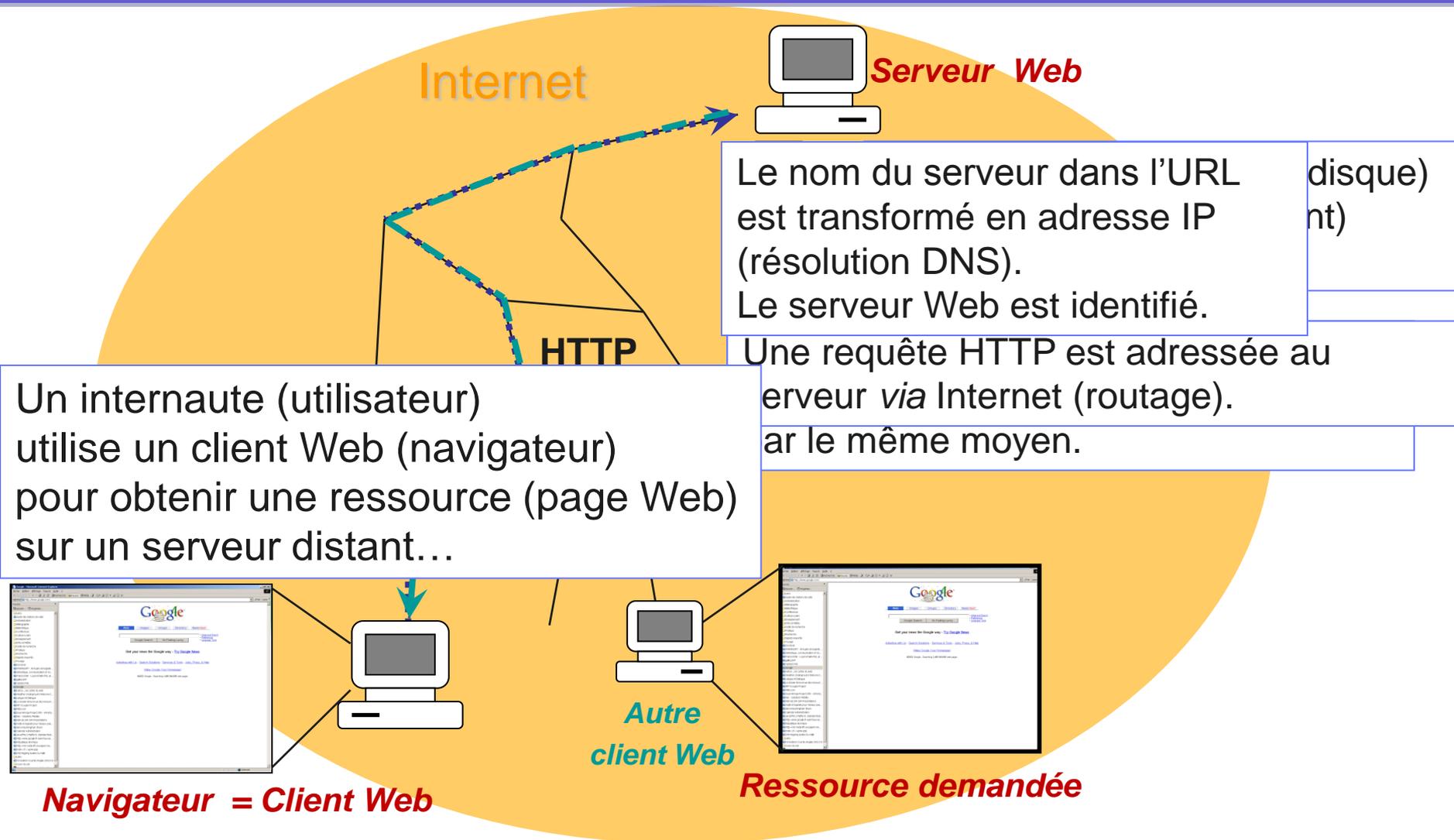
□ HTTP

- Protocole de niveau applicatif
- Paradigme client-serveur
- Protocole sans état (pas de « mémoire » des transactions précédentes)

□ HTML

- Langage de description de « pages Web »
- Texte, images et autres objets
- Liens hypermédias entre les pages
- Programmation déclarative

Fonctionnement du Web en un schéma



Les forces en présence

Côté serveur

❑ Serveur Web

- attend les requêtes HTTP et y répond
- deux façons de fournir des ressources
 - statique : le serveur renvoie les ressources dont il dispose
 - dynamique : la ressource est générée à la réception de la requête

❑ Machine serveur

- environnement contrôlé
- doit être dimensionnée en fonction de la charge attendue
- remarque : on parle de serveur Web pour désigner la machine qui héberge le programme serveur (abus de langage)

❑ Utilisateurs

- Webmaster : administration
- Web designer : conception et réalisation des contenus

Côté client

❑ Client Web

- identifient les serveurs sur Internet
- demandent les ressources aux serveurs
- affichent les ressources aux utilisateurs
- peuvent effectuer des traitements complémentaires (scripts, plugins)

❑ Machines clientes

- environnements variables
- pas d'accès possible aux autres logiciels installés (sauf plugins)

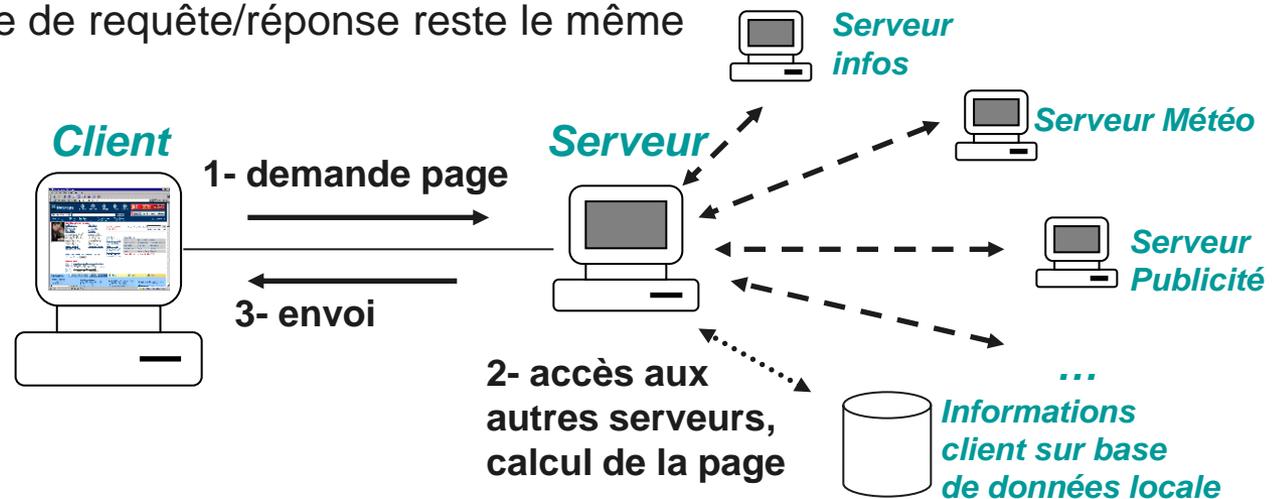
❑ Utilisateurs

- « utilisateurs finaux »
 - Identifiés ou non
 - peuvent correspondre à des « profils » modélisés
 - objectifs : utiliser le service offert par l'application Web

Pages générées dynamiquement

□ Le serveur doit calculer la ressource avant de la renvoyer

- en exécutant des « modules » (programmes)
 - sécurité, scripting, redirection...
- avec un langage de programmation côté serveur
 - PHP, ASP, Java, Python...
- en fonction des données
 - indiquées par le client (formulaires, cookies)
 - présentes sur le serveur (fichiers de données, pages statiques)
 - présentes sur d'autres serveurs (web, bases de données, mail...)
 - ❖ le serveur considéré est le client d'autres serveurs
 - ❖ Le mécanisme de requête/réponse reste le même



Comment choisir une solution logicielle ?

□ En fonction

- du type d'application visé
- de ses préférences (langage / environnement de développement)

□ Exemples

- Site statique
 - Serveur « simple » : Apache, Node.JS
- Application « simple » (scripting côté serveur)
 - Serveur simple + langage de scripts (PHP, Servlets/JSP...)
- Site éditorial
 - Articles simples : Drupal, PHP nuke, SPIP, CMS Made Simple
 - Wiki : WikiMedia, DokuWiki
 - Blog : WordPress, Over-Blog
- Application complexe
 - Système de gestion de contenus (CMS) + « frameworks Web »
 - ❖ PHP : Joomla!, Moodle
 - ❖ Java : OpenCMS, Alfresco
 - ❖ Python : Zope, Plone
 - ❖ Ruby on Rails : BrowserCMS

Navigateur Web (browser, butineur)

□ Fonctionnalités

- de base
 - client HTTP(S)
 - moteur de rendu HTML
 - moteur de scripts
- extensions
 - support de plugins
- « cosmétiques »
 - historique, signets, onglets...

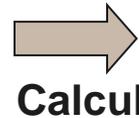
□ Offre logicielle

- Mozilla Firefox, Google Chrome/Chromium, Opera, Apple Safari, Microsoft Internet Explorer

Fichier HTML

```
<html>
<head>
<title>Universit&eacute;
; Claude Bernard Lyon
1</title>
<meta http-
equiv="Content-Type"
content="text/html;
charset=iso-8859-1">
</head>
</html>
```

Description
du document



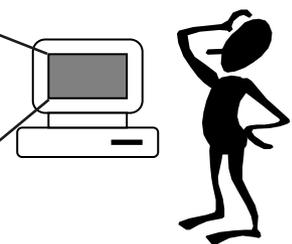
Calcul

Navigateur



Présentation
à l'utilisateur

Utilisateur



Interpréteurs de scripts

❑ Scripts

- éléments de programmes (identifiés par l'élément **<script>**)
 - à l'intérieur des pages HTML ou dans des fichiers séparés
 - écrits dans des langages de programmation spécifiques (JavaScript, VBScript)
 - exécutés (interprétés) par le navigateur de l'ordinateur client
- soumis à des restrictions de sécurité (pas d'accès aux fichiers du client, pas d'exécution d'autres programmes...)

❑ Exemples

- gestion du navigateur (lancer des fenêtres, créer des info-bulles...)
- gestion de la navigation (vérifier qu'un formulaire est bien rempli avant de l'envoyer...)
- ➔ documents plus interactifs et dynamiques qu'avec (X)HTML seul

❑ Il est possible – mais pas recommandé – de désactiver l'interpréteur Javascript du navigateur

Exécution d'applets Java

□ Java

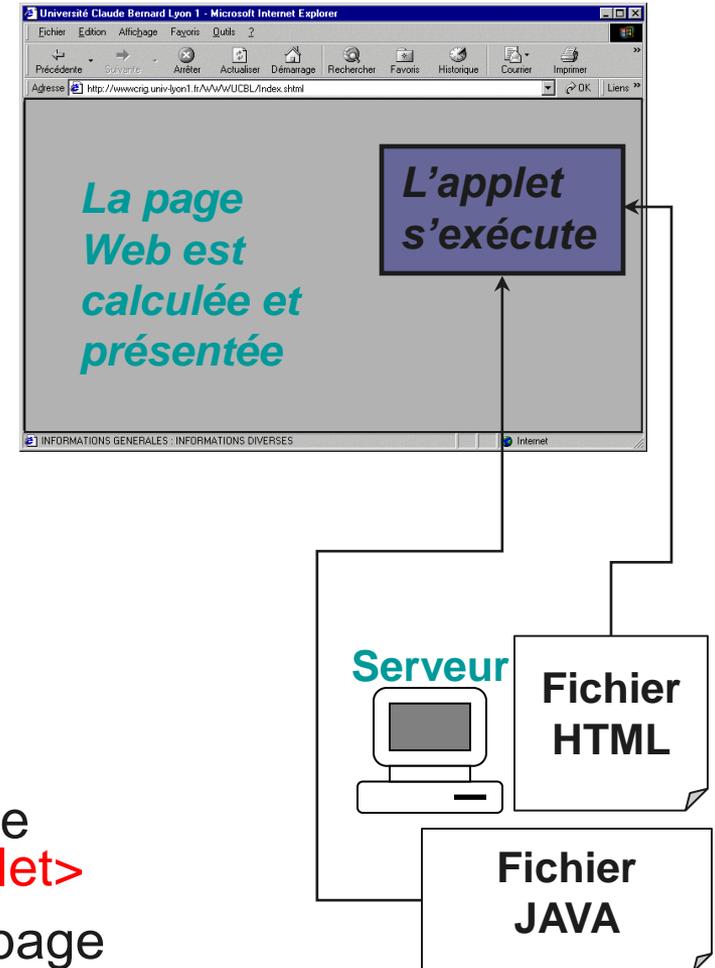
- langage de programmation qui peut fonctionner sur **tout** type de machine (matériels et systèmes d'exploitation différents)
- doit être installé sur la machine avant de pouvoir exécuter des programmes

□ Applet

- élément de programme Java
- restrictions de sécurité au niveau des fonctionnalités autorisées

□ Utilisation

- dans une page Web, on spécifie le nom de l'applet à exécuter dans un élément **<applet>**
- l'applet s'exécute dans une portion de la page présentée à l'utilisateur



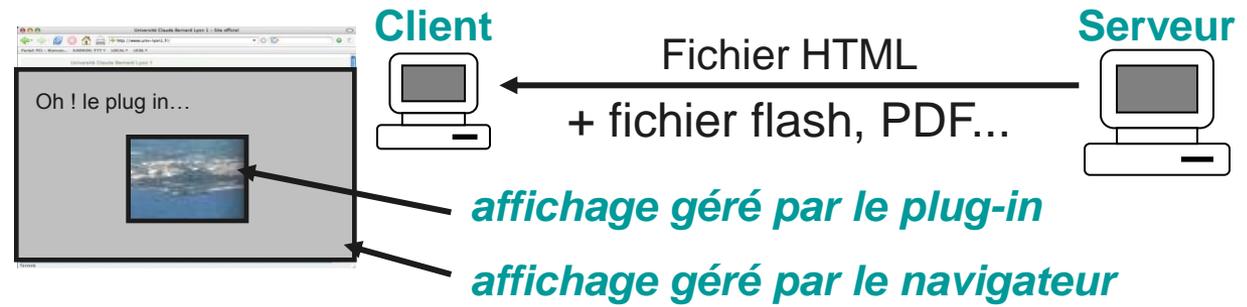
Plug-in

□ Principe

- étendre les possibilités du navigateur pour présenter des éléments HTML (décrits par l'élément **<object>**) non gérés par le navigateur
- le navigateur donne toute ou une portion de la page Web à un programme « branché » (plug-in) qui prend en charge l'exécution ou l'affichage (cf. applet Java)
- S'il n'est pas disponible, on peut en général le télécharger et l'installer

□ Exemples

- Flash / Shockwave : documents multimédias
- Real Audio / Vidéo : sons ou vidéos
- Acrobat : documents PDF



Introduction aux langages à balises

□ Donnée

- valeur associée à un type de données
 - Décrite par des caractéristiques de forme
 - Indépendante de son interprétation : donnée « brute »

□ Élément d'information

- ensemble de données faisant « sens »
 - On parle aussi de « grains d'information »

□ Document

- regroupement cohérent d'éléments d'information
 - Autour d'une thématique commune
 - Selon une structure donnée

Introduction aux langages à balises

- ❑ **Information structurée : bases de données**
 - Éléments d'information stockés séparément
 - Accès par requêtes
 - Traitements facilités
- ❑ **Information non structurée : corpus documentaires**
 - Éléments d'information stockés sous forme de textes
 - Accès par recherche d'information
 - Traitements complexes
- ❑ **Information semi-structurée : langages à balises**
 - Éléments d'information stockés dans des documents
 - Permet les deux types d'accès et de traitements

Introduction aux langages à balises

- **Balise** : signe marquant une position particulière
 - Signe = élément d'information
 - Marquer = permettre la distinction
 - Ex : fusée de détresse, signal radio, « tag » HTML
 - Position = par rapport à l'espace considéré
 - Ex : espace 3D (avion), 2D (bateau), 1D (document)
 - Particulière = en général, la position d'un élément de l'espace qu'on cherche à repérer
 - Ex : aéroport, bateau, élément d'information
- ⇒ **Type d'information facilement repérable permettant d'identifier d'autres éléments informationnels (i.e. « méta-information »)**

Introduction aux langages à balises

- ❑ **Balisage documentaire : utilisation de balises**
 - Pour marquer des points précis d'un document
 - Pour marquer des zones (segments) de document
 - Balisage de début et de fin de zone
 - Pour structurer le document
- ⇒ **Utilisation de plusieurs types de balises**
 - ⇒ En fonction du type d'élément à marquer
 - ⇒ En fonction du type de marquage (point, début, fin)
- ❑ **Langage à balises : langage de description de documents utilisant ces techniques de balisage**
- ❑ **Élément : une balise et son contenu**

Introduction aux langages à balises

□ Quelques langages à balises

- Langages de *balisage procédural* : permettent de décrire la mise en forme (formatage) d'un document
 - Ex : PS, RTF, TeX, HTML
- Langages de *balisage descriptif* : se contentent de décrire les données, sans but de traitement
 - Ex : RDF, OWL, MathML
- *Méta-langages* de balisage : permettent de définir des langages de balisage
 - Ex : SGML, XML

Introduction aux langages à balises

□ Le langage Postscript

```
%!PS-Adobe-3.0
%%Title: Microsoft Word -
Document1
%%Creator: PSCRIPT.DRV
Version 4.0
%%CreationDate: 03/02/02
10:47:00
...
%%EndComments
%%BeginProlog
%%BeginProcSet:
Pscript_Res_Emul 1.0 0
/defineresource
where{pop}{userdict
begin/defineresource{userdict/R
esources 2
...
}ifelse}bind readonly def
end}ifelse
%%EndProcSet
```

```
%%BeginResource: file
...
%%EndResource
...
%%EndProlog
%%BeginSetup
...
%%BeginFeature: *PageSize
A4
...
%%EndFeature
...
%%EndSetup
%%Page: 1 1
%%BeginPageSetup
...
%%EndPageSetup
...
```

```
%%IncludeFont: Courier
...
(Courier) cvn /Type1
...
(Essai impression)S
...
(%%[ Page: 1 ]%%) =
%%PageTrailer
%%Trailer
%%DocumentNeededFonts:
%%DocumentSuppliedFonts:
/Pscript_Win_Driver /ProcSet
findresource dup /terminate get exec
Pscript_Win_Compat dup /terminate
get exec
%%Pages: 1
(%%[ LastPage ]%%) =
%%EOF
```

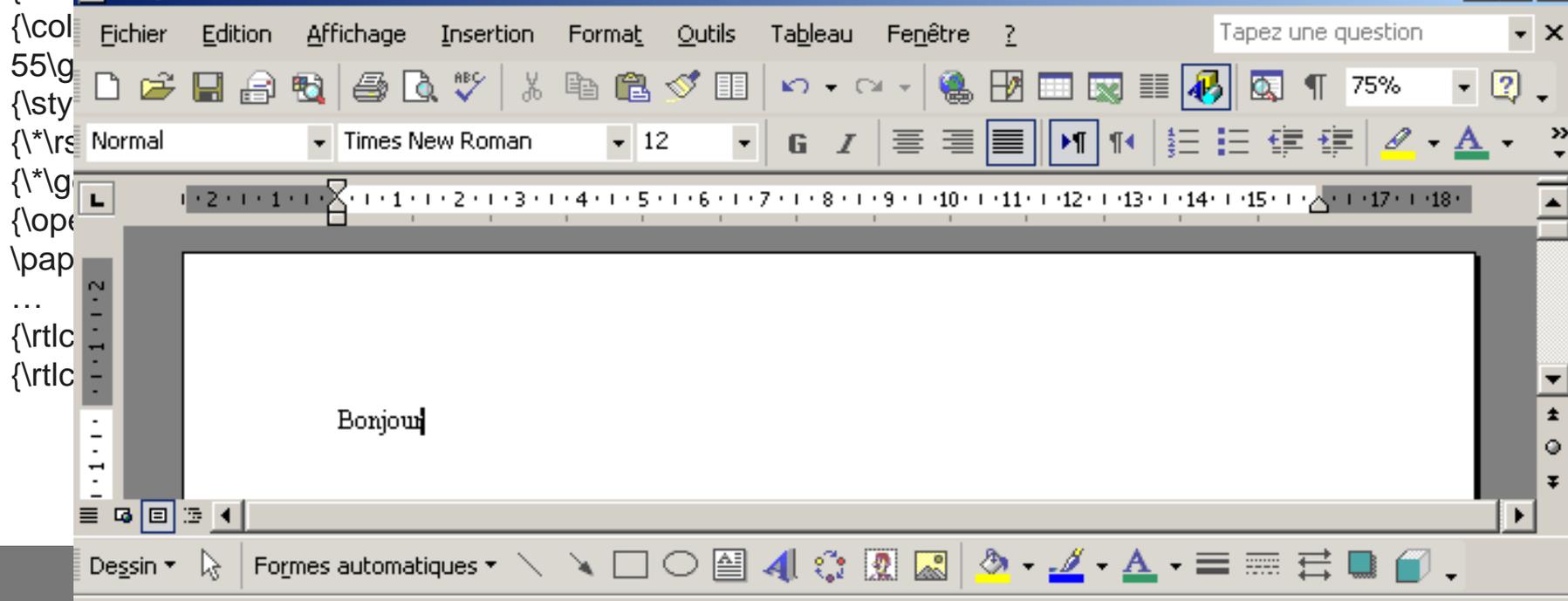
Introduction aux langages à balises

□ Le langage RTF

```
{\rtf1\adeflang1025\ansi\ansicpg1252\uc1\adef0\deff0\stshfdbch0\stshfloch0\stshfhich0\stshfbi0\deflang1036\deflangfe1036{\fonttbl{\f0\froman\fcharset0\prq2{\panose 02020603050405020304}Times New Roman;}{\f36\froman\fcharset238\prq2 Times New Roman CE;}{\f37\froman\fcharset204\prq2 Times New Roman Cyr;}}...
```

```
{\f43\froman\fcharset186\prq2 Times New Roman Baltic;}
```

```
{\f44
```



Les langages dédiés au Web

□ Historique 1/3

- 1960-1986 : SGML (norme ISO) et DTD
- 1989 : ODA (norme ISO, concurrent de SGML)
- Fin des années 80 : apparition/essor du web
- 1992-1997 : HTML (versions 1.0 -> 4.01)
- Octobre 1994 : création du World Wide Web Consortium (W3C) : <http://www.w3.org>
- Décembre 1996 : CSS Level 1 (fonctionnalités de base, rév. 01/99)
- Février 1998 : XML version 1.0
- Mai 1998 : CSS Level 2 (fonctionnalités supplémentaires, rév. 04/08)
- Octobre 1998 DOM Level 1 (supporte XML et HTML)
- Décembre 1999 : XHTML 1.0

Les langages dédiés au Web

□ Historique 2/3

- 1999 : RDF et RDF-Schema 1.0 (rév. 2004)
- Novembre 2000 : DOM Level 2 (support CSS & namespaces XML)
- Mai 2001 : schémas XML 1.0
- Mai 2001 : XHTML 1.1 (module-based)
- Juillet 2001 : SVG 1.0
- Novembre 2001 : XSL 1.0
- Janvier 2003 : SVG 1.1
- Février 2004 : XML 1.1 (rév. août 2006)
- Février 2004 : OWL 1.0
- Avril 2004 : DOM Level 3 Core
- Octobre 2004 : XML Schema (2^e édition)

Les langages dédiés au Web

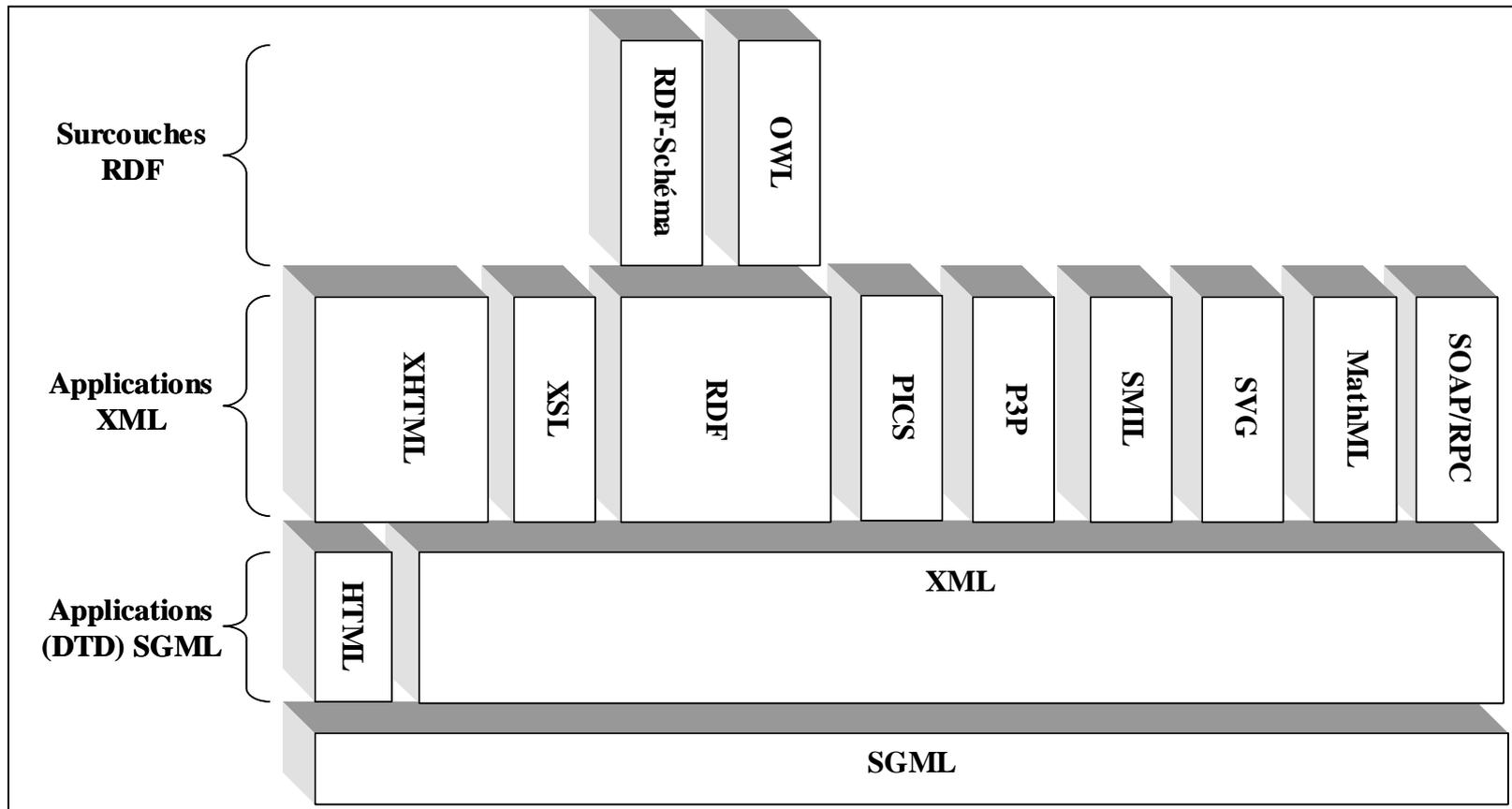
□ Historique 3/3

- Janvier 2007 : XQuery 1.0
- Janvier 2007 : XPath 2.0
- Janvier 2007 : XSLT 2.0
- Avril 2007 : SOAP 1.2
- Juin 2007 : WSDL 2.0
- Juillet 2008 : XHTML 1.1 Basic
- Octobre 2008 : XHTML Modularization 1.1
- Septembre 2009 : CSS level 2.1 (normalisation de fonctionnalités existantes)
- Septembre 2009 : HTML 5 (editor draft)

□ Restent en développement : CSS L3, DOM L3, XHTML 2.0...

Les langages dédiés au Web

□ Aperçu de quelques langages (source : W3C)

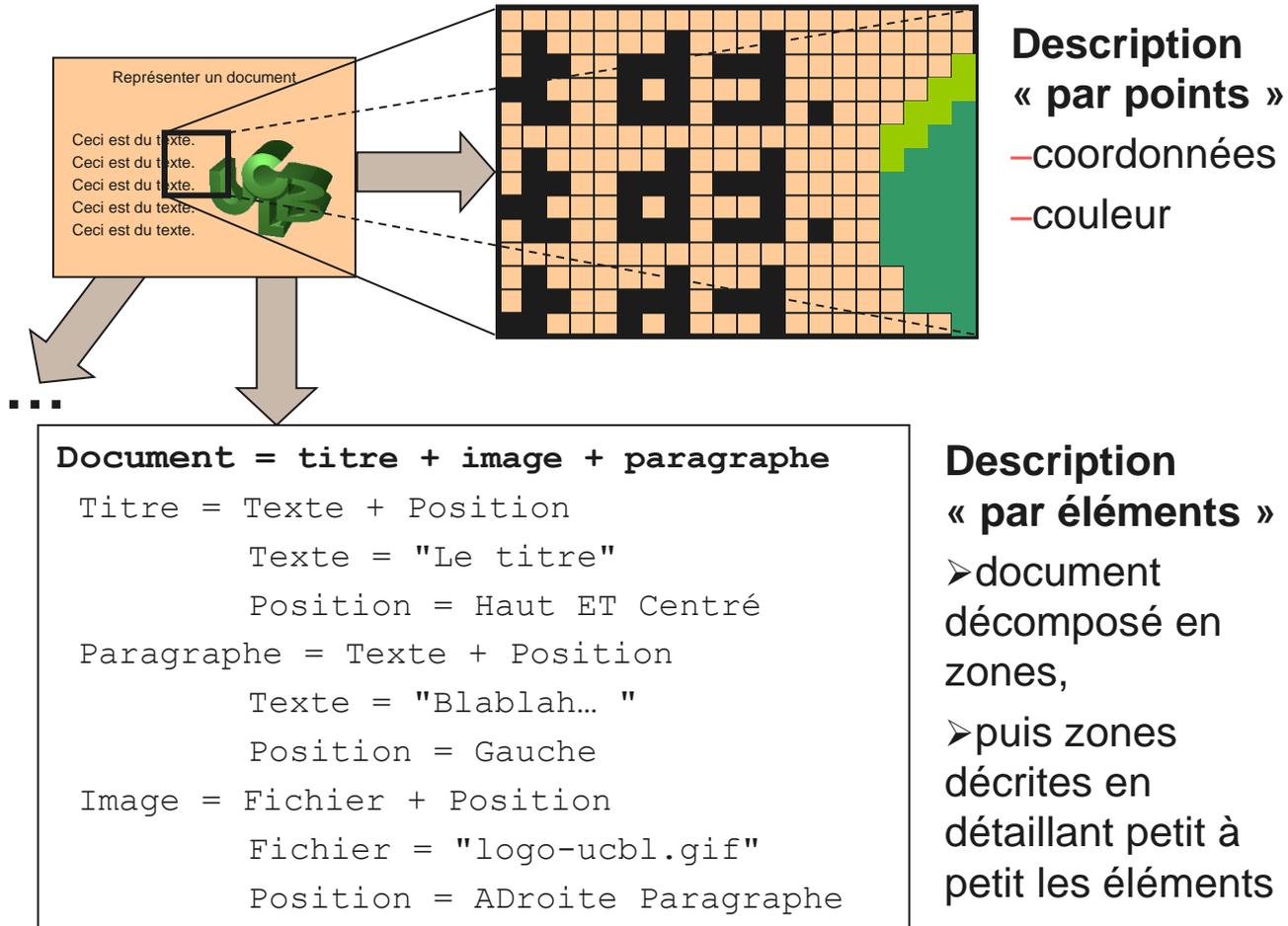


➤ Autre version : <http://kensall.com/big-picture/bigpix22.html>

Introduction aux langages à balises

- Plus tout ce que vous ne trouverez pas sur le site du W3C
 - EcmaScript, JavaScript
 - TCP/IP
 - Unicode
 - ...

Plusieurs façons de décrire un document



HTML (HyperText Markup Language)

□ Principe

- décrire des pages Web → programmation déclarative
- dans des fichiers textuels → encodage
- sous forme d'un ensemble d'éléments → arborescence
- en utilisant des balises → syntaxe, grammaire

HTML (HyperText Markup Language)

❑ Balise ouvrante

- avant le contenu de l'élément

`<title>`

❑ Balise fermante

- après le contenu de l'élément

`</title>`

❑ Balise vide (balise d'élément vide)

- pas de contenu

`<hr />`

❑ Nom de balise

- mot-clé simple spécifiant le type d'élément à délimiter

`title, img, p...`

❑ Attribut de balise

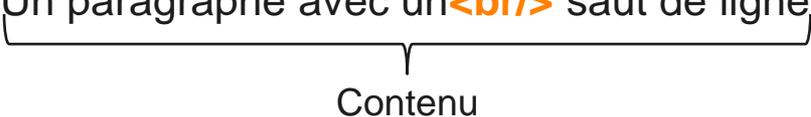
- « morceau d'information » permettant de préciser un élément

``

- Toujours à l'intérieur de la balise ouvrante (ou vide)
- Ne fait pas partie du contenu de l'élément
- Structure : `nom="valeur"`

HTML (HyperText Markup Language)

□ Contenu d'un élément

- texte et/ou autres éléments entre une balise ouvrante et une balise fermante
- Exemples
 - `<p>Un paragraphe...</p>`
 - `<p>Un paragraphe avec un mot en gras (bold).</p>`
 - `<p>Un paragraphe avec un
 saut de ligne</p>`


□ Un élément est décrit par

- soit une balise ouvrante, un contenu et une balise fermante...
- ...soit une balise vide
 - Exemples
 - ❖ `
`
 - ❖ ``

□ Un élément peut contenir d'autres éléments

- On parle de structure **arborescente**

HTML (HyperText Markup Language)

□ Entité

- Chaîne de caractères prédéfinie et fixe (constante)
- Possède un nom qui permet d'y faire référence

Caractères ISO Latin1			Noms ISO	Noms Unicode	Caractères
Noms ISO	Noms Unicode	Caractères			
Agrave	#192	À	agrave	#224	à
Aacute	#193	Á
Acirc	#194	Â	egrave	#232	è
Atilde	#195	Ã
Auml	#196	Ä	igrave	#236	ì
Aring	#197	Å
AElig	#198	Æ	ograve	#242	ò
Ccedil	#199	Ç
Egrave	#200	È	ugrave	#249	ù
Eacute	#201	É
...	yuml	#255	ÿ
...	1. Caractères de contrôle		
Igrave	#204	Ì	amp	#38	&
...	apos	#39	'
Ograve	#210	Ò	gt	#62	>
...	lt	#60	<
Ugrave	#217	Û	quot	#34	"
...		#91	[
Yacute	#221	Ý		#93]

- Utilisation : référence par son nom, précédé de '&' et terminé par ';'
 - Exemple : `<nom>Lionel Médini</nom>`

Exemple de structuration arborescente

Prologue →

<html>

Balise ouvrante →

<head>

<title>Merveilles du monde**</title>**

</head>

<body>

<p>Dans l'image ci-après

nous présentons un zèbre vivant dans la savane.

</p>

<!-- Fin zèbre début lion -->

Commentaire →

<p>Le lion n'est pas vraiment son ami.

Balise d'élément vide
(une seule balise) →

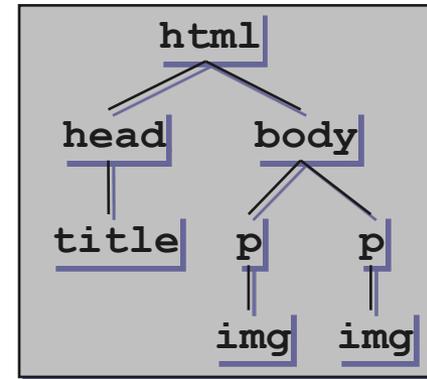
</p>

</body>

</html>

Balise fermante →

Attributs de l'élément Image



Arborescence

Structure générale d'un document XHTML

□ Prologue

- Déclaration XML
- Déclaration de DTD

□ Élément racine

- Élément `html`

□ En-tête

- Élément `head`
- dans `html`

□ Titre

- Élément `title`
- dans `head`

□ Corps

- Élément `body`
- dans `html`

□ Commentaires

- `<!-- ... -->`
- n'importe où

Exercice

- 1) Écrivez cette structure dans un fichier texte et sauvegardez ce fichier (→ « `structure.html` »)
- 2) Composez votre première page Web telle que :
 - le titre soit « `Ma première page Web` »
 - le corps de la page contienne « `Bonjour, le monde !` »
- 3) Enregistrez cette nouvelle page avec un autre nom
- 4) Ouvrez-la dans votre navigateur

XHTML Strict et Transitionnel

□ Transitionnel

- Les éléments de présentation de HTML sont encore autorisés
 - b, center, font, ...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

□ Strict

- Séparation stricte du document XHTML de sa présentation → utilisation de style CSS obligatoire

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

□ Déclaration d'espace de noms dans la balise ouvrante html

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
```

□ Pour des débutants : **XHTML strict + CSS**

En-tête : contenu de l'élément `head`

□ Titre du document

➤ `<title></title>`

□ Autre informations non affichées à l'écran, utilisées par le navigateur, les moteurs, etc.

➤ `<meta name="..." content="..." />`

- `<meta http-equiv="Refresh" content="4" ; URL=http://www.google.com" />`
- `<meta name="author" content="" />`
- `<meta name="Keywords" content="motcle1, motcle2, motcle3" />`
- `<meta name="language" content="fr" />`

➤ `<base href="URL-de-base" />`

- URL de base pour les URL relatives

□ Styles

➤ `<style />` → inclure une feuille de style CSS dans la page

➤ `<link />` → lier le document à une ressource externe (typiquement, feuille de style)

□ Scripts

➤ `<script />` → ajouter un script à la page

Corps

□ Élément body

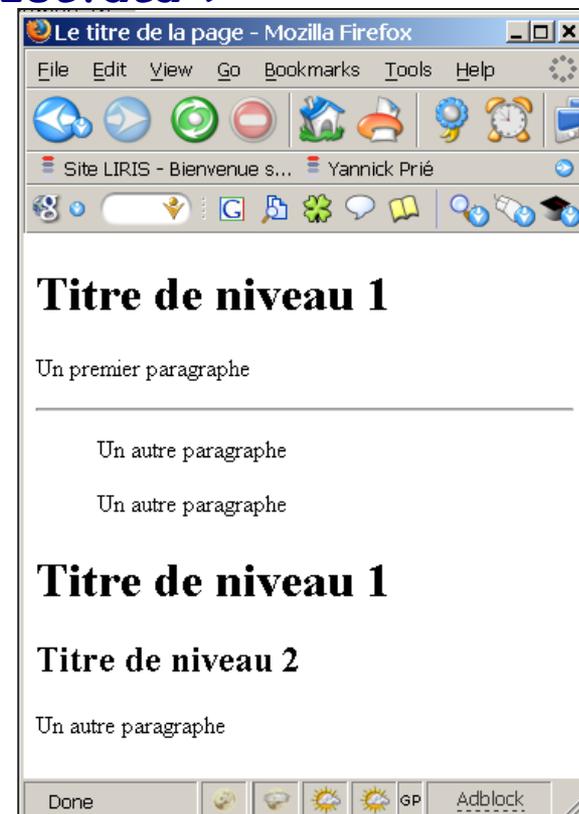
- Toutes les informations visualisables

□ Structuration

- `<p> ... </p>` → paragraphe
- `<h1> ... </h1>` → titre de niveau 1
- ...
- `<h6> ... </h6>` → titre de niveau 6
- `<hr />` → ligne horizontale
- `
` → saut de ligne

Corps : exemple

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
  <title>Le titre de la page</title>
</head>
<body>
  <h1>Titre de niveau 1</h1>
  <p>Un premier paragraphe</p>
  <hr /> <!-- Une ligne horizontale -->
  <blockquote><p>Un autre paragraphe</p>
  <p>Un autre paragraphe</p></blockquote>
  <h1>Titre de niveau 1</h1>
  <h2>Titre de niveau 2</h2>
  <p>Un autre paragraphe</p>
</body>
</html>
```



Mise en évidence

□ Emphase

➤ `` ... `` (emphasis)

□ Emphase forte

➤ `` ... ``

□ Exposants et indices

➤ `^{` ... `}` (superscript)

➤ `_{` ... `}` (subscripted)

Extraits, citations et références

❑ Citer quelque chose : `cite`

- ... `<cite>Discours de la méthode</cite>` de `<cite>Descartes</cite>` ...

❑ Citation courte : `q`

- Comme le disait `<cite>Ronsard</cite>` :
`<q>Mignonne allons voir si la rose...</q>`

❑ Citation longue : `blockquote`

- `<h5><cite>Barbara</cite> de <cite>Jacques Prévert</cite></h5>`
`<blockquote cite="http://wwwuser.gwdg.de/">`
`<p>Rappelle-toi Barbara</p>`
...
`<p>Dont il ne reste rien.</p>`
`</blockquote>`

Autres éléments de structuration

□ Texte spécial

- `<pre> ... </pre>` (preformatted text)
- `<code> ... </code>`
 - pour mettre du texte représentant du code informatique (par exemple du XML !)

□ Indications sur le contenu textuel

- Utilisation dans une feuille de style pour mettre en forme
 - **defn** (définition)
 - **acronym** (acronyme)
 - **abbr** (abréviation)
- Utilisation par des logiciels pour connaître la sémantique du contenu textuel (utilisé ?)
- Principe du web sémantique : généralisation, extériorisation des vocabulaires, *etc.*

Images

□ Principe

- Un élément image (img) est lié à un fichier image (attribut src)
- Une image est une ressource externe, désignée par une URL, qui doit être chargée par le navigateur

□ Exemple

- ```

```

## □ Images cliquables

- ```

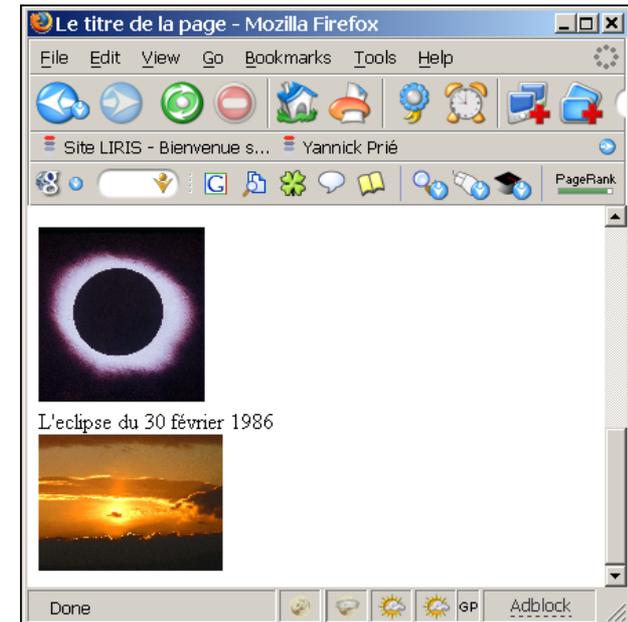
<map id="mymap">
  <area href="section1.html" alt="Route 20"
shape="rect" coords="0,0,49,49" />
  <area href="section2.html" alt="Route 35"
shape="rect" coords="0,49,49,99" />
</map>
```

Images : exemple

```
...  
  
<br />  
<img src ="eclispe.jpg"  
    alt ="L'eclipse du 30 février 1986"  
    width="150" /> <!-- erreur ! -->  
<br />  
<img src ="http://web.ccr.jussieu.fr/cim2/SAXaussois/soleil.JPG"  
    alt ="Soleil couchant"  
    height="100"/>  
...
```

- Remarque importante

- il faut trois ressources pour afficher cette page
 - Fichier source HTML
 - Fichier `eclipse.jpg` (chemin local : dans le même dossier que le fichier source HTML)
 - Fichier `soleil.JPG` (disponible sur la machine `web.ccr.jussieu.fr`, en suivant le chemin `/cim2/SAXaussois/soleil.JPG`)
- trois requêtes HTTP en tout pour l'affichage



Listes

❑ Principe général

- Un élément liste contient des élément items

❑ Listes classiques

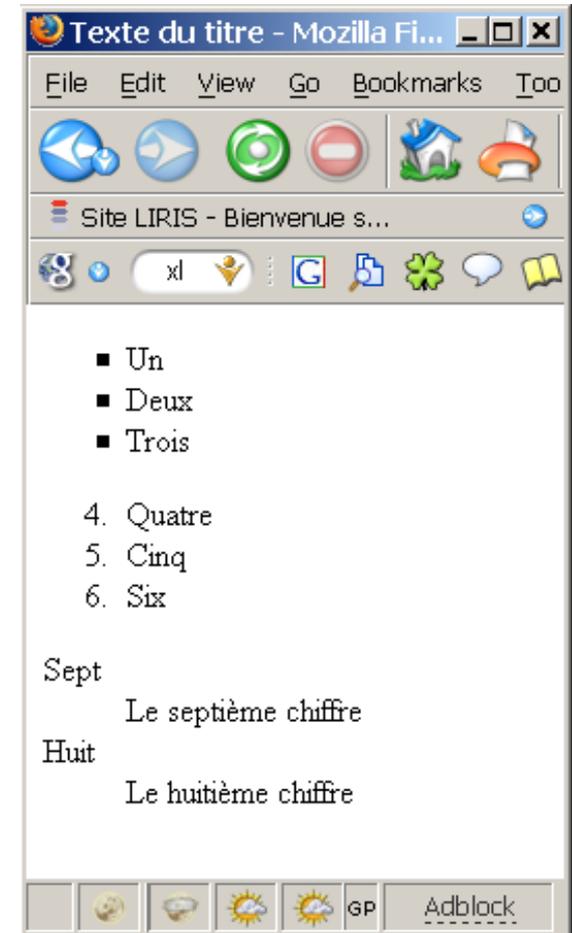
- Liste à puces
 - `<u1> ... </u1>` (unordered list)
- Liste ordonnée
 - `<o1> ... </o1>` (ordered list)
- Item de list
 - ` ... `

❑ Liste de définitions

- Conteneur
 - `<d1> ... </d1>`
- Terme de définition
 - `<dt> ... </dt>`
- Description de définition (= définition elle-même)
 - `<dd> ... </dd>`

Listes : exemple

```
...  
<ul type="square">  
  <li>Un</li>  
  <li>Deux</li>  
  <li>Trois</li>  
</ul>  
<ol start="4">  
  <li>Quatre</li>  
  <li>Cinq</li>  
  <li>Six</li>  
</ol>  
<dl>  
  <dt>Sept</dt>  
  <dd>Le septième chiffre</dd>  
  <dt>Huit</dt>  
  <dd>Le huitième chiffre</dd>  
</dl>  
...
```



Liens hypertextes

□ Principe

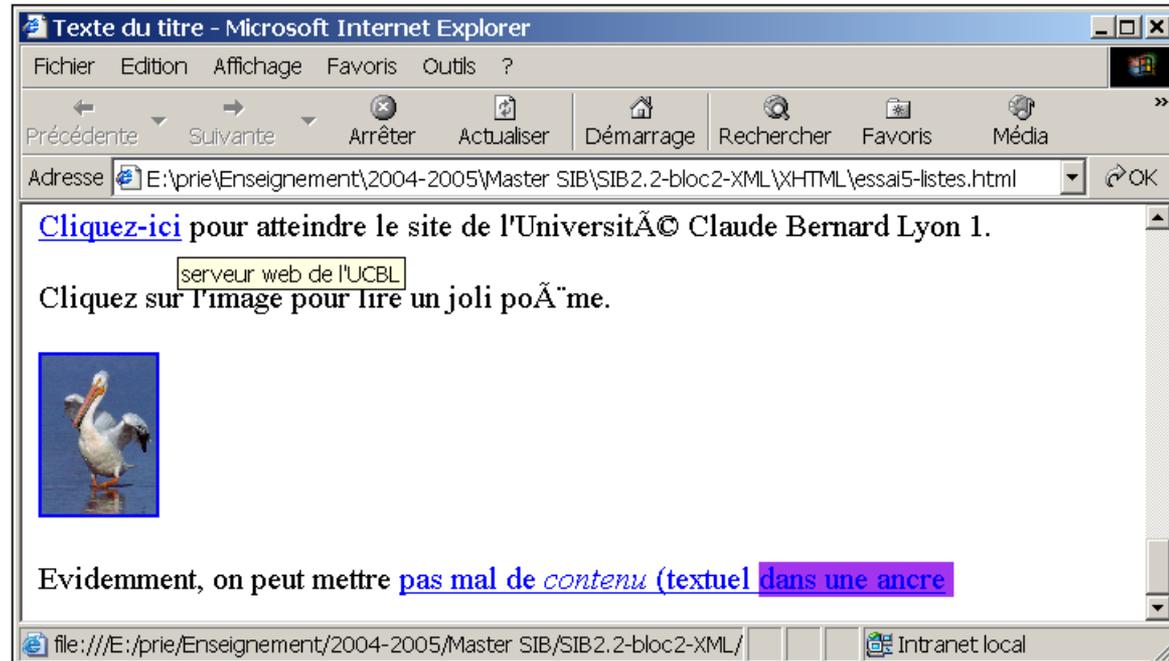
- une ancre contenant le texte ou les éléments sur lesquels on peut cliquer
- le clic redirige vers une URL
 - entraîne le chargement de la ressource désignée par l'URL

□ Syntaxe

- ``
`ancre`
``
- **href** : URL de destination du lien
- **title** : texte qui apparaîtra dans une info-bulle si on survole le lien

Liens hypertextes : exemple

```
<p>Quelques exemples de liens</p>
<p><a href="URL" title="serveur
web de l'UCBL">Cliquez-
ici</a> pour atteindre le
site de l'Université Claude
Bernard Lyon 1.</p>
<!-- Remarque : on a ici un lien
absolu (URL complète) -->
<p>Cliquez sur l'image pour lire
un joli poème.</p>
<a href="poeme.html"></a>
<!-- Remarque : on a ici un lien
relatif -->
<p>Evidemment, on peut mettre <a
href="http://www.w3.org/TR/xh
tml1/">pas mal de
<em>contenu</em> (textuel
<span style="background-
color: #A234EE">
éléments)</span> dans une
anc</a> </p>
```



Tableaux

□ Principe

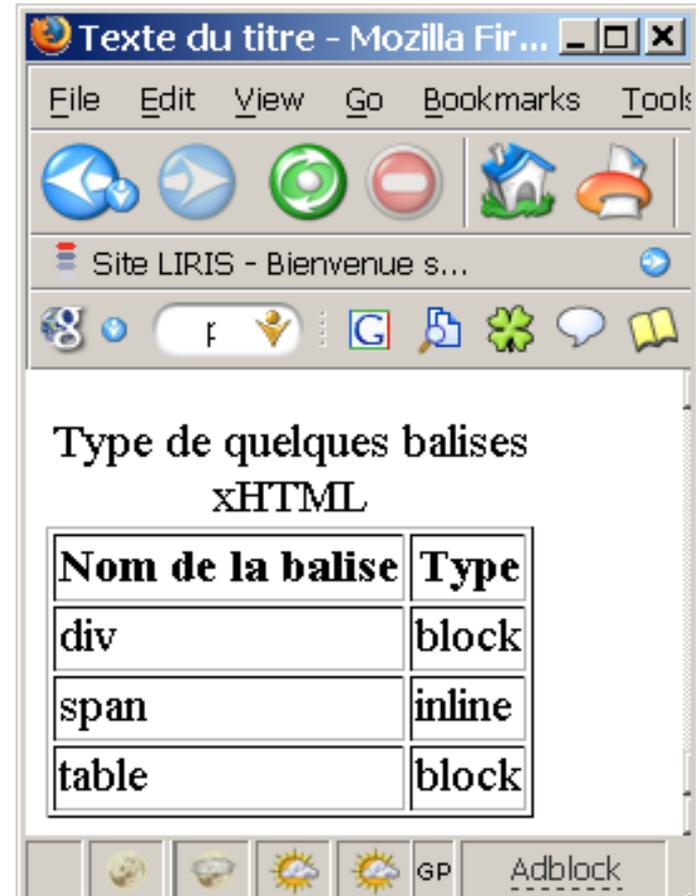
- Un tableau contient des lignes, lesquelles contiennent des cellules

□ Éléments

- `<table> ... </table>` (élément général)
- `<tr> ... </tr>` (table row)
- `<td> ... </td>` (table cell)
- `<th> ... </th>` (table header)
- `<caption> ... </caption>` (table caption)
- ...

Tableaux : exemple

```
<table summary="Ce tableau présente des balises xHTML et leur type." border="1">  
<caption>Type de quelques balises xHTML</caption>  
<thead>  
<tr>  
  <th scope="col">Nom de la balise</th>  
  <th scope="col">Type</th>  
</tr>  
</thead>  
<tbody>  
<tr>  
  <td>div</td>  
  <td>block</td>  
</tr>  
<tr>  
  <td>span</td>  
  <td>inline</td>  
</tr>  
<tr>  
  <td>table</td>  
  <td>block</td>  
</tr>  
</tbody>  
</table>
```



Deux types d'éléments en HTML

□ Éléments *block*

- Définissent des blocs physiques (des « boîtes ») dans le document
- Induisent un saut de ligne à la fin de leurs contenus
- Exemples
 - p, h1-h6, div, ul, ol, li, pre, hr, table, etc.

□ Éléments *inline*

- Dans le flux des caractères
- « Semblables » à des caractères
- Exemple
 - em, img, strong, a, etc.

Les éléments span et div

□ Utilité

- Regrouper des sous-parties de document
 - Pour leur donner un nom
 - Pour contenir du texte ou d'autres éléments XHTML
 - Pour les traiter globalement en leur affectant des styles, des événements...
- Mais sans leur donner de sémantique particulière

□ ` ... `

- Élément de type *inline*
- Contient du texte ou d'autres éléments *inline*

□ `<div> ... </div>`

- Élément de type *block*
- Contient du texte, d'autres éléments *inline* ou *block*
- Retour chariot à la fin

div / span : exemple

```
<h1>Exemples avec <code>div</code> et  
<code>span</code></h1>  
<div id="premiere_strophe" style="color: purple ;  
font-style: italic">  
<p>Le Capitaine <span class="heros">  
Jonathan</span>,</p>  
<p>Etant âgé de dix-huit ans</p>  
<p>Capture un jour un <span style="font-weight:bold">  
pélican</span></p>  
<p>Dans une île d'Extrême-orient.</p>  
</div><div style="text-align: right">  
<p>Le pélican de <span class="heros">  
Jonathan</span></p>  
<p>Au matin, pond un oeuf tout blanc</p>  
<p>Et il en sort un <span style="font-weight:bold">  
pélican</span></p>  
<p>Lui ressemblant étonnamment.</p>  
</div><div style="text-align: justified">  
<p>Et ce deuxième <span style="font-weight:bold">  
pélican</span></p>  
<p>Pond, à son tour, un oeuf tout blanc</p>  
<p>D'où sort, inévitablement</p>  
<p>Un autre, qui en fait autant.</p>  
</div><div style="text-align: right">  
<p>Cela peut durer <span style="text-align:  
right">pendant</span> très longtemps</p>  
<p>Si l'on ne fait pas d'omelette avant.</p></div>
```

Exemples avec div et span

Le Capitaine Jonathan,

Etant âgé de dix-huit ans

Capture un jour un pélican

Dans une île d'Extrême-orient.

Le pélican de Jonathan

Au matin, pond un oeuf tout blanc

Et il en sort un pélican

Lui ressemblant étonnamment.

Et ce deuxième pélican

Pond, à son tour, un oeuf tout blanc

D'où sort, inévitablement

Un autre, qui en fait autant.

Cela peut durer très longtemps

Si l'on ne fait pas d'omelette avant.

Formulaires

□ Objectif

- Permettre à l'utilisateur d'envoyer des informations à un serveur
- Saisir les informations
 - listes déroulantes, cases à cocher, zones de texte, etc.
 - bouton pour remettre à zéro
- Envoyer les informations
 - méthode GET
 - ❖ les informations passent par l'URL
 - ❖ Ex. : `http://www.google.com/search?sourceid=mozclient&ie=utf-8&oe=utf-8&q=html+4.01`
 - Méthode POST
 - ❖ les informations sont échangées par le protocole HTTP

□ Éléments

- `<form> ... </form>` → définit un formulaire
- `<input> ... </input>` → entrée de formulaire
- `<textarea> ...</textarea>` → zone de texte
- `<select> ... </select>` → liste déroulante
- `<option> ... </option>` → option du select

Formulaires : exemple

```
<form method="post" action="http://serveur.com/script.php">

<p><input type="text" name="Champ_saisie" value="Texte"
/></p>

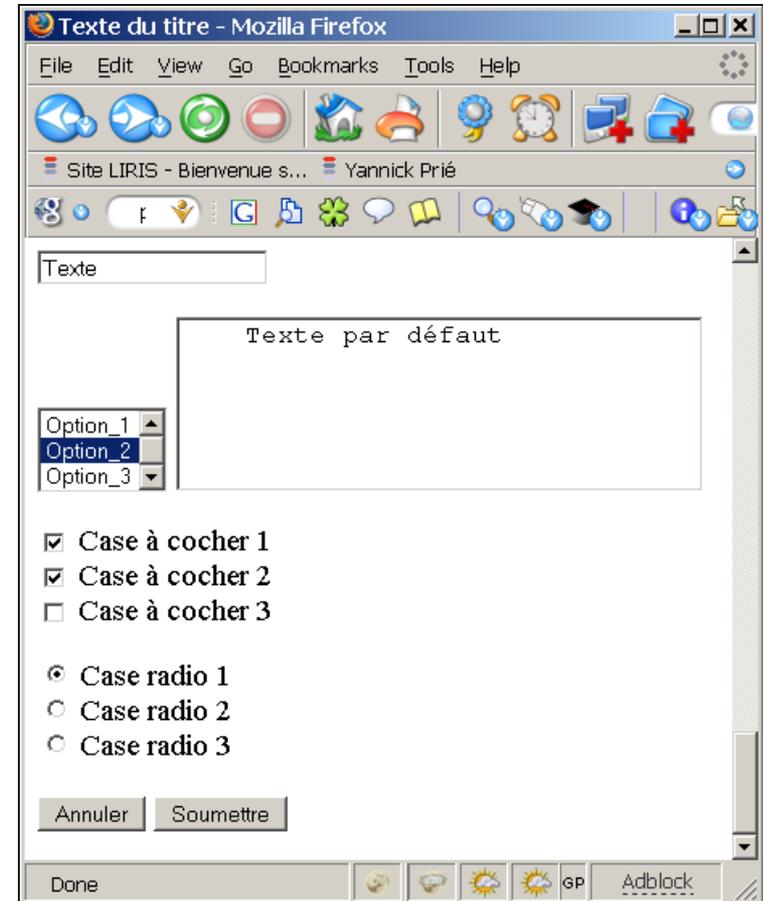
<p><select name="Liste_Choix" size="3">
  <option value="Option_1">Option_1</option>
  <option value="Option_2">Option_2</option>
  <option value="Option_3">Option_3</option>
</select>

<textarea name="Zone_Texte" cols="30" rows="5">
  Texte par défaut </textarea></p>

<p><input type="checkbox" name="case1" value="Case_1">
  Case à cocher 1<br />
<input type="checkbox" name="case2" value="Case_2">
  Case à cocher 2<br />
<input type="checkbox" name="case3" value="Case_3">
  Case à cocher 3<br /></p>

<p><input type="radio" name="Case_Radio"
  value="Case radio 1">Case radio 1<br />
<input type="radio" name="Case_Radio"
  value="Case radio 2">Case radio 2<br />
  <input type="radio" name="Case_Radio"
  value="Case radio 3">Case radio 3<br /></p>

<input type="reset" name="Annulation" value="Annuler">
<input type="submit" name="Soumission" value="Soumettre">
</form>
```



Cadres (frames)

□ Principe

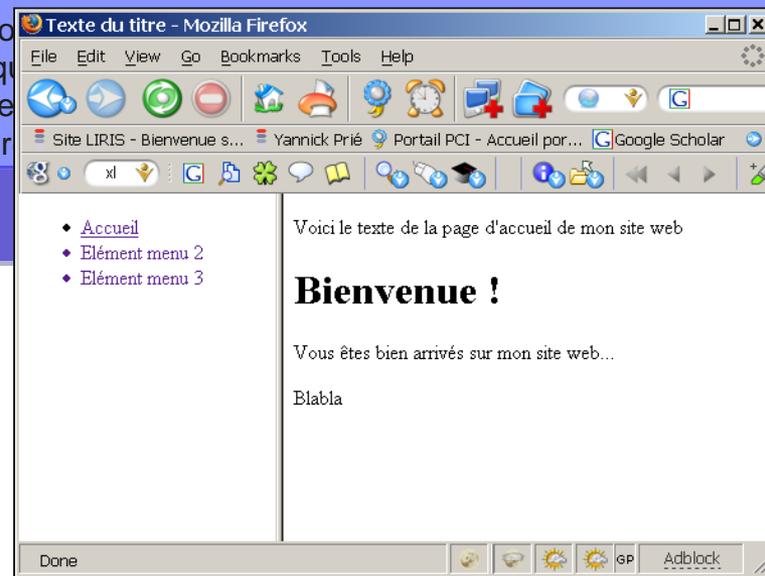
- diviser la fenêtre d'affichage en frames
- afficher dans chaque frame une page HTML
- Nombre de fichiers concernés
 - 1 pour la déclaration du frameset + 1 pour chaque frame
 - Ex : 3 frames → minimum 4 fichiers

□ En XHTML

- Le fichier principal obéit à la DTD frameset
- Les autres sont en XHTML

□ Remarque

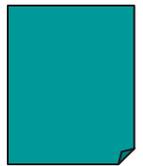
- Ne pas utiliser car pose beaucoup de problèmes
 - gestion
 - indexation



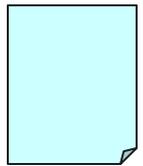
Cadres : exemple

Le fichier principal fait appel à menu.html et accueil.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title>Texte du titre</title>
</head>
<frameset cols="200,*">
  <frame src="menu.html" name="cadre1" scrolling="no">
  <frame src="accueil.html" name="cadre2"
scrolling="yes">
</frameset>
<p>Ce projet utilise des cadres. Chez vous les
cadres ne sont pas affichés.</p>
</frameset>
</html>
```



menu.html



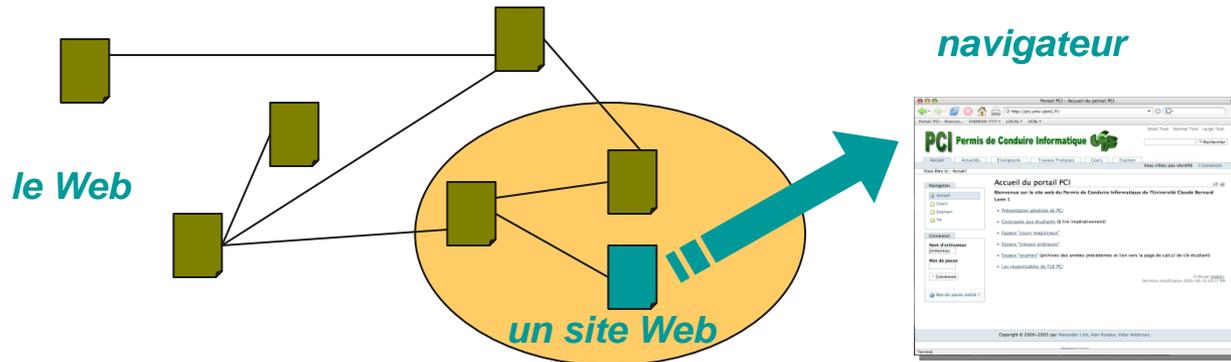
accueil.html

Conclusion sur XHTML

- ❑ Possibilité de valider ses documents
 - <http://validator.w3.org/>
- ❑ Ce cours n'est pas une description complète des éléments et attributs de XHTML
 - Aller voir la référence
- ❑ Non abordés ici
 - Les scripts
 - Javascript
 - `<script> ... </script>`
 - Les objets pris en charge par des plugins
 - Applets JAVA, animations Flash, audio, vidéo, etc.
 - `<object> ... </object>`
 - Les sites web et leur gestion (conception, installation sur serveurs web...)

1. Introduction
2. Aspects techniques
3. Langage HTML
4. Conclusion

World Wide Web : une description plus formelle



- ❑ Ensemble de pages Web formant un **réseau** (ou une toile : *web* en anglais) sur lequel il est possible de naviguer
- ❑ WWW = World Wide Web
- ❑ **Navigateur**
 - outil permettant d'afficher des pages HTML et de passer d'une page à l'autre au moyen de liens hypertextes → navigation