

WORLD WIDE WEB

Introduction / langage HTML

Lionel Médini

Licence professionnelle DIST



Objectifs de cette UE

- Découvrir ce qu'est le Web et comment ça marche
- Connaître (et pratiquer) les différentes technologies de base
- Découvrir quelques outils pour être rapidement opérationnels

Planning de l'UE

- Présentation du WWW
 - Historique
 - Fonctionnement
- Technologies côté client
 - HTML
 - CSS
- Frameworks côté serveur
 - CMS : WordPress
- Transformation de documents
 - XPath
 - XSLT
- Contrôle continu terminal

Modalités d'interaction entre les cours

- Un forum sur Spiral pour poser des questions
- Réponses au début du cours suivant

Modalités d'évaluation

- MCC communes aux 3 cours du module DIST 1
 - CCI : pas de 2ème session, note individuelle
 - Notes :
 - Rendus des TPs
 - CCF

Objectifs du cours et des TP

- Découvrir ce qu'est le Web et comment ça marche
 - Internet
 - URL
 - HTTP
 - XHTML
- Être capable de créer une page Web mise en forme
 - Introduction au langage HTML
 - Introduction au langage CSS

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

Rappel : Internet

- Un réseau de réseaux interconnectés (d'où le nom)
- Un ensemble de matériels, logiciels et protocoles (notamment IP)
- Un ensemble de **services**
 - **Application** qui utilise un **protocole** et un numéro de **port**
 - e-mail, transfert de fichiers, connexion à distance, WWW...
- Une somme « d'inventions » qui s'accumulent
 - Mécanismes réseau de base (TCP/IP)
 - Nommage et adressage des ressources (DNS, URL)
 - Outils et protocoles spécialisés
 - Langages d'échange d'informations standardisés (HTML, XML...)

Bref historique d'Internet

- 1959-1968 : Programme ARPA
 - le ministère américain de la défense lance un réseau capable de supporter les conséquences d'un conflit nucléaire
 - 1969 : ARPANET, l'ancêtre d'Internet
 - les universités américaines s'équipent de gros ordinateurs et se connectent au réseau ARPANET
 - 1970-1982 : Ouverture sur le monde
 - premières connexions avec la Norvège et Londres
 - 1983 : Naissance d'Internet
 - protocole TCP/IP : tous les réseaux s'interconnectent
 - les militaires quittent le navire
 - 1986 : Les autoroutes de l'Information
 - la National Science Fondation déploie des super-ordinateurs pour augmenter le débit d'Internet
 - 1987-1992 : Les années d'expansion
 - les fournisseurs d'accès apparaissent
 - les entreprises privées se connectent au réseau
 - 1993-2003 : L'explosion d'Internet
 - ouverture au grand public
 - avènement du WEB et du courrier électronique
- } marché considérable

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 : 1^{er} site Web fonctionnel (mode texte)
 - 1^{er} serveur Web : nxoc01.cern.ch
 - 1^{er} navigateur Web : WorlWideWeb (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
- 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



Aspects techniques du Web

- 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

Aspects techniques du Web

- Les forces en présence
 - Côté client : l'utilisateur utilise un navigateur
 - Client HTTP
 - Interprète les pages Web et les affiche à l'utilisateur
 - Peut effectuer des traitements plus complexes (plugins, moteur de scripts...)
 - Côté serveur : le Webmaster gère un serveur Web
 - Attend les requêtes HTTP et y répond
 - En renvoyant des ressources dont il dispose
 - En interrogeant plusieurs modules (sécurité, scripting, redirection...)
 - En interrogeant d'autres outils pour les traitements complexes
 - On parle aussi de serveur Web pour désigner la machine qui héberge le programme serveur (abus de langage)

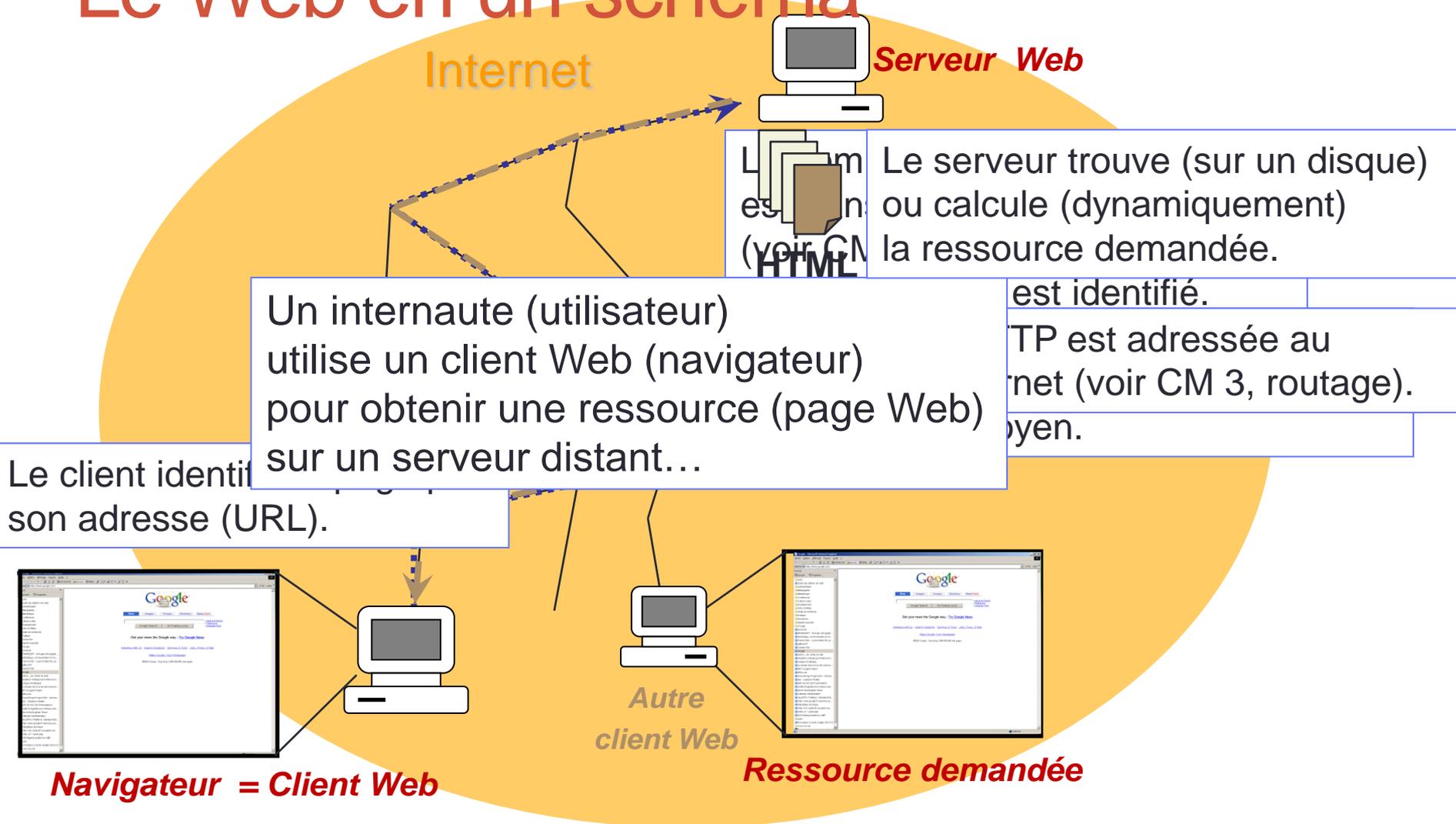
Le Web : première définition

- World Wide Web (WWW, Toile)
- Définition
 - ensemble de données disponibles sur les serveurs accessibles sur le réseau Internet
 - pouvant être visualisées et/ou utilisées avec un navigateur Web
- Attention : Web \neq Internet
 - Internet = support de communication général
 - Web = une partie des contenus circulant sur l'Internet

- 1. Introduction
- 2. Aspects techniques
- 3. Page Web
- 4. Conclusion

- 1. Objectifs du cours
- 2. Principes généraux

Le Web en un schéma



URL (Uniform Resource Locator)

- Permet d'identifier une **ressource** sur le réseau, c'est-à-dire :
 - une page Web
 - une image (seule ou utilisée dans une page Web)
 - un programme
 - un fichier à télécharger...
- Indique
 - un **protocole** (langage de communication entre deux programmes sur deux machines).
 - Exemple: FTP (File Transfert Protocol), HTTP (HyperText Transfert Protocol)...
 - une adresse et un chemin
 - forme générale : `protocole://adresse`
 - exemple : `http://www.univ-lyon1.fr/`

Différents types d'URL

- Forme principale (protocole *HTTP*)

http://pci.univ-lyon1.fr/TP/sujets-TP.pdf

Protocole

Adresse machine

Chemin fichier

- Forme pour désigner les fichiers *locaux*

- Chemin relatif : fichier.html ou dossier/toto.html
- Sur un disque : file://C:/chemin/fichier.htm

- Forme pour le *transfert* de fichiers

- **ftp://ftp.inria.fr/INRIA/tech-reports/RR-5645.pdf**

- Forme pour l'envoi de *courrier* électronique

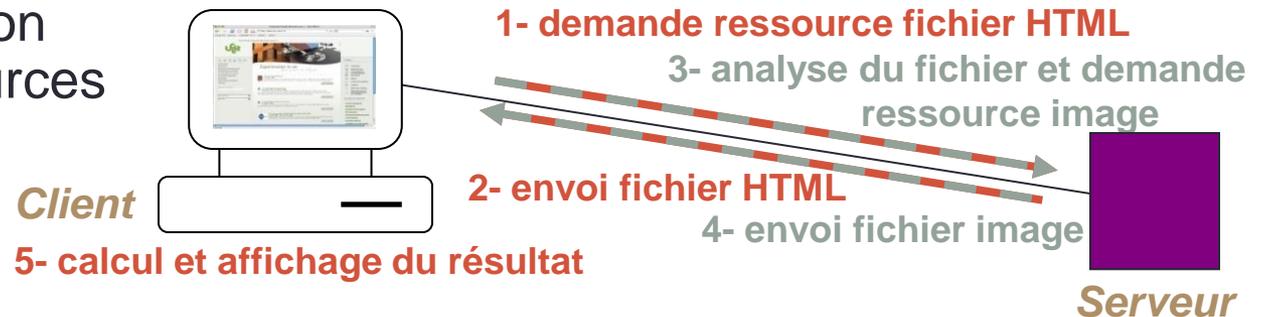
- **mailto:jean.durand@univ-lyon1.fr**

DEMO...

HTTP (HyperText Transfer Protocol)

- Rappels du CM 3
 - Protocole de type client-serveur
 - Protocole de niveau applicatif
- Mécanisme de requête / réponse HTTP
 - **Requête** du client : URL de la ressource
 - **Réponse** du serveur : code de statut / d'erreur + ressource (si « OK »)
 - tout s'est bien passé : « OK » (code de statut : 200)
 - pas le droit d'accéder : « forbidden » (code d'erreur : 403)
 - la ressource n'existe pas : « not found » : (code d'erreur : 404)

- Exemple de transaction avec plusieurs ressources



- Remarque : le protocole **HTTPS** (sécurisé) fonctionne de la même façon

1. Introduction
2. Aspects techniques
3. Page Web
4. Conclusion

1. Ressource et URL
2. Protocole HTTP
3. Serveur Web
4. Client Web

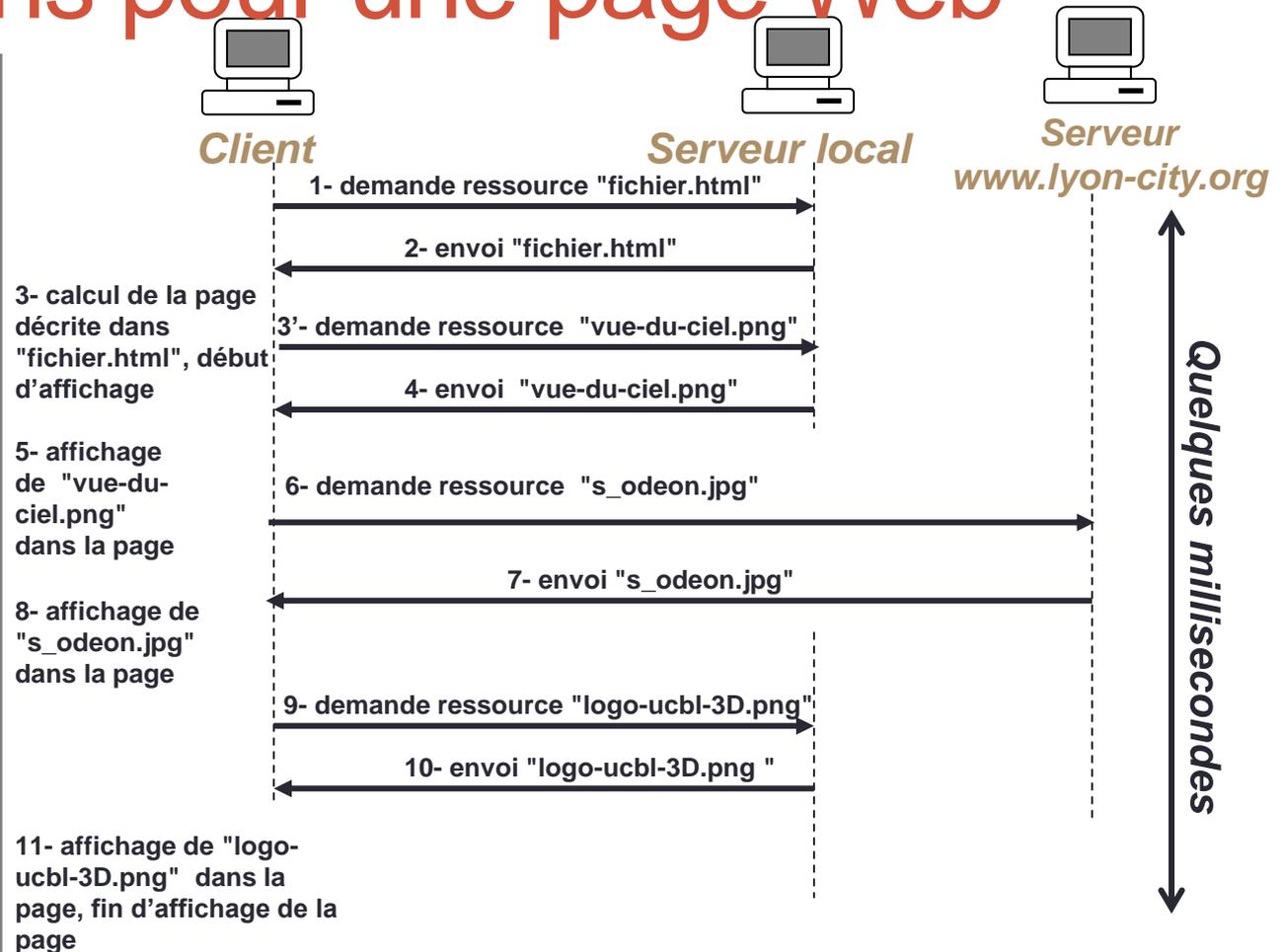
Transactions pour une page Web

fichier.html

```
<html>
<head>
<title>INSERTION IMAGES</title>
</head>
<body bgcolor="#ffffff">
<p>Voici une photo aérienne de Lyon
(chemin relatif) : <img src= "vue-
du-ciel.png" alt="vue du ciel"
/>.</p>

<p>Voici une autre photo (URL
distante) : .</p>

<p>Cliquer sur le logo <a
href="http://www.univ-
lyon1.fr/"></a> pour visiter le site
Web de Lyon 1.</p>
</body>
</html>
```



Vitesse d'une transaction HTTP

- Temps d'affichage d'une page Web lié
 - à la complexité de la page à afficher
 - nombre et volume des ressources composant la page
 - traitements éventuellement nécessaires pour générer dynamiquement ces ressources
 - au débit et à l'encombrement des réseaux parcourus
 - à la possibilité de présenter une page avant que toutes les données soient arrivées
(ex. texte avant images, chargement asynchrone...)
 - à la charge du serveur
 - réponse à de nombreux clients en même temps
 - cryptage ou non des données (HTTP / HTTPS)
 - autres applications tournant sur la même machine
 - à la vitesse de la machine cliente

Serveur Web

□ Définition

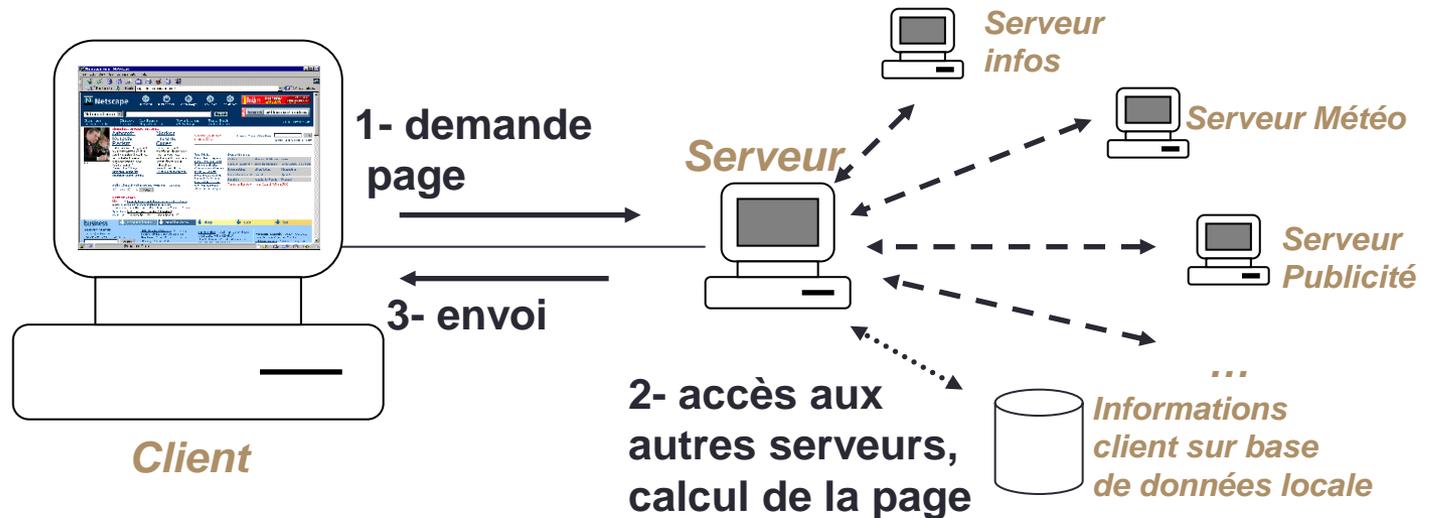
- **Application** dont le rôle est de fournir des ressources Web aux clients qui les demandent
 - par le protocole HTTP(S)
 - sur Internet ou sur un réseau local
- Par extension, **machine** sur laquelle s'exécute cette application

□ Deux façons de fournir des ressources

- statique : à partir de fichiers existants sur le serveur
- dynamique : en générant la ressource au moment de la requête

Pages générées dynamiquement

- Le serveur doit calculer la ressource avant de la renvoyer
 - 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 ou bases de données) : le serveur considéré est le **client** d'autres serveurs
 - avec un langage de programmation côté serveur (PHP, ASP, Java, Python...)
- Le mécanisme de requête / réponse reste le même
- Exemple



1. Introduction
2. Aspects techniques
3. Page Web
4. Conclusion

1. Ressource et URL
2. Protocole HTTP
3. Serveur Web
4. Client Web

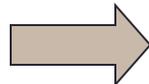
Navigateur Web (browser, butineur)

- Caractéristiques
 - logiciel client pour les protocoles HTTP et HTTPS
 - demande des fichiers (X)HTML à un serveur
 - interprète ces fichiers
 - demande éventuellement d'autres ressources
 - les présente à l'utilisateur
 - peut aussi réaliser quelques traitements (scripts)
- Principaux navigateurs
 - Mozilla Firefox, Chrome, Safari, Internet Explorer, Opera

Fichier HTML

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

Description du document



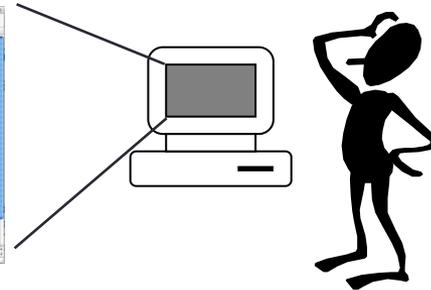
Calcul

Navigateur



Présentation à l'utilisateur

Utilisateur



Différents médias accessibles sur le Web

- Formes du point de vue de l'utilisateur
 - texte
 - image (photo, dessin, graphique)
 - son (musique, voix)
 - vidéo (images animées + son)
- Formats de fichiers correspondants
 - texte simple : ASCII (.txt)
 - images : JPEG (.jpg), GIF (.gif)
 - sons : WAVE (.wav), MPEG3 (.mp3)
 - vidéo : MPEG (.mpg), AVI (.avi), Quicktime (.qt)...
- Applications permettant de « lire » les fichiers (ouvrir les documents)
 - éditeur de texte
 - visualisateur d'images
 - lecteur de son, de vidéo

➔ Question : comment un navigateur sait-il quoi faire d'un type d'une ressource récupérée sur un serveur ?

Document multimédia

- Combinaison de différents médias
 - texte, image, son, vidéo...
- Deux caractéristiques principales
 - « **ça bouge** » : le document se déroule temporellement (vidéo, son, objets animés)
 - « **c'est interactif** » : on peut cliquer sur un bouton, suivre un lien hypermédia, remplir une zone texte...
- Exemples d'utilisations
 - cours, jeux interactifs, logiciels éducatifs, encyclopédies interactives, bornes dans les lieux publics, pages Web, films DVD...

Page Web

- Une page Web, c'est
 - un document multimédia
 - images, textes, possibilité d'interaction, liens
 - décrit élément par élément
 - titre, morceaux de texte, images...
 - avec un langage de description
 - HTML (1992), XHTML (2002) ou HTML5 (?)
 - 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)

Portail Web

- Principe

- proposer un ensemble de **services** pour une **communauté** sur un site Web
- exemples de communauté
 - générale (Yahoo!)
 - liée à un fournisseur d'accès Internet (Infonie, Voilà)
 - thématique (santé, enfants)...
- exemples de services : moteur de recherche, annuaires, météo, informations, jeux, publicité, accès courrier électronique, forums, discussion...

- Modèle économique

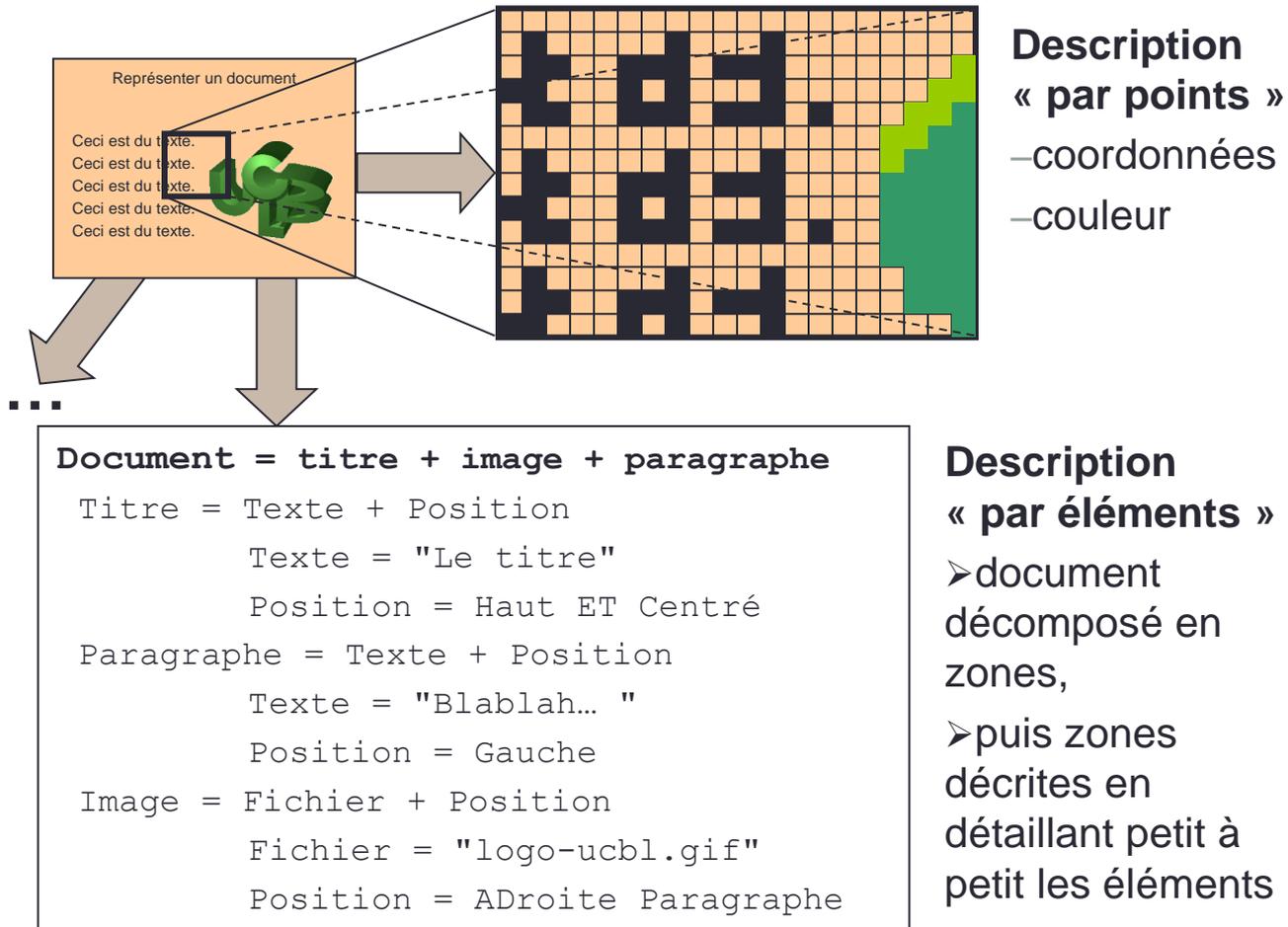
- fournir un nombre de services suffisants pour que le portail devienne l'accès principal au Web de certains utilisateurs
- proposer des publicités ciblées

- Nécessite une appréciation critique pour ne pas rester bloqué dans la sous-partie du Web accessible depuis le portail

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...

Plusieurs façons de décrire un document



Document comme composition d'éléments

- Un **élément**
 - est une partie significative d'un document
 - TITRE, AUTEUR, PARAGRAPHE, IMAGE, LIEN, TABLEAU...
 - peut être composé d'autres éléments
 - un DOCUMENT est composé d'un TITRE et d'une suite de PARAGRAPHE
 - un TABLEAU est composé de LIGNES
 - une LIGNE est composé de CASEs...
 - peut avoir des attributs
 - un attribut "nom du fichier" pour un élément IMAGE
 - un attribut "URL de destination" pour un élément LIEN
- Pour les pages Web
 - \approx 40 éléments de description dans les langages (X)HTML
 - (PCI : quelques éléments de base, syntaxe XHTML)

HTML (HyperText Markup Language)

- HTML permet de décrire des pages Web dans des fichiers textuels (ASCII) en utilisant des balises
- Balises
 - mots-clés simples pour délimiter des descriptions d'éléments (**title**, **img**, **p**...)
 - balises ouvrantes (**<title>**), fermantes (**</title>**), vides (**<hr />**)
 - balises avec **attributs** (****)
- Un élément
 - est soit délimité par deux balises...
 - entre lesquelles se situe son contenu : **<p>Un paragraphe...</p>**
 - dans ce cas, il peut contenir d'autres éléments (et ainsi de suite)
<p>Un paragraphe avec un mot en gras (bold).</p>
(= l'élément **p** contient l'élément **b**)
 - ...soit entièrement décrit par une balise vide : **
**
 - avec éventuellement des attributs : ****

Exemple de structuration en HTML

Déclaration de type

de document →

`<!doctype html>`

Balise ouvrante →

`<html>`

`<head>`

`<title>Merveilles du monde</title>`

`</head>`

Élément avec contenu

`<body>`

`<p>` Dans l'image ci-après

``
 nous présentons un zèbre vivant
 dans la savane.

Attributs de l'élément Image

`</p>`

Balise d'élément vide (une seule balise) →

`<p>`

``

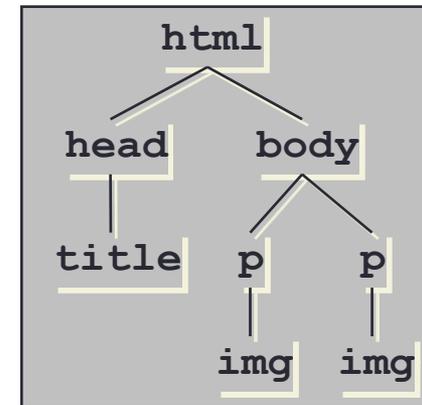
Le lion n'est pas vraiment son ami

`</p>`

Balise fermante →

`</body>`

`</html>`



Arborescence

XHTML

- XHTML 1.0 (1999)
 - reformulation XML de HTML4
 - définit trois DTD pour exprimer des documents HTML
 - XHTML-1.0-Strict
 - XHTML-1.0-Transitional
 - XHTML-1.0-Frameset
 - sémantique des balises
 - Définie dans HTML4
- XHTML 1.1
 - Modularisation
 - faciliter le mixage de fragments XML dans XHTML
- XHTML 2
 - Modulaire
 - Nouvelles fonctionnalités
 - En cours de discussion
- HTML le retour...
 - HTML 5 : W3C Working Draft (19 octobre 2010)

HTML → XHTML : à savoir

- De la rigueur...
 - Nom d'éléments en minuscules
 - `<P>` → `<p>`
 - Fermetures correctes : toute balise ouverte doit être fermée
 - `<p> ... </p>`
 - Éléments vides fermés :
 - `<hr />`
 - Un attribut associé à une valeur, entre guillemets
 - `` → ``
 - Attribut id pour identifier un élément (vs name avant)
 - `<h1 id="partie1" name="partie1"> ... </h1>`

Structure générale d'un document XHTML

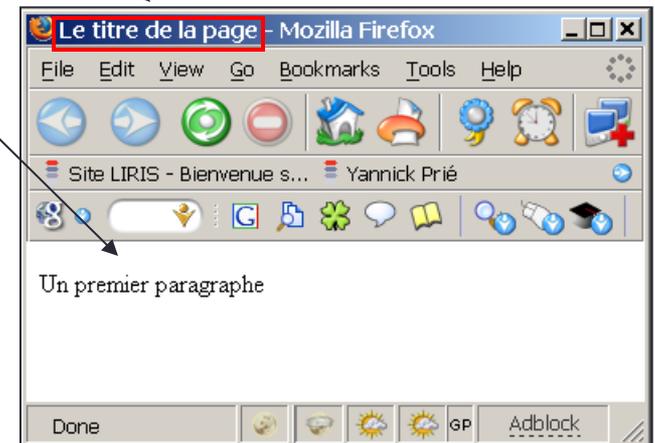
- Prologue
 - Déclaration XML
 - Déclaration de DTD
- En-tête
 - Élément **head**
- Corps
 - Élément **body**
- Commentaires n'importe où
 - `<!-- ... -->`

Premier 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>
  <!-- un commentaire -->
  <p>Un premier
  paragraphe</p>
</body>
</html>
```

Remarque :

ici le navigateur utilise une feuille de style par défaut pour afficher du XHTML strict non lié à une feuille de style CSS.



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 nom 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

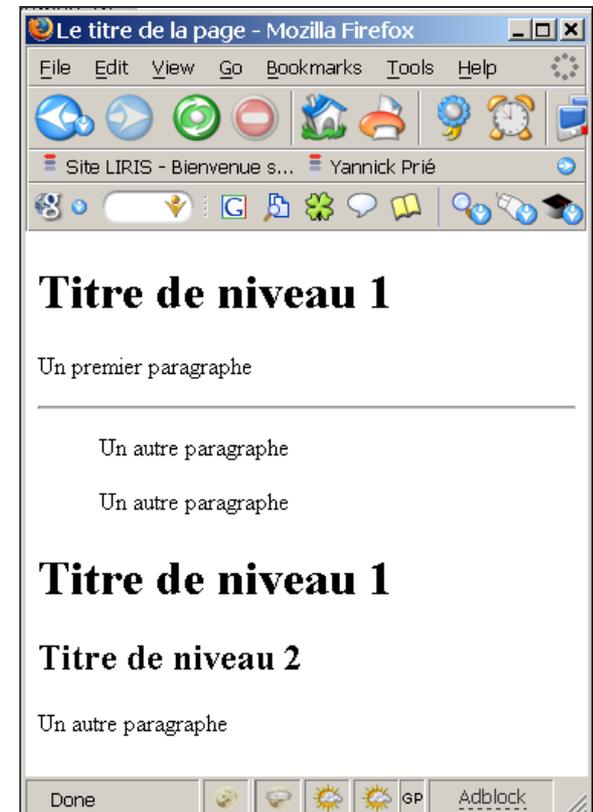
- Titre du document
 - `<title></title>`
- Autre informations non affichées à l'écran, utilisées par le navigateur, les moteurs, etc.
 - `<meta name="..." content="..." />`
 - `<meta http-equiv="Refresh" content="4" ; URL=http://www.google.com" />`
 - `<meta name="author" content="" />`
 - `<meta name="Keywords" 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>`

(<http://fusa2.free.fr/Webmaster/xhtml.htm>)

Autres balises de structuration

- Texte spécial
 - `<pre> ... </pre>` (preformatted text)
 - `<code> ... </code>`
 - pour mettre du texte représente 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** (abbréviation)
 - 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 />
```

```
 <!-- erreur ! -->
```

```
<br />
```

```

```

```
...
```

• 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

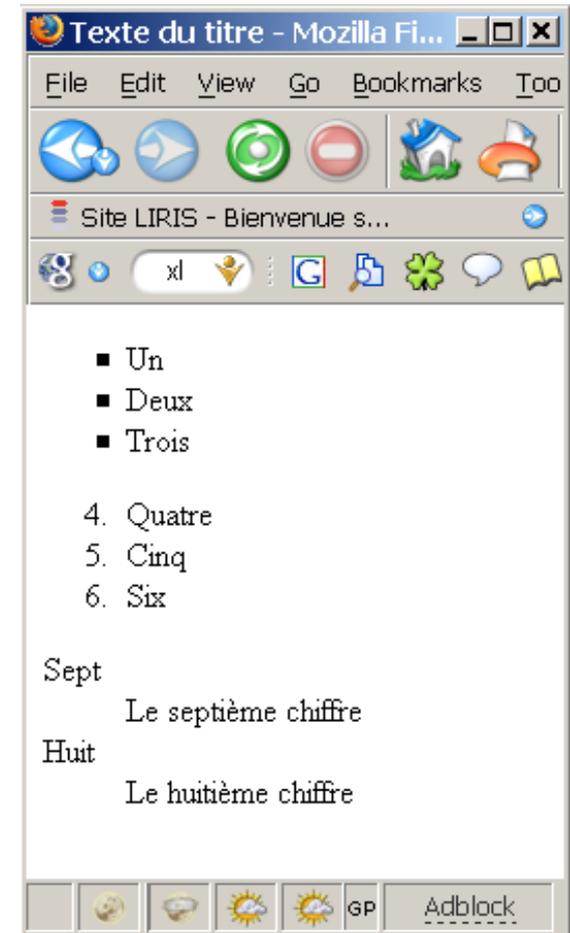


Listes

- Principe général
 - Un élément liste contient des élément items
- Listes classiques
 - Liste à puces
 - ` ... ` (unordered list)
 - Liste ordonnée
 - ` ... ` (ordered list)
 - Item de list
 - ` ... `
- Liste de définitions
 - Conteneur
 - `<dl> ... </dl>`
 - 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

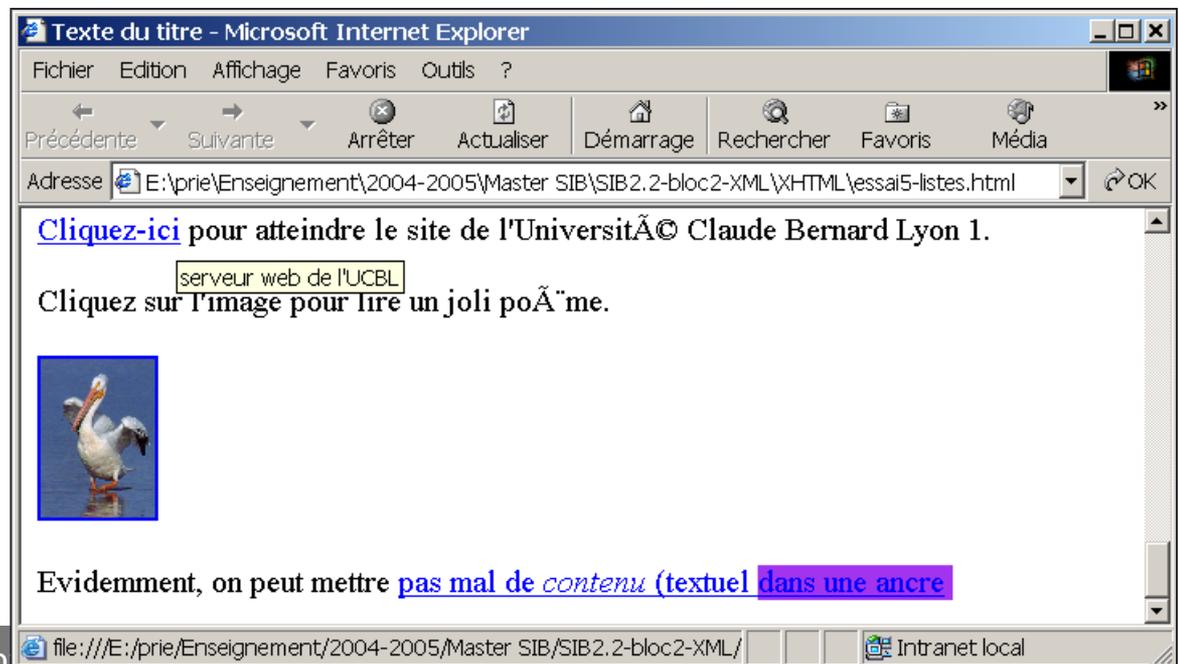
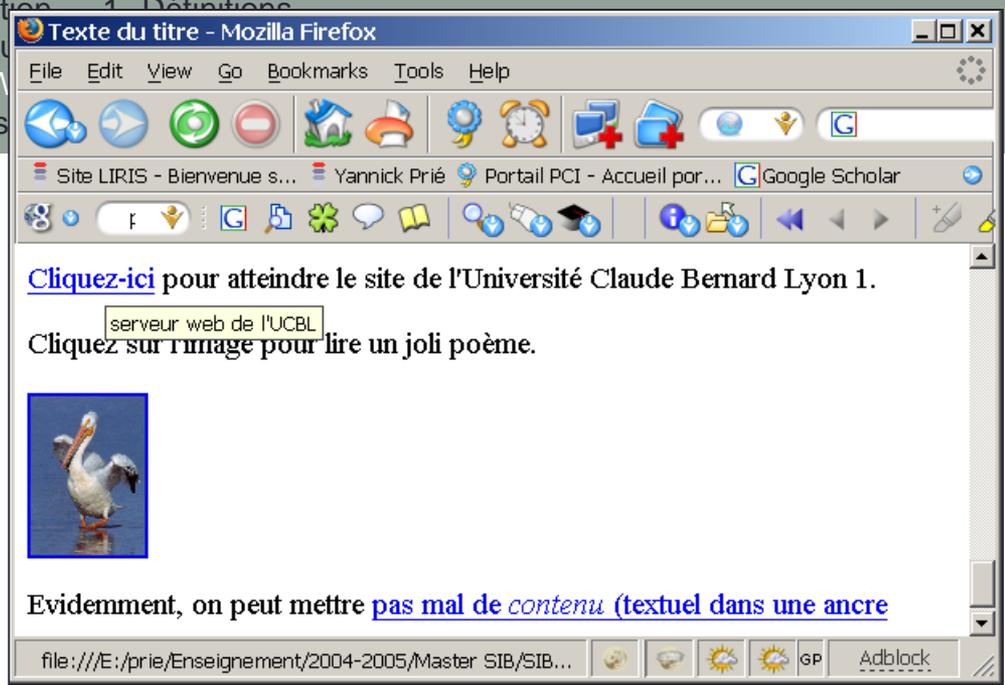
1. Introduction
2. Aspects techniques
3. Page Web
4. Conclusion

Liens hypertextes : exemples

```

<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/xhtml1/">pas mal de <em>contenu</em> (textuel <span style="background-color: #A234EE" éléments)</span> dans une ancre</a> </p>

```



Tableaux

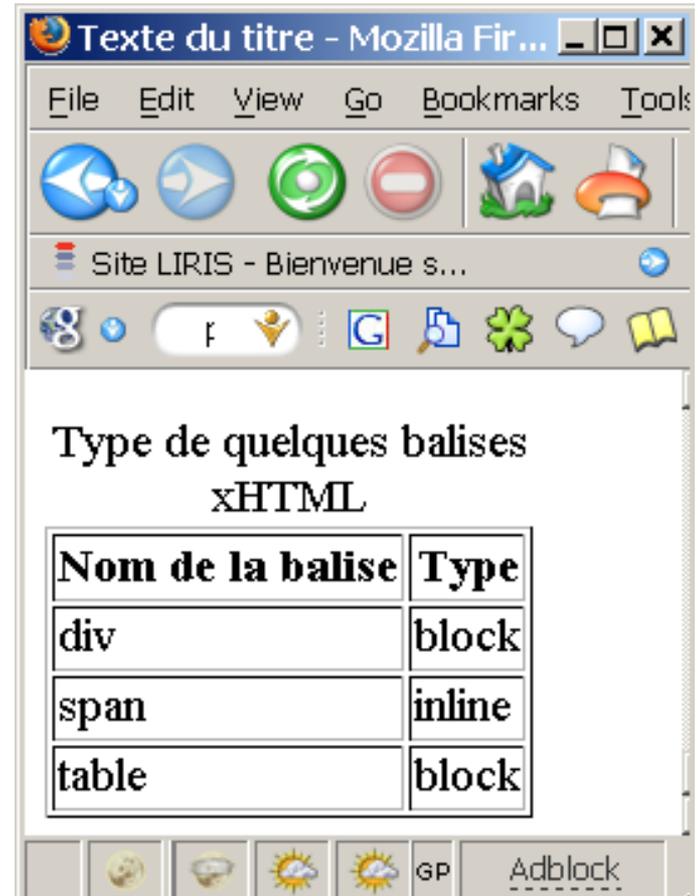
- Principe
 - Un tableau contient des lignes, lesquelles contiennent des cellules
- Éléments
 - `<table> ... </table>` (élément général)
 - `<tr> ... </tr>` (table row)
 - `<td> ... </td>` (table cell)
 - `<th> ... </th>` (**table header**)
 - `<caption> ... </caption>` (table caption)
 - ...

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.

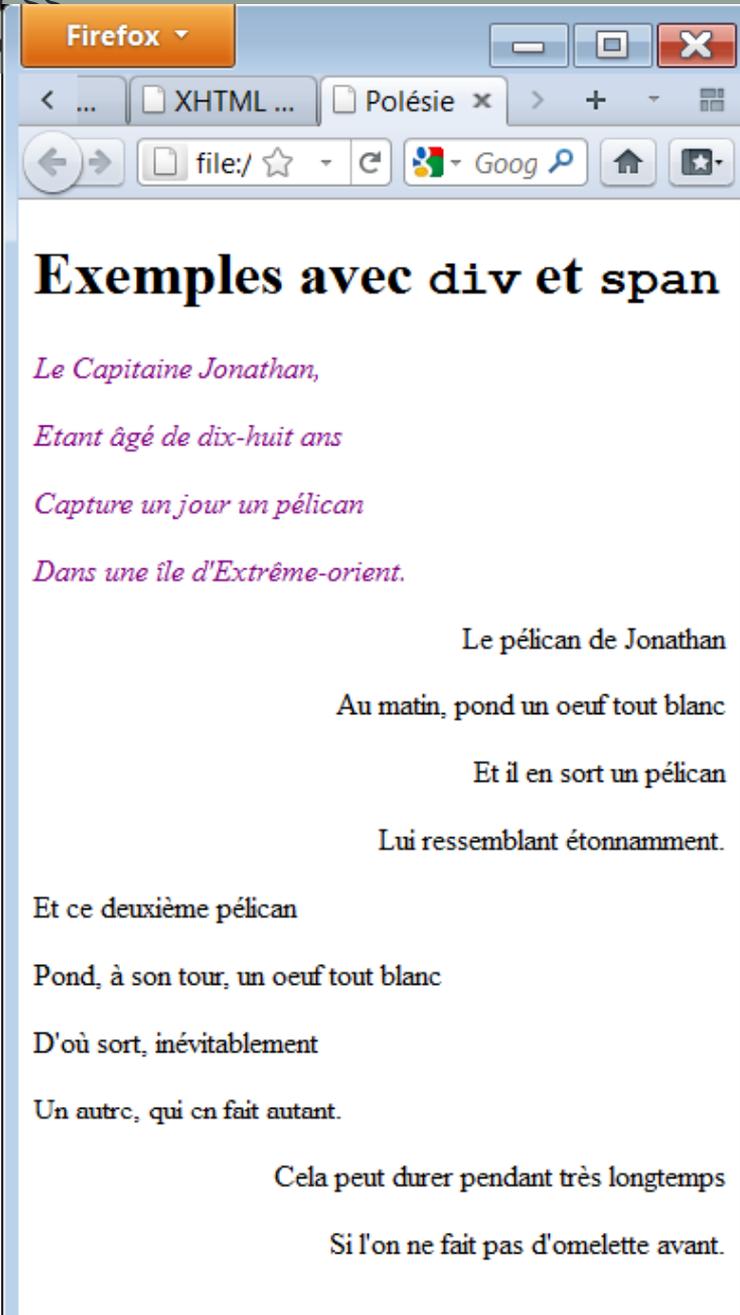
Les éléments span et div

- Utilité
 - Regrouper des sous-parties de document
 - Pour leur donner un nom
 - Pour enclore (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

1. Introduction
2. Aspects techniques
3. Page Web
4. Conclusion
1. Définitions
2. Langage HTML
3. Langage CSS
4. Publication

div / span : exemple (Desnos)

```
<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-
weigh: 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-
weigh: bold"> pélican</span></p>
<p>Lui ressemblant étonnamment.</p>
</div><div style="text-align: justified">
<p>Et ce deuxième <span style="font-weigh: 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>
```



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
- Eléments
 - `<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>
```

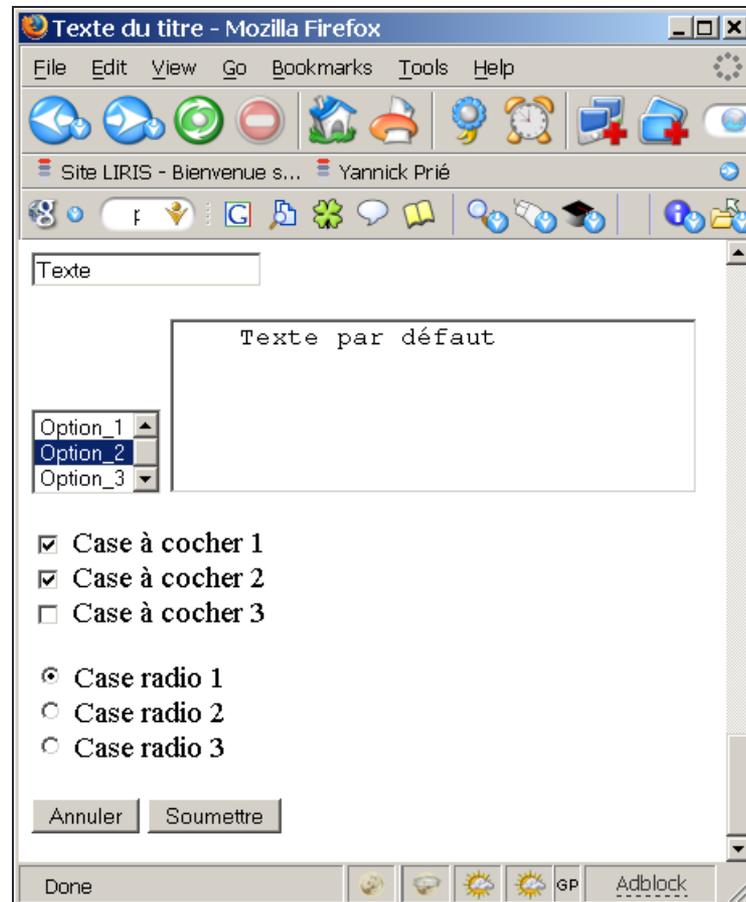
```
<p><select name="Liste_Choix" size="3">
  <option value="Option_1">Option_1</option>
  <option value="Option_2">Option_2</option>
  <option value="Option_3">Option_3</option>
</select>
```

```
<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>
```



Conclusion sur XHTML

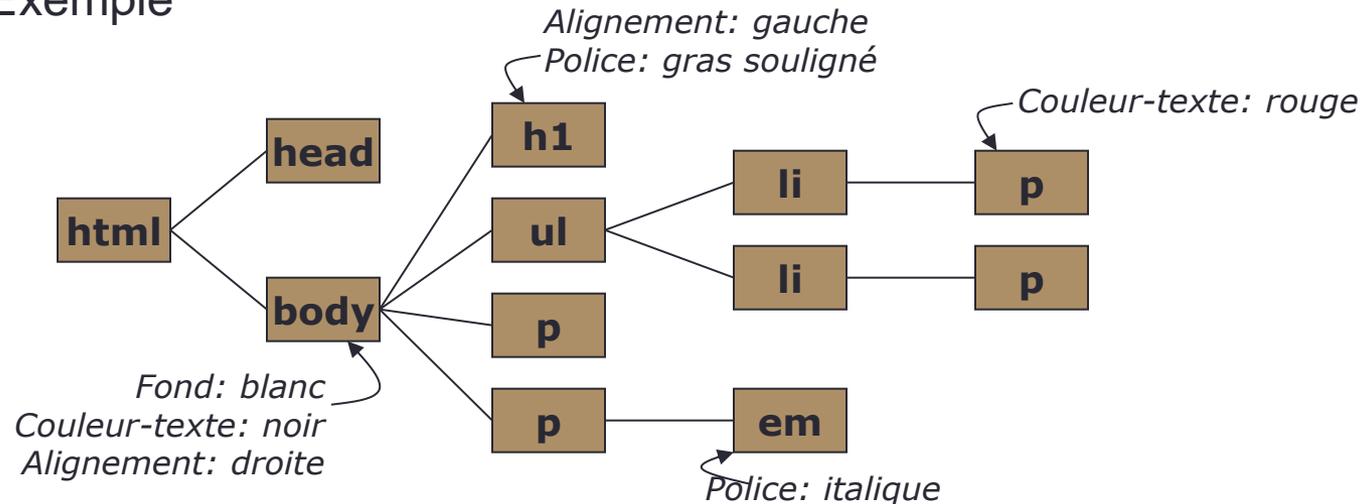
- Possibilité de valider ses documents
 - <http://validator.w3.org/>
- Ce cours n'est pas une description complète des éléments et attributs de XHTML
 - Aller voir la référence
- Non abordés ici
 - Les scripts
 - Javascript
 - `<script> ... </script>`
 - Les objets pris en charge par des plugins
 - Applets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

Règle à respecter en (X)HTML

- Deux types d'éléments
 - Blocs
 - Peuvent être positionnés sur l'écran
 - Peuvent contenir d'autres éléments de bloc ou des élément en-ligne
 - L'élément body ne peut contenir que des balises de bloc
 - Exemples : h1-h6, p, div...
 - En-ligne
 - Suivent le flot documentaire
 - Ne peuvent contenir que des élément en-ligne
 - Exemples : a, span, img, object, input...
- ➔ On ne peut pas directement positionner une image sur l'écran ; il faut la mettre dans un div.

CSS : généralités

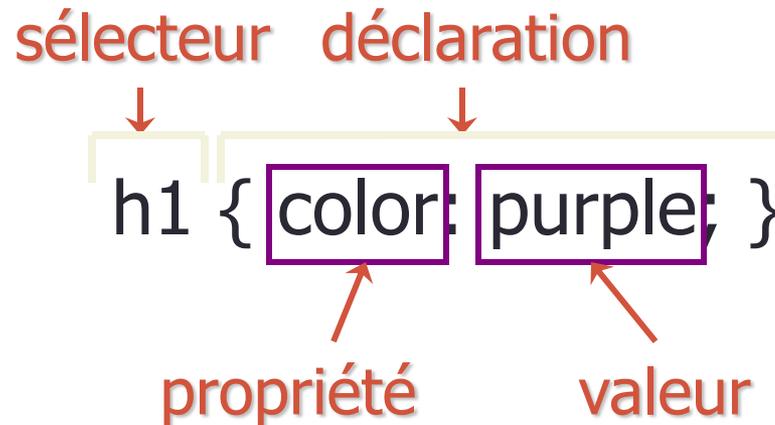
- Objectif
 - Décrire comment un document XHTML doit être affiché
 - Remplace les éléments d'affichage des anciennes versions de HTML (séparation réelle du contenu et de sa présentation)
 - L'affichage est pris en charge par le navigateur (normalement)
- Principe
 - Décoration de l'arbre des éléments XHTML
 - Associer un certain nombre d'attributs de style à un élément
- Exemple



Feuille de style CSS

- Une feuille de style est composée d'un certain nombre de règles (*rules*)
- Une règle se compose
 - d'un sélecteur
 - d'une déclaration
- Une déclaration se compose d'un ensemble de propriétés/valeurs
- Remarque
 - L'ordre des règles a une importance
 - Tous les styles ne peuvent pas s'appliquer à tous les éléments
 - Ex. : une image n'a pas de style de police

CSS : structure des règles



```
h1 { font-family: Arial, sans-serif; font-style: italic }
```

séparateur de propriétés/valeurs

Exemple de règle CSS

```
body {  
    background: #FFFFFF;  
    color: black; /* commentaire */  
    margin-left: 5%;  
    margin-right: 5%;  
    font-family: Tahoma, Optima,  
    Arial, sans-serif;  
}
```

Types de sélecteurs

- Simples et groupes
- Classes
- Pseudo-classes
- Pseudo-éléments
- Contextuels

Sélecteurs CSS : simple et groupe

- Simple
 - Lié à un type d'élément HTML
 - Utilisation de son nom
 - Exemple
 - `h1 { text-align: center; }`
- Groupe
 - Regroupement de règles qui s'appliquent à plusieurs éléments
 - Exemple
 - `h2, p { font-family: Optima, Arial, sans-serif; }`
- Impossibilité de considérer différemment des éléments de même type

Sélecteurs CSS : classes

- On peut assigner une classe à un élément HTML

```
<h1 class="header" >
```

- Celle-ci spécifie un sélecteur particulier dans la feuille de style

```
h1.header { text-align: center; }
```

- Une classe peut s'appliquer à de multiples éléments

```
.header { text-align: center; }
```

s'appliquera aussi à `<h2 class="header">`, *etc.*

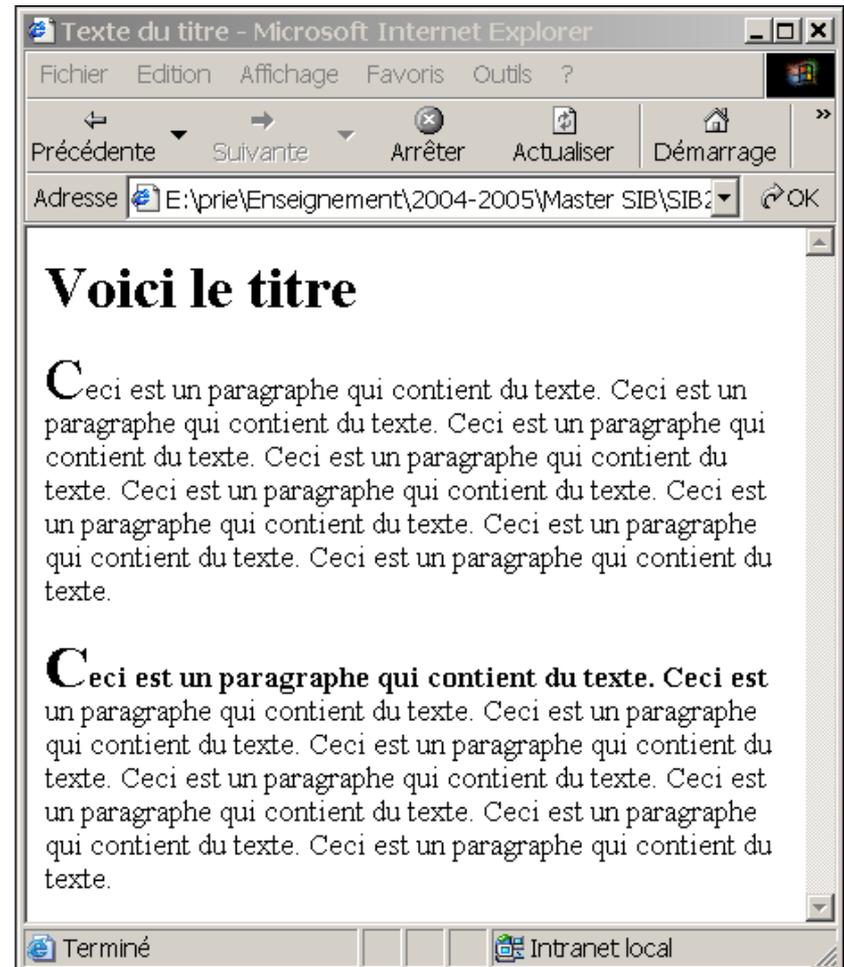
Sélecteurs CSS : pseudo-classes

- Sélecteurs qui sélectionnent des éléments en fonction de leur état à un moment donné
- Exemple et intérêt principal
 - `a:link` — lien non visité et inactif
 - `a:visited` — lien déjà visité
 - `a:hover` — lien sur lequel passe le pointeur de la souris
 - `a:active` — lien sur lequel on clique
- Exemple

```
a:link {color: blue;}  
a:visited {color: magenta;}  
a:hover {color: red; text-decoration:none;  
font-weight: bold;}  
a:active {color: red;}
```

Sélecteurs CSS : pseudo-éléments

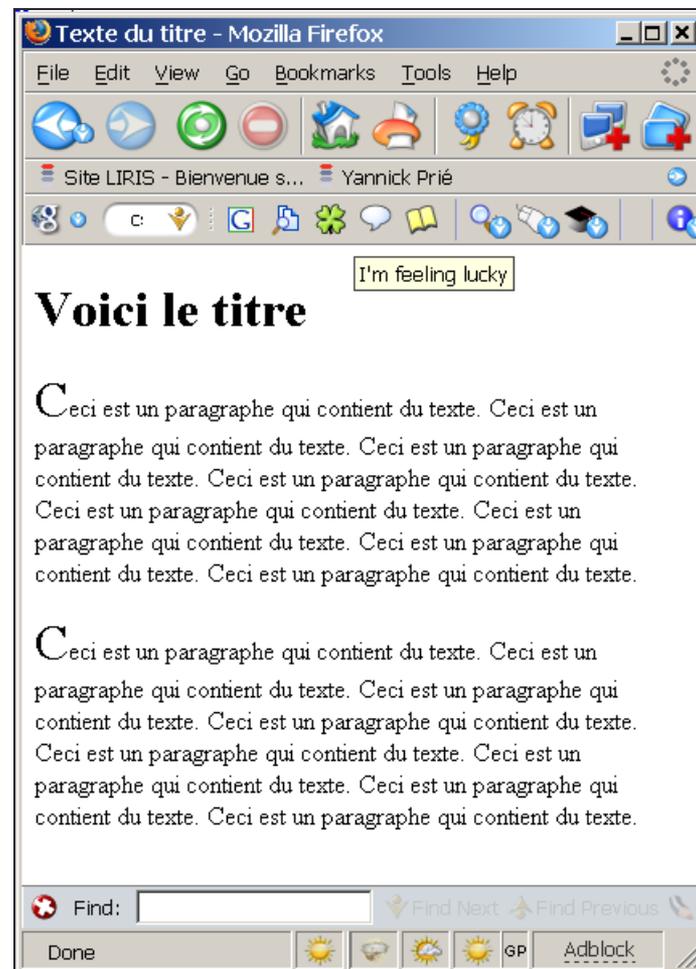
- :first-letter
 - première lettre dans un élément bloc (ex. p, h1, ...)
- :first-line
 - première ligne dans un élément bloc (ex. p, h1, ...)



1. Introduction
 2. Aspects techniques
 3. Page Web
 4. Conclusion
1. Définitions
 2. Langage HTML
 3. Langage CSS
 4. Publication de pages Web

Pseudo-éléments : 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>Texte du titre</title>
<style type="text/css">
  p:first-letter {font-size: 200%;}
  p.lign1:first-line {font: bold;}
</style>
</head>
<body>
<h1>Voici le titre</h1>
<p>Ceci est <!-- ... --> du texte. </p>
<p class="lign1">Ceci <!-- ... --> qui contient
  du texte. Ceci est un paragraphe qui
  contient du texte. </p>
</body>
</html>
```



Sélecteurs CSS : contextuels

- Sélecteurs qui ne sélectionnent que des éléments dans un certain contexte

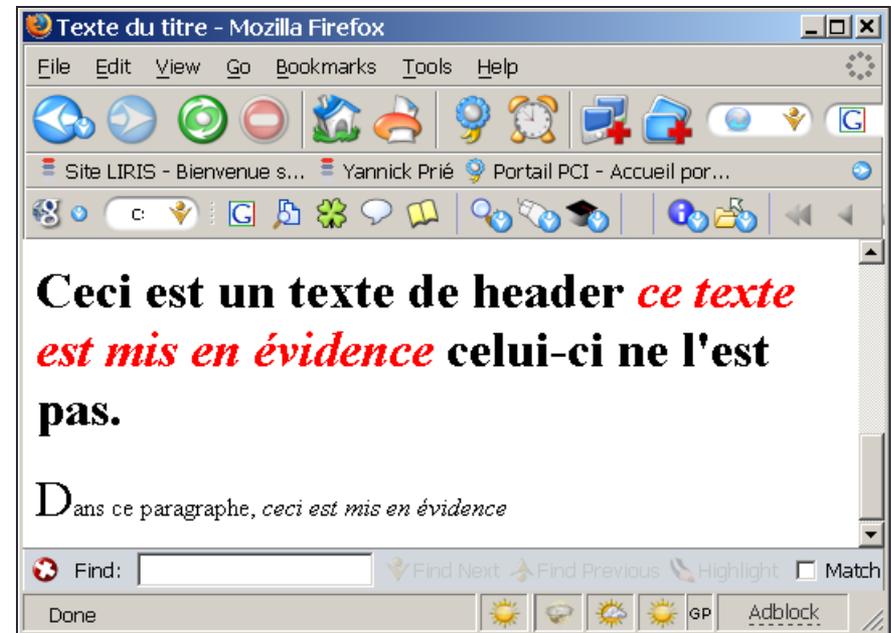
Style :

```
h1 em { color: red; }
```

XHTML :

```
<h1>Ceci est un texte  
de header <em>ce texte  
est mis en  
évidence</em> celui-ci  
ne l'est pas.</h1>
```

```
<p>Dans ce paragraphe,  
<em>ceci est mis en  
évidence</em></p>
```



Quelques propriétés de textes

- font-size:
 - small | medium... | % | x pt
- font-family:
 - fontname1, fontname2 (*si la première n'est pas disponible*), familyname (*serif, sans-serif, etc.*)
- font-weight:
 - bold | lighter
- font-style:
 - italic, oblique
- text-align:
 - left | center | right | justify
- text-indent: (*retrait de première ligne*)
 - % | x cm

Les couleurs

- color:
 - red | blue...| hexcode
- background-color:
 - red | blue...| hexcode

■ Black = "#000000"

■ Silver = "#C0C0C0"

■ Gray = "#808080"

□ White = "#FFFFFF"

■ Maroon = "#800000"

■ Red = "#FF0000"

■ Purple = "#800080"

■ Fuchsia = "#FF00FF"

■ Green = "#008000"

■ Lime = "#00FF00"

■ Olive = "#808000"

■ Yellow = "#FFFF00"

■ Navy = "#000080"

■ Blue = "#0000FF"

■ Teal = "#008080"

■ Aqua = "#00FFFF"

Types d'éléments

- Éléments blocs
 - Prennent la forme d'un bloc dans la page
 - ensemble de lignes
 - ne peuvent être contenus que dans d'autres éléments blocs
 - Exemple
 - p, img, ul, table, h1, **div**...
 - Propriétés de bloc
- Éléments inline
 - S'inscrivent dans la continuité des éléments
 - ne forcent pas un changement de ligne
 - peuvent être inclus dans n'importe quel élément
 - Exemple
 - a, em, **span**...
- Éléments de listes
 - éléments HTML qui ont un marqueur (bullet, number) et un ordre

Rappels sur div et span

- div
 - élément contenant d'autres éléments, servant à définir un bloc
 - servira essentiellement à faire du positionnement de bloc
- span
 - élément contenant d'autres éléments, inline
 - servira essentiellement à regrouper des caractéristiques globales pour des éléments textuels

Propriétés de styles graphiques

- S'appliquent aux éléments blocs
- Attribut **position**
 - **static**
 - bloc placé en fonction de sa position dans l'ordre des balises, ne peut pas être modifiée dynamiquement
 - **absolute**
 - bloc placé précisément par rapport aux bords de la fenêtre d'affichage
 - la position peut être modifiée dynamiquement (déplacement)
 - **relative**
 - bloc positionné par rapport à sa position normale, dans le flux (ex. décalage de 20 pts à droite)

Propriétés de styles graphiques (2)

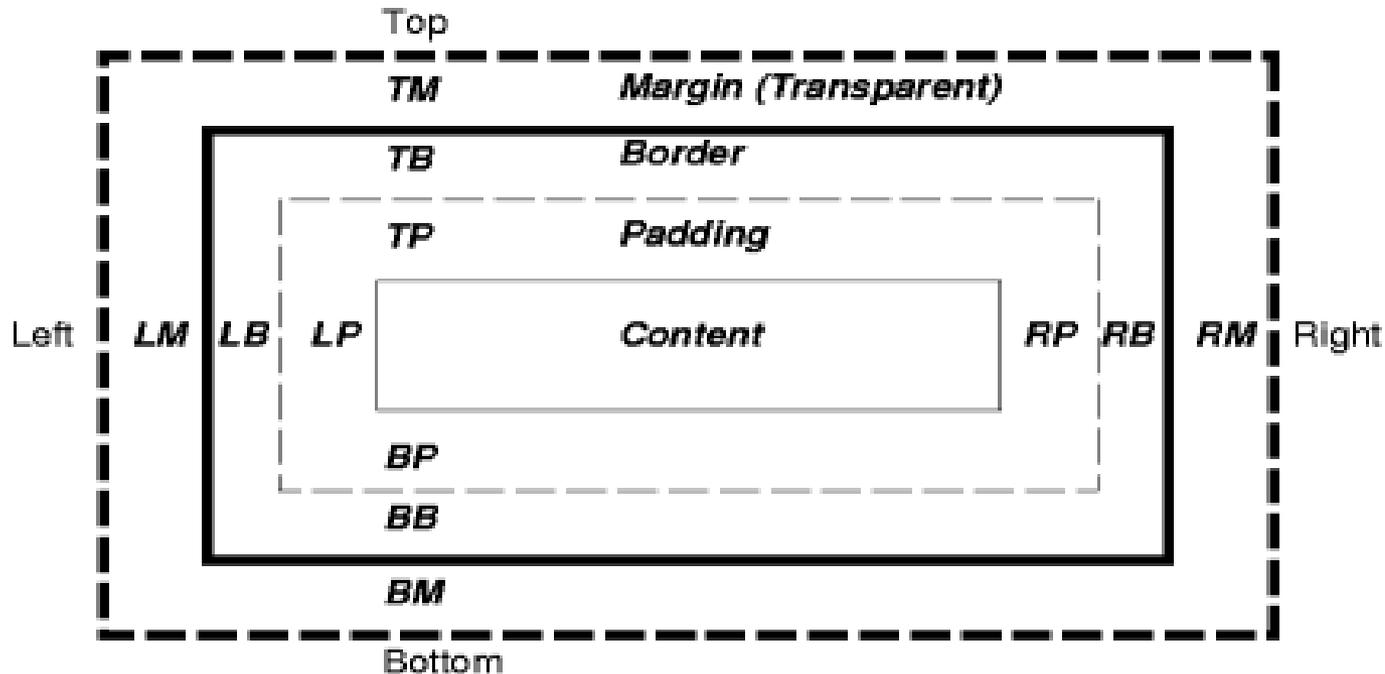
- Attribut **clip**
 - définit une zone de rognage du bloc
 - par exemple pour rogner une image
- Attribut **visibility** et **display**
 - indiquent si (visible|hidden) et comment un élément est affiché
- Attribut **z-index**
 - en cas de superposition de blocs d'affichage, indique l'ordre dans lesquels le navigateur doit les afficher (cf. logiciel de dessin)
- Remarques
 - les distances s'expriment en pixels (**px**), points (**pt**), unités métriques (**cm**, **mm**), ou pourcentages de la taille de la fenêtre (%)

Propriétés de styles graphiques (3)

- Arrière-plan
 - background-color
 - couleur de fond
 - background-image
 - image de fond
 - background-repeat
 - types de répétition de l'image
 - background-attachement
 - arrière-plan fixe quand on utilise l'ascenseur
 - background-position
 - position par rapport au coin supérieur gauche
 - background
 - attribut de résumé
- Exemple
 - `p { background: image.gif fixed repeat-y }`

- | | |
|-----------------------|-----------------------------|
| 1. Introduction | 1. Définitions |
| 2. Aspects techniques | 2. Langage HTML |
| 3. Page Web | 3. Langage CSS |
| 4. Conclusion | 4. Publication de pages Web |

Marge, bordures, padding



- Margin edge
- Border edge
- Padding edge
- Content edge

CSS définit des attributs pour définir précisément ces zones

Styles intégrés

- Déclarés comme attribut `style` d'un élément
- Exemple
 - `<h1 style="font-family: Arial; font-style: italic;">Un texte qui se retrouvera en arial italique</h1>`

Feuilles de styles incorporées

- La feuille de style est déclarée dans l'en-tête (**head**) du document XHTML
- Elle s'appliquera aux éléments dans **body**
- Exemple

```
<html>
  <head>
    <title>Texte du titre</title>
    <style type="text/css">
      .important {color: red;}
      h1 {font-family: Arial; font-style: italic; }
      em { color: green }
    </style>
  </head>
  <body>
    <!-- éléments sur lesquels s'appliqueront les styles -->
  </body>
</html>
```

Feuilles de styles liées

- Les règles se trouvent dans un fichier extérieur
- On indique au navigateur où se trouve cette ressource de style, en utilisant un élément `link` dans l'en-tête
 - `<link rel="stylesheet" type="text/css" href="fichier-de-style.css" >`
- Intérêt
 - Plusieurs documents XHTML peuvent faire appel à la même feuille de style

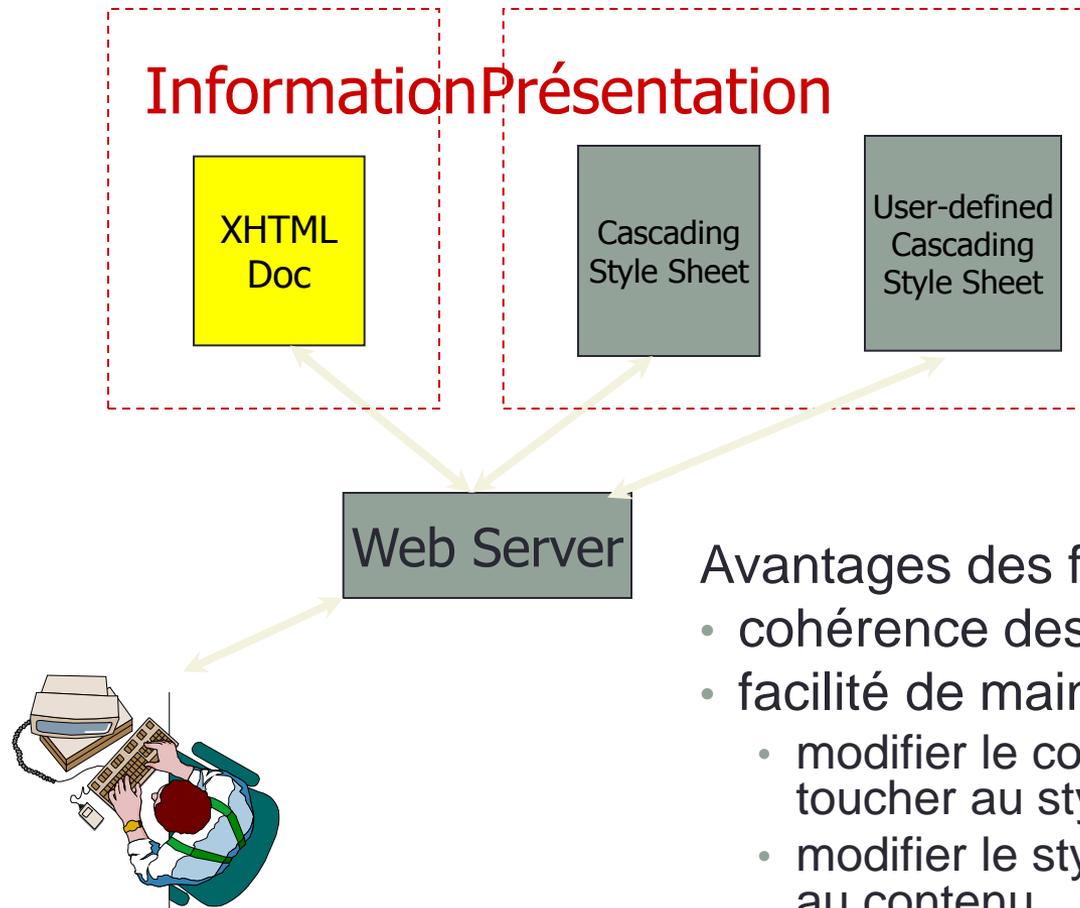
Héritage de styles

- Par défaut
 - Les styles d'un élément sont héritées par ses éléments descendants
 - Exemple :
 - `<p style="color: red ;">Du texte mis en évidence, pas mis en évidence.</p>`
 - Du texte *mis en évidence*, pas mis en évidence.
- A condition que l'héritage ait un sens
 - *i.e.* que les caractéristiques soient applicables à l'élément enfant
 - un positionnement de bloc n'a pas d'intérêt pour un élément `em` qui y est contenu
 - si un style est défini spécialement pour un élément (ex. `em { color: blue; }`), l'héritage ne se fait pas
 - Du texte *mis en évidence*, pas mis en évidence.

Cascading style sheets : cascade

- On peut avoir concurrence entre plusieurs styles définis dans de multiples endroits
 - styles par défaut (1- navigateur)
 - fichiers CSS externes (2- spécifications globales au site)
 - élément `head` du document XHTML (3- spécification globales au doc.)
 - attributs `style` des éléments (4- spécification locales)
 - style utilisateur (5- spécification de l'utilisateur)
- Notion de cascade ou ordre de priorité des styles
 - trouver toutes les déclarations qui s'appliquent à un élément
 - les classer par spécificité
 - système de poids qui s'ajoutent
 - les classer par ordre d'apparence
 - plus un déclaration apparaît tard, plus elle a de poids
- Exemple
 - style (4) > style (4 hérité) > style (3) > style (2) > style (1)

Présentation avec des CSS



Avantages des feuilles de style

- cohérence des présentations
- facilité de maintenance
 - modifier le contenu sans toucher au style
 - modifier le style sans toucher au contenu

Conclusion sur CSS

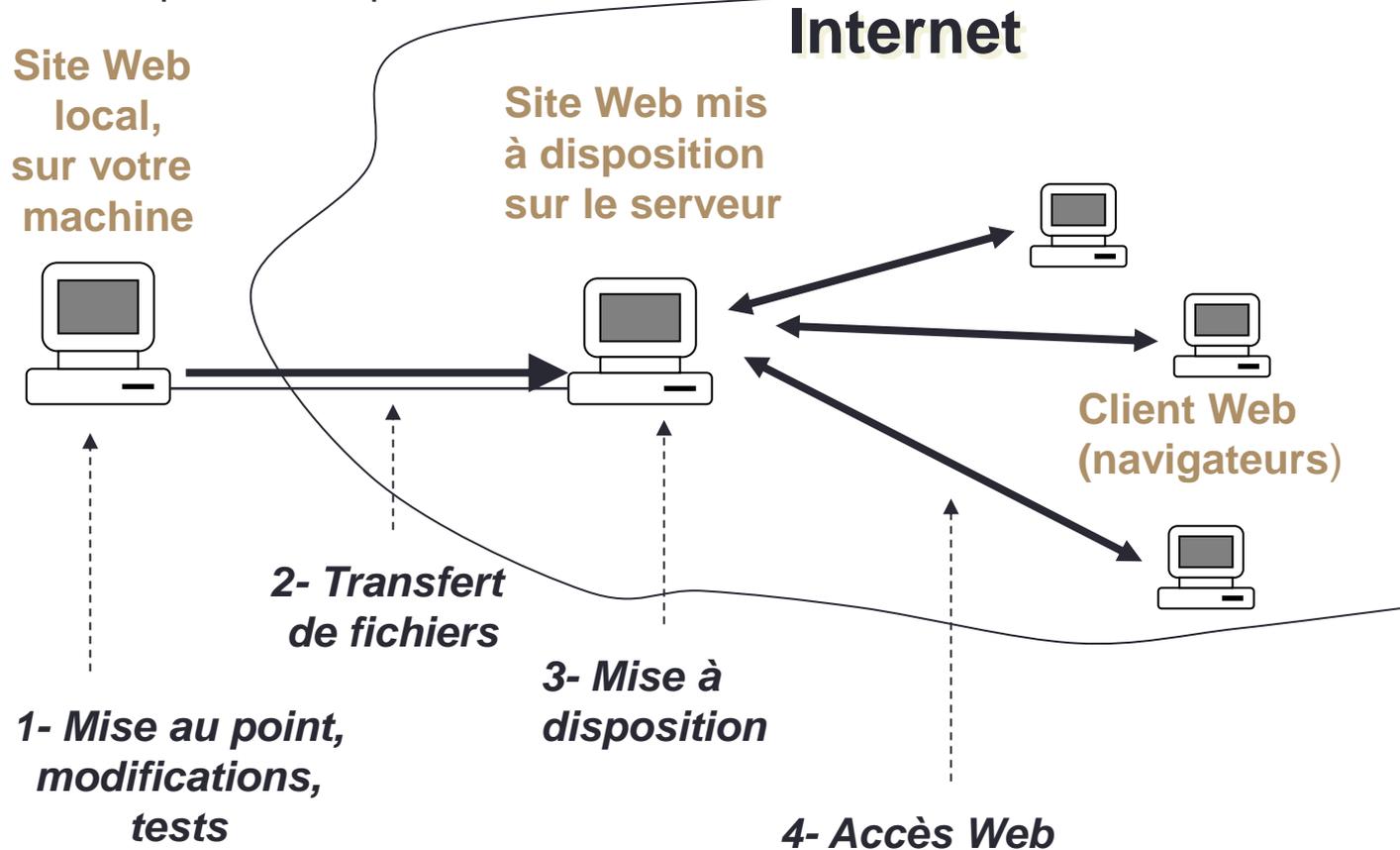
- Versions
 - CSS2 caractéristiques de style « simples »
 - CSS3 styles « améliorés »
- Les navigateurs gèrent diversement les feuilles de style !
 - rester simple
 - tester avec plusieurs navigateurs
- Aller voir les références
- Valider !

Plusieurs situations type

- Cas d'un particulier
 - site hébergé par un Fournisseur d'Accès Internet
 - conception en local à la maison
 - transfert des fichiers chez le FAI par FTP
 - site sur une machine à la maison
 - conception en local sur une partie du disque
 - copie de fichiers sur le site publié
- Cas d'une entreprise
 - site externalisé (conception, mise à disposition)
 - site conçu en interne et mis à disposition sur un serveur de l'entreprise

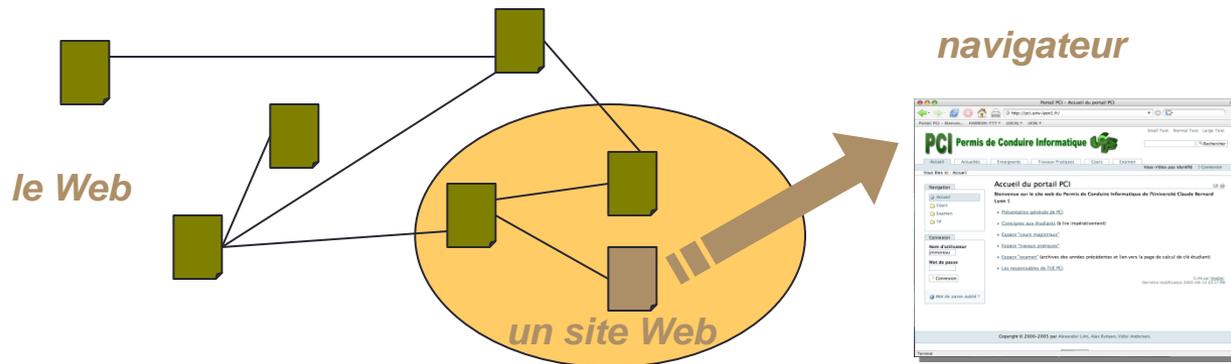
Conception, transfert et accès

- Dans tous les cas
 - Cycle de conception et de publication d'un site Web



1. Introduction
2. Aspects techniques
3. Page Web
4. Conclusion

Conclusion : une description plus formelle



- World Wide Web :
 - Ensemble de ressources interreliées formant un **réseau** (ou une toile) sur lequel il est possible de naviguer
- Navigateur
 - Outil permettant d'accéder à ces ressources et d'exploiter les liens hypermédias pour la navigation

Remerciements

- Sources de ces transparents
 - PCI (Pratiques et Compétences Informatiques, UCBL, 2006)
 - MIF13 (M1 informatique, UCBL, 2013)
 - Cours de Yannick Prié (M1 SIB, UCBL, 2009)
 - Professional Web Authoring With XHTML and CSS – Roy Tennant
- Images
 - deviantART : <http://e-designer.deviantart.com/>