POLYTECH PARIS-SUD

UNIVERSITÉ PARIS SUD

Introduction au Web

Claude Barras

Université Paris-Sud

Plan

- Introduction
  - Le Web
  - Architecture distribuée
  - Navigateurs
  - Les URL
  - HTML
  - Balises et attributs
  - Structure d'un document
  - Encodage
- Evolution de HTML
  - CSS
  - XML
  - XHTML
  - HTML 5
- Web dynamique
  - Client dynamique
  - Serveur dynamique
  - Scripts Serveur
  - Session
  - AJAX
- Conclusions

1. Introduction
Le Web

- Qu'est ce que le Web (World Wide Web) ?
  - à l'origine, de **l'hypermédia distribué**
    - hypertexte : accès non séquentiel aux documents (hyperliens)
    - multimédia : textes, images et sons
    - distribué : accès par le réseau Internet
  - créé par **Tim Berners-Lee** en 1989 au CERN (Genève)
- Briques technologiques de base
  - URL (Uniform Resource Locator)

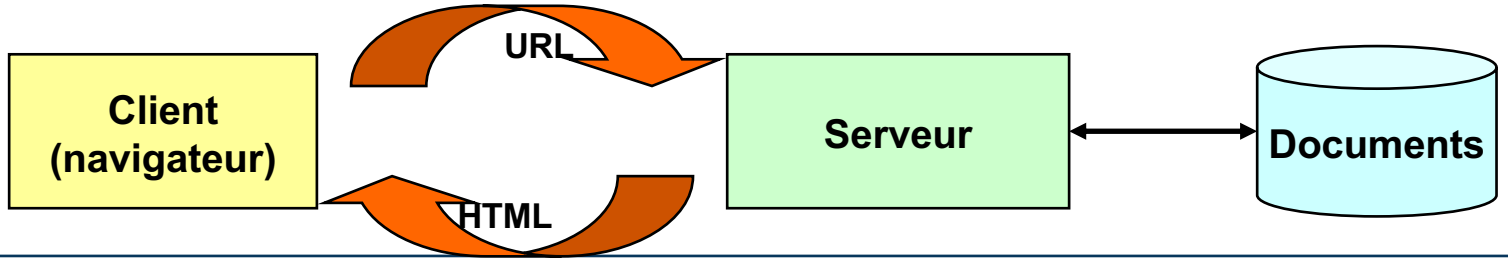
    Désignation des documents (localisation) pour les hyperliens
  - HTTP (Hypertext Markup Protocol)

    Protocole de distribution des documents
  - HTML (Hypertext Markup Language)

    Format des documents hypermédia

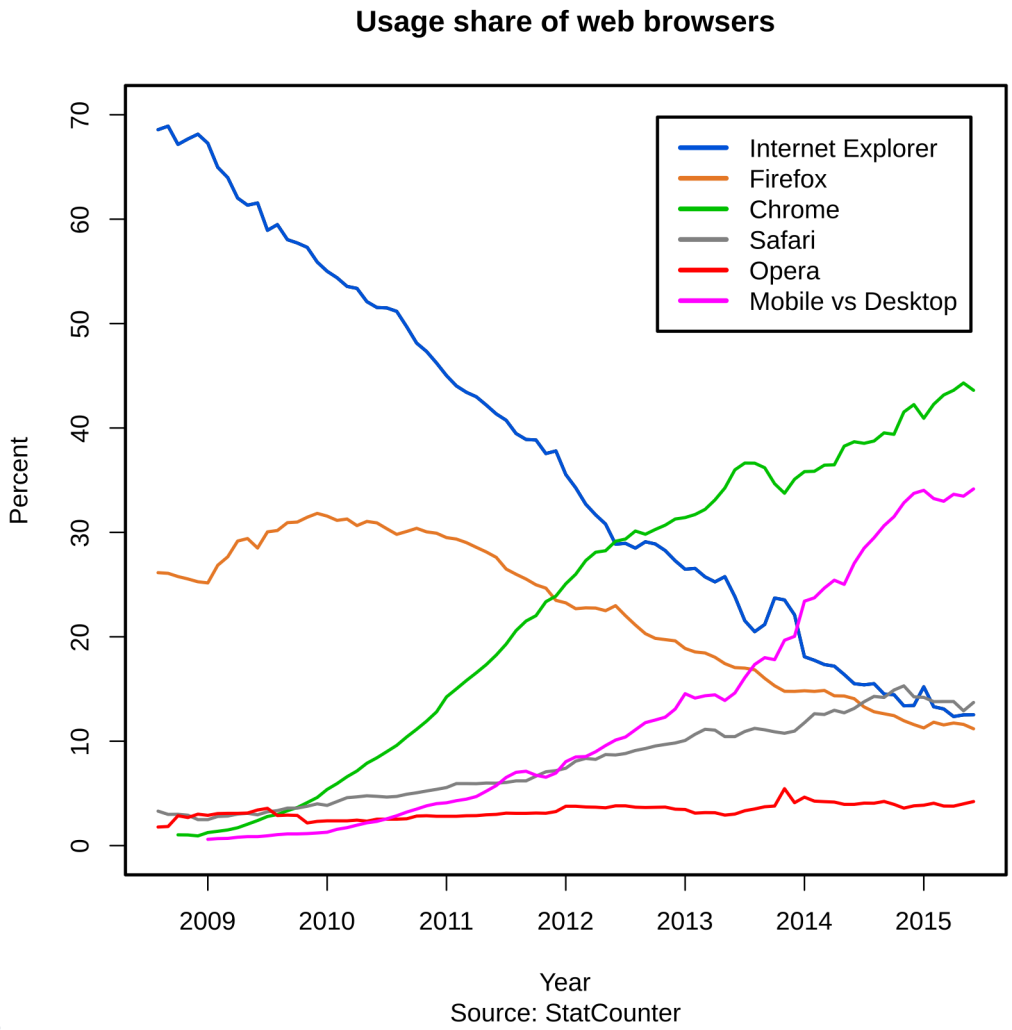
1. Introduction
Architecture distribuée

- Le client (navigateur: Explorer, Chrome, Firefox…)
  - demande au serveur des informations
  - affiche les pages pour l'utilisateur
- Le serveur (Apache HTTP, Microsoft IIS, nginx…)
  - reçoit en permanence les requêtes du client
  - renvoie les documents correspondants
- Le réseau Internet
  - basé sur le protocole TCP/IP
  - chaque machine dispose d'une numéro IP et d'un nom associé grâce aux serveurs de nom de domaine (DNS)



1. Introduction
Navigateurs

- Guerre des navigateurs (95-02): Netscape vs. Microsoft
- Depuis : baisse d'IE au profit de Firefox (surtout en Europe) puis Chrome
- Explosion de l'accès mobile



1. Introduction
URL et URI

- Syntaxe d'un URL (Uniform Resource Locator) :
  - http://www.limsi.fr:80/Individu/barras/index.html#cours

    protocole **machine** **port** **répertoire** **fichier** **fragment**
  - principaux protocoles utilisés dans des URL :

    ftp, http, https, news, nntp, mailto, telnet…
- URI (Universal Resource Identifier) : généralisation des URL

  possibilité d'utiliser aussi des noms symboliques (URN)
- Syntaxe d'un URI :
  - protocole ":" identification (// … / … # ... ? … + … +…)
    - / : séparateur hiérarchique d'accès au document
    - # (fragment) : désigne une partie d'un document
    - ? (query): ce qui suit est transmis au serveur ("+" pour espace)
    - %(code) : caractère d'échappement (ex: %25)
  - URI/URL relatifs

1. Introduction
HTML

- HTML est un langage de balises
  - dérivé de SGML (Standard Generalized Markup Language)
  - alternance de texte et de balises
    - séparation claire entre le contenu et la présentation
- HTML décrit la présentation de documents hypermédia
  - textes, listes, tableaux, images, hyperliens, formulaires…
- Recommandation du World Wide Web Consortium (W3C)

  http://www.w3.org/
  - Groupes de travail
    - Architecture : HTTP, URI, XML, DOM…
    - Interface : HTML, styles, math, graphiques, accès vocal ou par mobiles…
  - Processus W3C
    - Working draft, Candidate Recommandation, Proposed Recommandation puis W3C Recommandation
  - http://www.w3.org/Markup/

1. Introduction
Balises et attributs

- Balises de début et de fin d'élément (avec contenu)

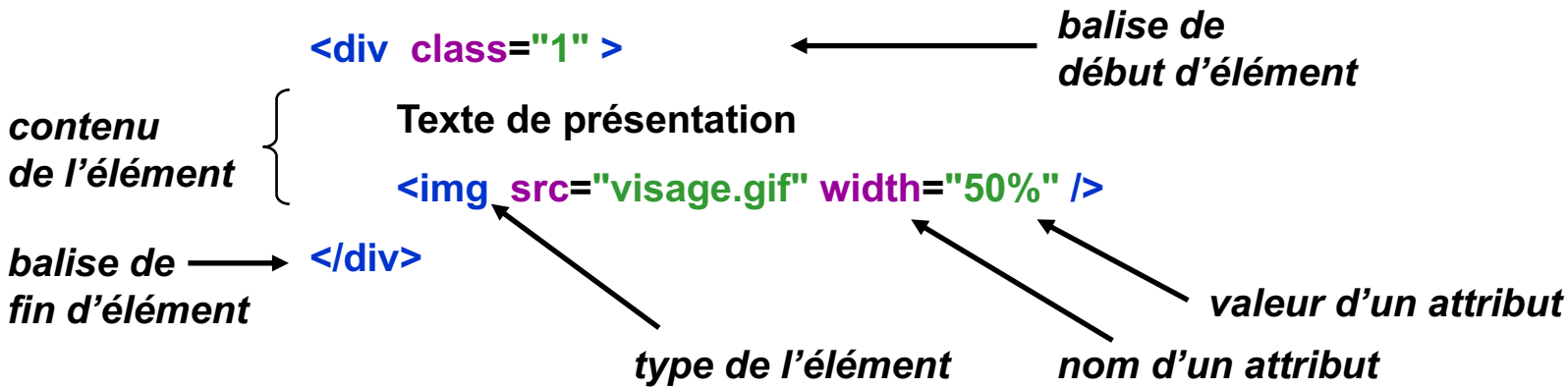
  `<title> Document </title>`

  ou des balises vides (sans contenu)

  `<br />`

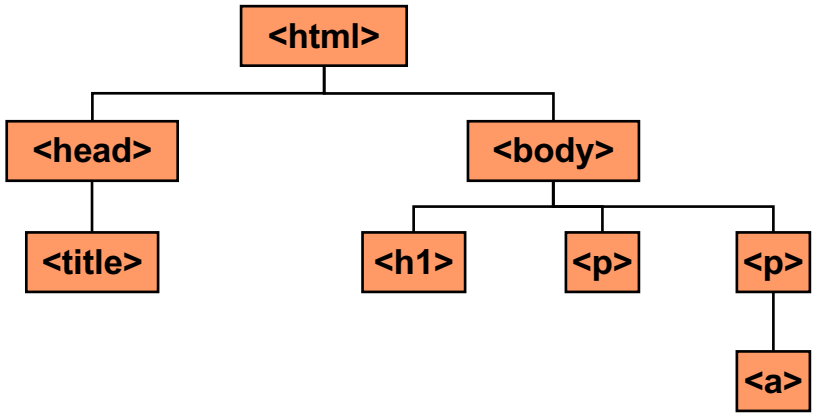
- Attributs d'un élément

  une liste de couples attribut/valeur peut être associée à chaque balise de début d'élément ou à chaque balise vide



Structure d'un document

- Deux parties principales :
  - entête
  - corps




```
<html>
<head>
  <!-- Created with MyTool-2.0 -->
  <title>Un page &eacute;l&eacute;mentaire</title>
</head>
<body>
<h1>
  Une page simple
</h1>
<p>Voici un exemple de page simple.</p>
<p><a href="source.html" target="_blank">Ce lien ouvrira une
fen&ecirc;tre contenant le source de cette page.</a></p>
</body>
</html>
```

1. Introduction
Encodage

- Différents encodages possibles
  - ASCII: 7 bits, 96 caractères (26 lettres latines)
  - ISO-8859-1 (latin-1): 8 bits, 96 caractères supplémentaires (diacritiques)
  - Universal Multiple-Octet Coded Character Set (UCS), ISO 10646
    - Unicode (http://www.unicode.org/): 16 bits (UCS-2)
    - UTF-8 : encodage UCS de longueur variable, compatible ASCII
- Choix de l'encodage
  - au cours de la transmission par protocole HTTP
  - dans l'entête du document (différent suivant HTML ou XHTML)
- Caractères spéciaux
  - représentation compatible avec tout encodage
  - exemple: 'é' : é ou é

2. Evolution de HTML

- Évolutions continues de HTML depuis ses origines
- Applications moins techniques et plus commerciales
  - Évolution en direction de la présentation, au détriment de la structuration (problèmes d'accessibilité pour les mobiles, par synthèse vocale…)
  - Rajout de nouvelles balises et attributs de balises "propriétaires"

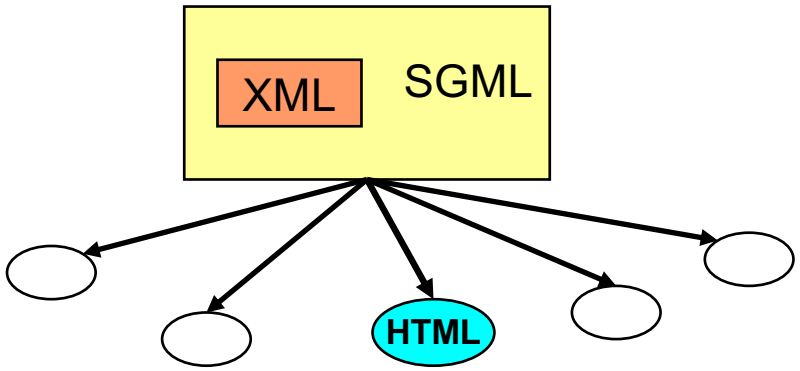
    (problèmes de compatibilité entre versions de navigateurs)
- Solutions pour retrouver la flexibilité
  - gestion de feuilles de style, pour séparer présentation et contenu
    - CSS (Cascading Style Sheets)
  - retour à la source (SGML) en la simplifiant :
    - XML (Extensible Markup Language)

2. Evolution de HTML
CSS

- Commandes de formatage visuel (police, taille, couleurs…)
  - absence de structure
    - inexploitable dans certains contextes (smartphone, synthèse vocale, braille…)
    - maintenance difficile
- Feuilles de style CSS (Cascading Style Sheets)
  - un contrôle global et local du formatage
  - séparation entre la présentation et le contenu
  - un « look » plus homogène (charte graphique)
- Héritage hiérarchique des styles définis par des règles
  - sélecteur1, sélecteur2,... { propriété1: valeur1; … }

2. Evolution de HTML
XML

- XML (Extensible Markup Language)
  - Recommandation du W3C de février 1998, dernière révision 2006.
  - Objectifs :
    - « super » HTML
    - SGML allégé pour le Web
  - Sous ensemble (simplifié mais compatible) de SGML



2. Evolution de HTML
XHTML

- XHTML re-exprime HTML dans la syntaxe restreinte XML
  - facilite le traitement automatique et l'évolution vers XML
- La syntaxe de XHTML est plus stricte, ex:
  - différence majuscules/minuscules
    - <html> au lieu de <HTML> ou <Html>
  - valeurs des attributs entre guillemets
    - <table border="3"> au lieu de <table border=3>
  - une balise fermante est toujours associée à la balise ouvrante (sinon il faut utiliser une balise vide)
    - … ou
 au lieu de <P> … ou
- Évoluer de HTML à XHTML:
  - mettre tous les noms des balises en minuscules, rajouter des guillemets, des balises fermantes...
  - Services de validation et de conversion de pages HTML du W3C

2. Evolution de HTML
HTML 5

- Versions de HTML produites par le W3C
  - HTML 2.0 (1994-95), HTML 3.2 (1997)
  - HTML 4.0 (1998), HTML 4.01 (révision de décembre 1999)
  - XHTML 1.0 (2000, rev. 2002) : convergence entre HTML et XML
- HTML 5 (en cours) réintègre les travaux sur XHTML2 (stoppé)
  - Les syntaxes XHTML et HTML « standard » coexistent
  - Une partie de l'héritage SGML devient obsolète (syntaxe DTD…)
- Apports
  - Structuration sémantique
  - Formulaires améliorés
  - Extensions graphiques et multimédia
    - Canvas (bitmap) et SVG (vectoriel)
    - Audio et Video
  - …

3. Web dynamique

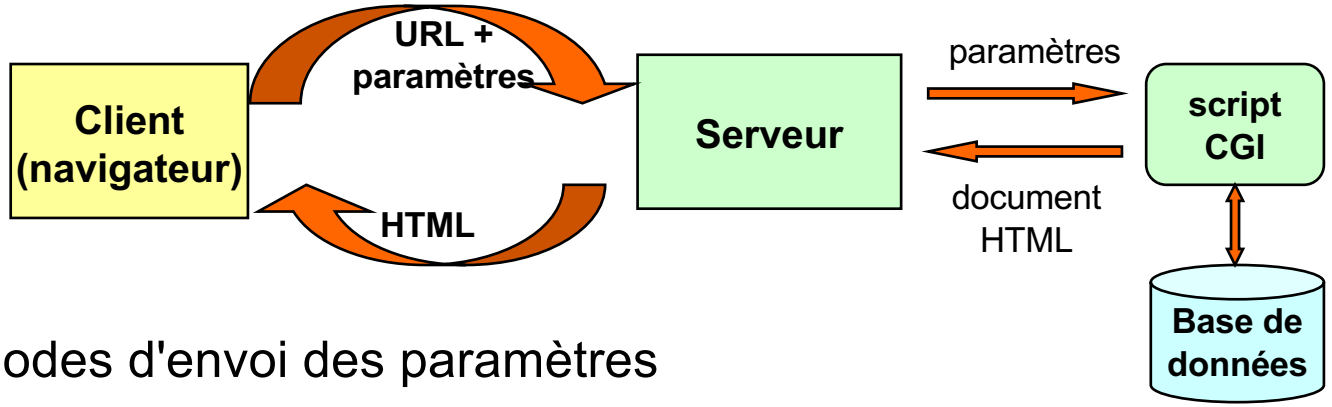
- Pour faire une application Web, deux points de vue
  - Côté client
    - le code HTML standard est très statique
      - on veut réagir plus vite aux actions de l'utilisateur
    - exécution de code par le navigateur
      - les programmes sont transmis sur le réseau
  - Côté serveur
    - le serveur fabrique la page en fonction des requêtes
      - charge de calcul pour le serveur
    - le navigateur reçoit du HTML standard

3. Web dynamique
Client dynamique

- Scripts
  - élément **<script>** dans l'entête d'un document HTML
    - Intégration de code en langage **JavaScript**
  - exécution de scripts déclenchée par des **événements**
    - clics ou déplacements de la souris, touches du clavier…
- Autres langages
  - élément **<applet>** : lance l'exécution de code Java sur le client
    - remplacé par l'élément générique **<object>** en HTML5
- Soucis de sécurité, de fiabilité
  - le fonctionnement des scripts peut dépendre du navigateur
    - optimisation considérable des performances de JavaScript

3. Web dynamique
Serveur dynamique

- Scripts CGI (Common Gateway Interface)
  - le serveur génère une page HTML
    - typiquement en réponse à un formulaire



- Deux méthodes d'envoi des paramètres
  - ajoutées à la fin de l'URL (GET)

    http://www.azcom.fr/saisie.asp?Name=Durand&Model=Ford
  - envoyées séparément par l'intermédiaire du protocole HTTP (POST)

3. Web dynamique
Scripts Serveur

- langages de script plus adaptés: PHP, ASP, JSP
  - inclusion dans le code HTML
  - interprétation par le serveur
  - analyse simplifiée des paramètres
  - transparent pour le client (reçoit exclusivement du code HTML)
- Hypertext Preprocessor (PHP, `php.net`)
  - très répandu (simple, gratuit, fonctionne avec Apache)
    - extension .php ou dérivée
    - syntaxe inspirée de C/Java/Perl, liens avec les bases de données…

```
<?php  if(strstr($HTTP_USER_AGENT,"MSIE")) {
        echo "You are using Internet Explorer<br>"; } ?>
```

3. Web dynamique
Session

- Navigation web en mode déconnecté
  - pas de gestion des sessions, pas d'historique…
- Solution: générer un identifiant de session et le propager à chaque page
- Deux possibilités:
  1. chaque page reçoit l'identifiant en paramètre et le retransmet (formulaire avec champ caché)
     - pas très sécurisé
  2. création et utilisation d'un Cookie qui contient l'identifiant
     - caractéristiques d'un Cookie:
       - nom, valeur, domaine, durée de validité
     - il faut que l'utilisation des Cookies soit activée
- Les langages de script (PHP…) facilitent la gestion des sessions

3. Web dynamique
AJAX

- Une véritable application Web
  - ne pas changer de « page » à chaque action de l'utilisateur…
- La solution AJAX (Asynchronous JavaScript and XML)
  - envoi d'une requête au serveur
    - XMLHttpRequest
  - attente non bloquante (asynchrone) de la réponse du serveur
    - réception d'un objet XML (ou JSON)
  - modification partielle de la page courante
    - utilisation de JavaScript et du DOM (accès au contenu de la page Web)

Conclusions

- Nombreux langages et frameworks de développement côté serveur
  - ASP.NET (C#), J2EE (Java), Ruby on Rails (Ruby), AngularJS (JavaScript), Django (Python), Symfony (PHP)…
- …et aussi côté client
  - librairies JavaScript et CSS : jQuery, Bootstrap…
- Innombrables ressources documentaires disponibles sur le Web
  - http://www.w3.org/
  - www.w3schools.com
  - www.openclassrooms.com