

Web multimédia

Cours 0 — Introduction

Quentin Bouillaguet

`quentin.bouillaguet@u-psud.fr`

2019–2020

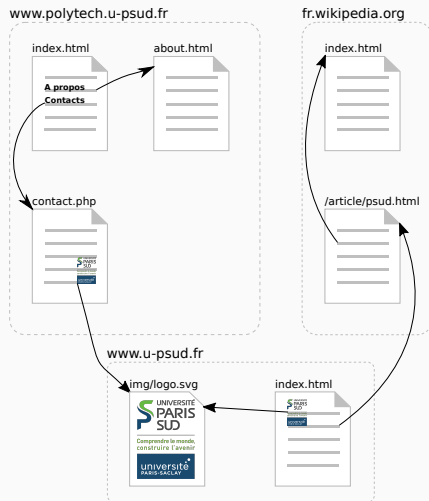
Polytech Paris Sud

World Wide Web

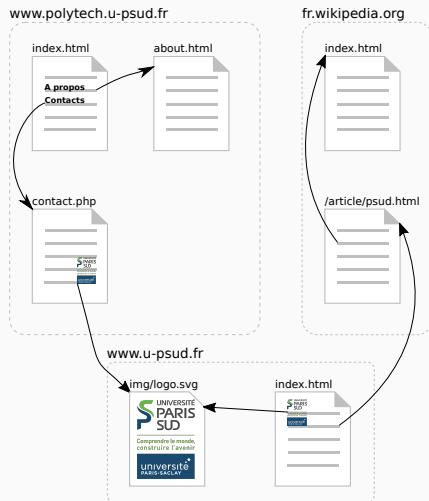
The web is more a social creation than a technical one. I designed it for a social effect — to help people work together — and not as a technical toy. The ultimate goal of the Web is to support and improve our weblike existence in the world. We clump into families, associations, and companies. We develop trust across the miles and distrust around the corner.

— Tim Berners Lee

- Hypertexte
 - accès non séquentiel aux documents
 - identification et localisation des documents
- Hyperlien
 - lien vers une localisation
- Hypermédia
 - hypertexte étendu aux médias (images, sons, vidéos, ...)



- Qu'est ce que le Web (World Wide Web) ?
 - **graphe** de documents hypermédias **décentralisé**
 - les documents **réfèrent** d'autres documents
 - liens **unidirectionnels**
 - aucune garantie de disponibilité, de cohérence
- crée par **Tim Berners-Lee** en 1989 au CERN (Genève)



Interaction avec des documents hypertextes/hypermédias:

URL (Uniform Resource Locator)

Localisation des ressources sur Internet (documents hypertextes, images, ...).

HTTP (HyperText Transfer Protocol)

Protocole de communication client pour effectuer des actions sur un serveur.

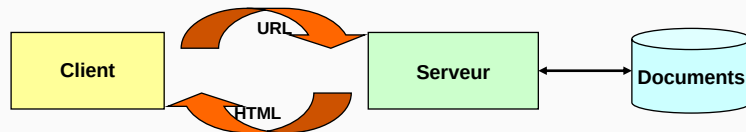
HTML (HyperText Markup Language)

Format des documents hypermédias.

- Une ressource est **identifiée** et **localisée** par un URL.
 - <http://www.lri.fr:80/~bouillaguet/teaching/index.html#web>
protocole serveur port répertoire fichier fragment
 - principaux protocoles utilisés dans des URL:
ftp, http, https, mailto, telnet, ...
- URI (Universal Resource Identifier): Généralisation des URL
 - **identification** des ressources
 - URN: indépendance vis à vis de la localisation (ex. ISBN pour les livres)
- Syntaxe d'un URI:
 - protocole ":" identification (// ... / ... # ... ? ... + ... + ...)
 - / séparateur hiérarchique d'accès à des documents
 - # (fragment): désigne une partie du document
 - ? (requête): ce qui suit est transmis au serveur ("+" pour espace)
 - % (code): caractère d'échappement (ex: %25)
 - URI/URL relatifs

HyperText Transfer Protocol (HTTP)

- Client vers n serveurs (modèle distribué)
- Ne conserve pas d'état



- Le **client** (navigateur) fait une **requête** d'accès à une **ressource** auprès d'un serveur Web selon le protocole **HTTP**.
- Le **serveur** reçoit en permanence des **requêtes** et renvoie éventuellement la **ressource** demandée.
- Le **client** interprète la **réponse** reçue et renvoie des **requêtes** pour les **ressources** restantes.

Requête

GET /index.html HTTP/1.1

Host: www.lri.fr

Réponse

HTTP/1.1 **404 Not Found**

Date: Fri, 14 Feb 2020 11:54:17 GMT

Server: Apache/2.4.7 (Ubuntu)

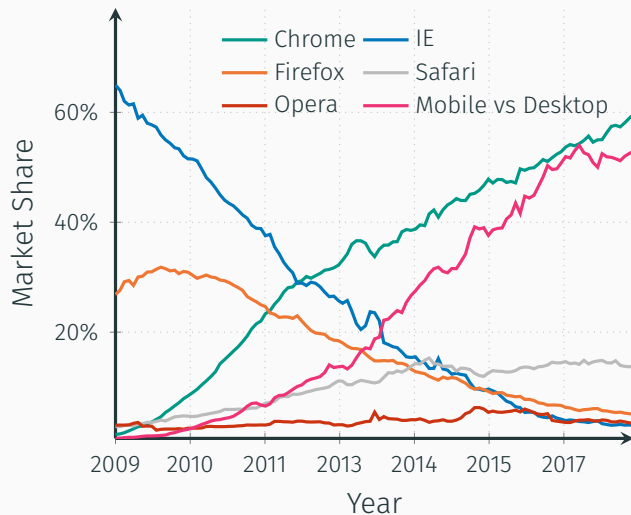
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

<html>

...

- Navigateurs web
 - Guerre des navigateurs
 - Explosion des plate-formes mobile
- Robots d'indexation (web crawler)
- ...



HTML

- HTML est un langage de balises
 - dérivé de SGML (Standard Generalized Markup Language)
 - alternance de texte et de balises
 - séparation claire entre contenu et présentation
- HTML décrit la présentation de documents hypermédia
 - textes, listes, tableaux, images, hyperliens, formulaire, ...
- Recommandation du World Wide Web Consortium (W3C)
 - Groupes de travail
 - Architecture: HTTP, URI, XML, DOM, ...
 - Interface: HTML, styles, graphiques, accessibilité, ...
 - Processus W3C
 - Working draft, Candidate Recommandation, Proposed Recommandation puis W3C Recommandation
 - <http://www.w3.org/Markup/>

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

```
<title>Cat</title>
```

ou des balises vides (sans contenu)

```
<br />
```

- Attributs d'un élément

```
<div class="behavior">
```

```
  Outdoor cats are active both day and night.
```

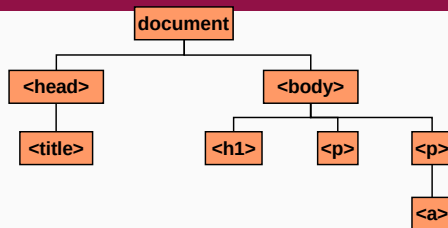
```
  
```

```
</div>
```

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

Structure d'un document

- Deux parties principales
 - entête
 - corps



```
<!doctype html>
<html>
  <head>
    <!-- Document generated by MyTool v3.0 -->
    <title>Une 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&eacute;tre contenant le source de cette page.</a></p>
  </body>
</html>
```

- 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

- 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 navigateurs)
- Solutions pour retrouver la flexibilité
 - gestion de feuilles de style, pour séparer présentation et contenu
 - retour à la source (SGML) en la simplifiant

Feuilles de styles CSS (Cascading Style Sheet)

- Séparation entre contenu (HTML) et style d'affichage
- Contrôle global et local du formatage
 - selon le type de balise (ex. `<h1>`, `<p>`, ...)
 - selon l'attribut `class` (ex. `<div class="#behavior">`)
 - selon le type de média/d'appareil
- Cohérence du style graphique
- Héritage hiérarchique des styles définis par des règles
 - `selecteur1, selecteur2, ...{ propriete1: valeur1; ...}`

XML (Extensible Markup Language)

- Recommandation du W3C de février 1998, dernière révision 2006
- Dérivé de SGML tout comme HTML
- Possède une **DTD** (Document Type Description) tout comme SGML
 - Spécifie la structure d'un langage XML
 - Permet la vérification de conformité
- **DOM** (Document Object Model): lecture complète du document dans une structure d'arbre

- XHTML ré-exprime HTML dans la syntaxe restreinte XML
 - Facilite le traitement automatique et l'évolution vers XML (ex. DOM)
- La syntaxe de XHTML est plus stricte, ex:
 - différence majuscules/minuscules
`<html>` au lieu de `<HTML>` ou `<Html>`
 - valeur 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)
`<p>...</p>` ou `
` au lieu de `<P>` ou `
`

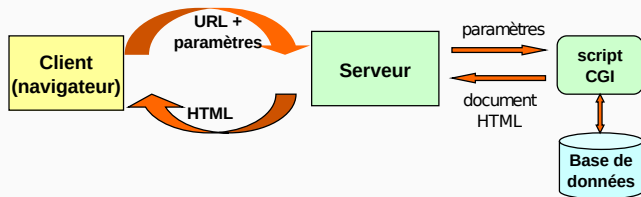
- HTML5 réintègre les travaux sur XHTML2 (stoppé)
 - Les syntaxes XHTML et HTML “standard” coexistent
 - Une partie de l’héritage SGML devient obsolète (DTD, ...)
- Apports
 - Structuration sémantique
 - Formulaire améliorés
 - Extensions graphiques et multimédia
 - Canvas (bitmap) et SVG (vectoriel)
 - Audio et vidéo
 - WebGL (3D)
 - Et bien d’autres...
- Standard évolutif

Web dynamique

Le traitement du côté client

- JavaScript
 - élément `<script>` dans l'entête d'un document HTML
 - programmation événementielle
 - clics ou déplacements de la souris, touches du clavier, redimensionnement, ...
 - page modifiée côté client
- Autres langages
 - élément `<applet>` pour l'exécution de code Java
 - remplacé par l'élément générique `<object>` en HTML 5
 - WebAssembly
- Soucis de sécurité, de fiabilité
 - le fonctionnement des scripts peut dépendre du navigateur
 - optimisation considérable des performances de JavaScript

- Scripts CGI (Common Gateway Interface)
 - le serveur génère une page HTML



- Deux méthodes d'envoi des paramètres
 - ajoutées à la fin de l'URL (GET)
`http://www.amazon.fr/search.php?kind=game&name=Demon+Souls`
 - envoyées séparément par l'intermédiaire du protocole HTTP (POST)
 - par le biais d'un formulaire

- Inclusion ou génération de code HTML
- Interprétation par le serveur
- Analyse simplifiée des paramètres
- Transparent pour le client (reçoit exclusivement du code HTML)

Différents langages de scripts:

- PHP, Ruby (Ruby on Rails, ...), Javascript (Node.js), Python (Django, ...), Java

- Problème: HTTP ne conserve pas d'état (session)
- Solution: Générer un identifiant de session et le propager à chaque page
- Plusieurs mécanismes pour le faire:
 - à travers l'URL
 - par une variable cachée (même mécanisme que les formulaires)
 - par un "cookie"
 - nom, valeur, domaine, durée de validité
 - stockage Web (HTML5)
- Nativement supportés par les cadres applicatifs côté serveur