

# Science(s) informatique(s) : compléments 2018-2019 Web

Sylvie Alayrangues<sup>1</sup>

Gilles Subrenat<sup>1</sup>

<sup>1</sup>Université de Poitiers Laboratoire XLIM-SIC  
Sylvie.Alayrangues@univ-poitiers.fr

Informatique, 2018-2019

## Web $\neq$ Internet

*« I just had to take the hypertext idea and connect it to the TCP and DNS ideas and – ta-da! – the World Wide Web. » Tim Berners-Lee*

Web  $\neq$  Internet

Ingrédients « initiaux » du World Wide Web

- Pages **HTML** (langage de description de texte + hyperliens)
- Navigateur (ou plus généralement **User Agent**)
- **URL**
- Protocole **HTTP**
- Architecture **Client-Serveur**

développement des **standards du Web**

W3C : World Wide Web Consortium

Web  $\neq$  Internet

## Ingrédients du World Wide Web

- Pages **HTML** (langage de description de texte + hyperliens)
- Navigateur (ou plus généralement **User Agent**)
- **URL**
- Protocole **HTTP**
- Architecture **Client-Serveur**
- Pages dynamiques (Javascript, PhP, Python...)
- Feuilles de style
- Données structurées (XML, JSON...) et services web

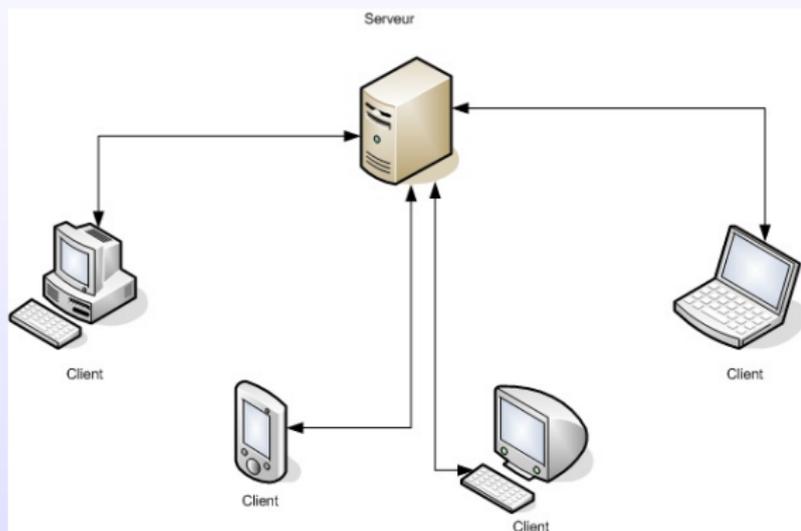
## développement des **standards du Web**

W3C : World Wide Web Consortium

WHATWG : Web Hypertext Application Technology Working Group

## Architecture Client-Serveur

- 1 **Client** : envoi d'une requête au serveur
- 2 **Serveur** : traitement de la requête
- 3 **Serveur** : envoi d'une réponse au client
- 4 **Client** : traitement de la réponse



## Protocole HTTP

- Niveau applications
- Système de requêtes / réponses
- Négociation de fonctionnalités
- Transfert bidirectionnel
- Support de cache
- Support des intermédiaires : proxy

## Méthodes usuelles

- **GET** : consultation d'informations
- **POST** : envoi de données au serveur
- **PUT** : mise à jour d'informations sur le serveur
- **DELETE** : suppression de données sur le serveur

# I - Introduction

## Identification des ressources

URI : Uniform Resource Identifier

- URL : Uniform Resource Locator :

`<scheme>://<net_loc>/<path>;<params>?<query>#<fragment>`

- URN : Uniform Resource Name

- <http://www.ac-poitiers.fr/>
- <http://www.w3.org/TR/xhtml1/#defs>
- <https://duckduckgo.com/>
- <ftp://mirrors.ircam.fr/pub/CTAN/>
- <mailto:sylvie.alayrangués@univ-poitiers.fr>
- <urn:isbn:1-58799-018-0>
- <urn:ietf:rfc:2396>

## Hyperlien

Un **hyperlien** permet d'accéder à une ressource via son URL

# I - Introduction

Pages (X)HTML statiques

Pages (X)HTML dynamiques

## Pages (X)HTML statiques

Contenu **figé**, écrit "en dur" par ses créateurs :

- en écrivant directement le texte structuré au format (X)HTML
- en utilisant un éditeur WYSIWYG
- en convertissant un document d'un autre format vers (X)HTML

## Pages (X)HTML dynamiques

## Pages (X)HTML statiques

Contenu **figé**, écrit "en dur" par ses créateurs :

- en écrivant directement le texte structuré au format (X)HTML
- en utilisant un éditeur WYSIWYG
- en convertissant un document d'un autre format vers (X)HTML

## Pages (X)HTML dynamiques

Tout ou partie du code (X)HTML est généré **à la volée**

- réagit aux actions de l'internaute
- utilise des langages de programmation (côté client et/ou côté serveur)
- souvent couplé avec une base de données côté serveur

# II - Page HTML

## II.1 - Qu'est-ce qu'une page HTML ?

### (X)HTML : (eXtended) HyperText Markup Language

Langage de description des pages web

- **HyperText** : langage permettant de référencer d'autres ressources via leur URL : pages web, images, autres types de documents. . .
- **Markup** : langage utilisant des balises pour structurer le document : titres, divisions, paragraphes, listes, tableaux. . .

(X)HTML n'est pas un langage de programmation

# II - Page HTML

## II.1 - Qu'est-ce qu'une page HTML ?

### Exemple d'un code HTML : structure générale

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

  +<head>

  +<body>

</html>
```

# II - Page HTML

## II.1 - Qu'est-ce qu'une page HTML ?

### Exemple d'un code HTML : en-tête du document

```
5 <head>
6 <title>SIF: Societe Informatique de France</title>
7 <meta http-equiv="content-language" content="fr" />
8 <meta http-equiv="content-type"
9   content="application/xhtml+xml; charset=utf-8"/>
10 <meta name="author" lang="fr" content="Michel Riveill" />
11 <meta name="description" lang="fr"
12   content="La Societe Informatique de France est un
13   espace de reflexion , de concertation sur les enjeux
14   de l' informatique ... " />
15 <meta name="keywords"
16   content="SIF, Informatique, Universite , CNRS, INRIA,
17   Enseignant, Chercheur, Industriel " />
18 <meta name="robot" content="follow, index, all " />
19 <meta name="revisit-after" content="15 days" />
20
21 <link rel="stylesheet" title="default" type="text/css"
22   media="screen" href="styles/base.css"/>
23 <link rel="stylesheet" title="default" type="text/css"
24   media="print" href="styles/baseprint.css"/>
25
26 <script type="application/javascript" src="general.js"> </script>
27 <script type="application/javascript" src="pageaccueil.js"> </script>
28
29 </head>
```

# II - Page HTML

## II.1 - Qu'est-ce qu'une page HTML ?

### Exemple d'un code HTML : corps du document

```
<body>  
  <div id="content">  
    +<p>  
    +<p>  
    +<p>  
    +<ul>  
    +<p>  
    +<p>  
  </div>  
<hr/>  
</body>
```

## II - Page HTML

### II.1 - Qu'est-ce qu'une page HTML ?

#### Exemple d'un code HTML : extrait de la division

```
52 <ul>
53 <li>
54     promouvoir l'informatique, servir et animer sa communauté
55     scientifique et technique,
56 </li>
57 <li>
58     contribuer à la culture des citoyens et à l'enseignement de
59     la discipline à tous les niveaux,
60 </li>
61 <li>
62     participer aux réflexions et aux initiatives sur la
63     formation et l'emploi des informaticiens ,
64 </li>
65 <li>
66     participer aux débats de société.
67 </li>
68 </ul>
```

# II - Page HTML

## II.1 - Qu'est-ce qu'une page HTML ?

```
<?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 xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="fr"
  lang="fr">
  <head>
    <title>Elementaire...</title>
    <meta http-equiv="content-type"
      content="application/xhtml+xml; charset=UTF-8"/>
  </head>
  <body>
    <div>
      <p>... mon cher
        <strong>
          Watson
        </strong> !
      </p>
      <p> Sir Arthur Conan Doyle disait
        <q>
          Nos idees doivent etre aussi vastes que la nature
          pour pouvoir en rendre compte.
        </q>
      </p>
    </div>
    <hr/>
  </body>
</html>
```

### Structure arborescente : Document Object Model



## II - Page HTML

### II.1 - Qu'est-ce qu'une page HTML ?

#### (X)HTML : éléments de structuration du corps du document

- Titres (de différentes importances)
- Blocs de texte quelconques (divisions, paragraphes),
- Blocs de citation,
- Blocs de code,
- Liste ordonnée, non ordonnée, liste de définitions,
- Tableau de données,
- Formulaires et éléments de formulaires,
- Portions de textes mises en valeurs,
- Liens hypertextes,
- Images,
- ...

#### Attributs associés aux éléments

### (X)HTML : éléments de structuration du corps du document

#### Attributs associés aux éléments

« méta-données » facultatives ou obligatoires

- définissables sur tous les éléments :
  - id : identifiant
  - class : classe d'appartenance
  - style
  - onclick, etc : gestion d'événements universels
  - ...
- spécifiques à un élément ou à un ensemble d'éléments :
  - src et alt pour les images,
  - action et method pour les formulaires,
  - href pour les liens
  - onload, etc : gestion d'événements spécifiques
  - ...

## II - Page HTML

### II.1 - Qu'est-ce qu'une page HTML ?

#### Construction de l'arbre DOM d'un document

- Parcours séquentiel du document HTML ;

# II - Page HTML

## II.1 - Qu'est-ce qu'une page HTML ?

### Construction de l'arbre DOM d'un document

- Parcours séquentiel du document HTML ;
- Pour chaque élément un nouveau nœud ;

### Construction de l'arbre DOM d'un document

- Parcours séquentiel du document HTML ;
- Pour chaque élément un nouveau nœud ;
- Si un élément est inclus dans un autre, son nœud est fils du nœud de l'autre :
  - balise incluse dans une autre,
  - texte inclus dans une balise. . .

#### Construction de l'arbre DOM d'un document

- Parcours séquentiel du document HTML ;
- Pour chaque élément un nouveau nœud ;
- Si un élément est inclus dans un autre, son nœud est fils du nœud de l'autre :
  - balise incluse dans une autre,
  - texte inclus dans une balise. . .
- Nœuds de même niveau ordonnés dans l'ordre de leur apparition dans le document.

# II - Page HTML

## II.2 - ... et une page HTML avec du style ?

### Qu'est-ce qu'un style ?

Un ensemble de règles :

**nom \_ propriété : valeur**

### Comment attribuer un style ?

Via des sélecteurs d'éléments fondés sur :

- le type d'éléments,
- la valeur de certains attributs,
- la position dans le DOM.

# II - Page HTML

## II.2 - ... et une page HTML avec du style ?

### Travail du navigateur ("User Agent")

- 1 Analyse du document (X)HTML  $\Rightarrow$  extraction de l'**arbre du document**
- 2 Identification du type de support  $\Rightarrow$  **media**
- 3 Récupération de toutes les **feuilles de style** associées
- 4 **Annotation des nœuds** de l'arbre avec une valeur par propriété de style
- 5 Création du **document formaté**
- 6 **Transfert** du document formaté au support.

## II - Page HTML

### II.3 - ... et une page HTML dynamique (côté client) ?

#### Langage de programmation

Javascript

#### Page (X)HTML entièrement créée côté serveur

⇒ une fois la page chargée côté client

- À chaque élément / nœud du DOM correspond un objet javascript,
- À chaque attribut d'un élément correspond une propriété de l'objet correspondant,
- Possibilité de brancher des fonctions Javascript sur les éléments pour répondre à des actions de l'utilisateur.

## II - Page HTML

### II.3 - ... et une page HTML dynamique (côté client) ?

Langage de programmation

Javascript

Page (X)HTML entièrement créée côté serveur

Page (X)HTML (partiellement) créée côté client

⇒ construction de la page de manière asynchrone

AJAX : Asynchronous JAVascript and XML

- Envoi de requêtes à un serveur pour récupérer des données structurées (e.g. XML, JSON)
- Construction d'une page web à partir de ces données

# II - Page HTML

## II.3 - ... et une page HTML dynamique (côté client) ?

### Langage de programmation

Javascript

Page (X)HTML entièrement créée côté serveur

Page (X)HTML (partiellement) créée côté client

### Actions via Javascript

- Modification de propriétés d'éléments
- Modification de la structure de la page
- Création d'une nouvelle page

## II - Page HTML

### II.3 - ... et une page HTML dynamique (côté client) ?

Navigation dans le DOM

Création de nouveau nœuds

Ajout / retrait de nœuds dans le DOM

## II - Page HTML

### II.3 - ...et une page HTML dynamique (côté client) ?

#### Navigation dans le DOM

- `var first_son = node.firstChild` :  
mémoire dans la variable `first_son` le premier fils  
représentant une balise de l'élément `node` ;
- `var next_son = son.nextElementSibling()` :

#### Création de nouveau nœuds

#### Ajout / retrait de nœuds dans le DOM

## II - Page HTML

### II.3 - ...et une page HTML dynamique (côté client)?

#### Navigation dans le DOM

- `var first_son = node.firstChild :`
- `var next_son = son.nextElementSibling() :`  
mémorise dans `next_son` le frère suivant le nœud `son`;

#### Création de nouveau nœuds

#### Ajout / retrait de nœuds dans le DOM

## II - Page HTML

### II.3 - ... et une page HTML dynamique (côté client) ?

#### Navigation dans le DOM

#### Création de nouveaux nœuds

- `var text_node = document.createTextNode("text")` : crée un nouveau nœud de texte contenant la chaîne de caractères passée en paramètre et mémorise le nœud dans la variable `text_node` ;
- `var element = document.createElement("nombalise")` ;
- `var new_node = node.cloneNode(true)` ; :

#### Ajout / retrait de nœuds dans le DOM

## II - Page HTML

### II.3 - ... et une page HTML dynamique (côté client) ?

#### Navigation dans le DOM

#### Création de nouveau nœuds

- `var text_node = document.createTextNode("text") :`
- `var element = document.createElement("nombalise");`  
crée un nouveau nœud correspondant à la balise dont le nom est passé en paramètre ;
- `var new_node = node.cloneNode(true); :`

#### Ajout / retrait de nœuds dans le DOM

## II - Page HTML

### II.3 - ... et une page HTML dynamique (côté client) ?

#### Navigation dans le DOM

#### Création de nouveaux nœuds

- `var text_node = document.createTextNode("text") :`
- `var element = document.createElement("nombalise");`
- `var new_node = node.cloneNode(true); :`  
clone le nœud `node` (avec ses descendants) et mémorise le nouveau nœud dans la variable `new_node` ;

#### Ajout / retrait de nœuds dans le DOM

# II - Page HTML

## II.3 - ... et une page HTML dynamique (côté client) ?

### Navigation dans le DOM

### Création de nouveau nœuds

### Ajout / retrait de nœuds dans le DOM

- `node.appendChild(fils);` :  
ajoute le nœud mémorisé dans la variable "fils" comme dernier enfant du nœud stocké dans node.
- `node.insertBefore(new_node, child);` :
- `node.remove(child);` :

# II - Page HTML

## II.3 - ... et une page HTML dynamique (côté client) ?

### Navigation dans le DOM

### Création de nouveau nœuds

### Ajout / retrait de nœuds dans le DOM

- `node.appendChild(fils); :`
- `node.insertBefore(new_node, child); :`  
insère le nœud stocké dans la variable `new_node` parmi les fils du nœud, `node`, juste avant le nœud `child`
- `node.remove(child); :`

# II - Page HTML

## II.3 - ... et une page HTML dynamique (côté client) ?

### Navigation dans le DOM

### Création de nouveau nœuds

### Ajout / retrait de nœuds dans le DOM

- `node.appendChild(fils); :`
- `node.insertBefore(new_node, child); :`
- `node.remove(child); :`  
retire du DOM le nœud stocké dans la variable `child`, enfant du nœud, `node`

### Structure « classique » d'un site web

Plusieurs dossiers contenant respectivement :

- des pages HTML,
- des feuilles de style :
  - ⇒ charte graphique générale du site,
  - ⇒ styles spécifiques à une ou plusieurs pages,
  - ⇒ styles alternatifs (accessibilité)
  - ⇒ styles pour différents supports (imprimante, écran)
- du code javascript (découpés en plusieurs fichiers)
- des fichiers multimédia (images, vidéos, sons)