Université de Msila

Faculté des mathématiques et de l'informatique

Département d'informatique

3ième année Licence ISIL

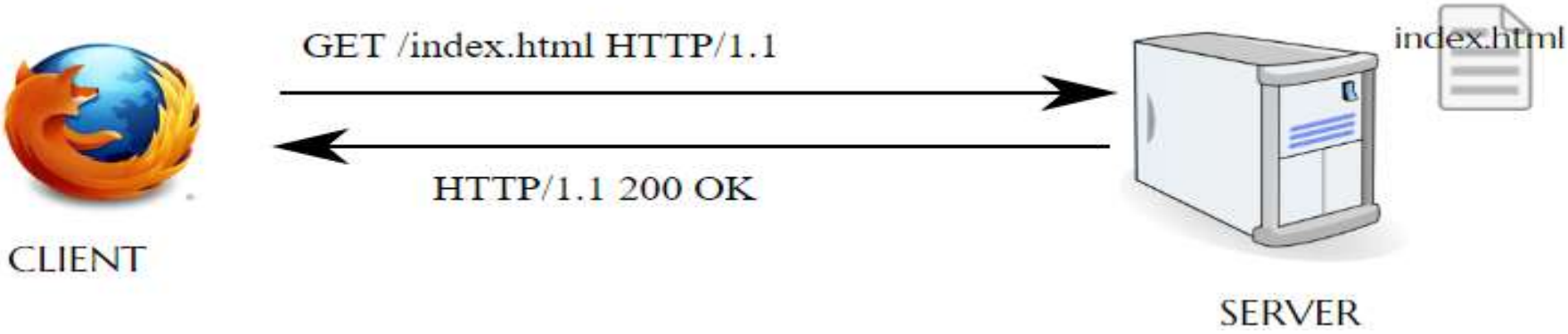
Rappels sur les sites Web

Présenté par : Meliouh.A

Comment ça marche

HTTP

- HTTP (Hypertext Transfer Protocol) est un protocole textuel, sans état, à requête-réponse destiné à servir des documents web.



- **Requête :** Le client (browser) demande à lire ou modifier un document (hypertexte, image, …)
- **Réponse :** Le serveur envoie une réponse (pas nécessairement le document).
- **Textuel :** Toutes les communications sont codées en ASCII (ce n'est plus le cas avec HTTP 2.0).
- **Sans état :** Le serveur ne se souvient pas du client entre deux requêtes.

La requête HTTP


```
POST /document.html HTTP/1.1
Host: www.example.com
User-Agent: Mosaic/2.1
Cookie: sessionid=aa03x;
Content-Length: 10

1234567890
```

- Action,
  - Méthode (Head, GET, POST)
  - Ressource (adresse du document Web)
  - Protocole: HTTP/1.0 , HTTP/1.1 , HTTP/2.0
- Entêtes (seulement Host est obligatoire),
  - Servent à envoyer des meta-données au serveur.
- Une ligne vide ,
- Corps du message (Optionnel).

La Réponse HTTP

HTTP/1.1 200 OK
Date: Tue, 24 Jan 2017 18:34:40 GMT
Server: Apache/2.2.21 (Debian)
Last-Modified: Fri, 10 Dec 2010 14:10:25 GMT
Content-Length: 53
Content-Type: text/html
Set-Cookie: sessionid=jkWXBR; expires=Wed, 25-Jul-2012 17:09:10 GMT; path=/; domain=.google.fr; HttpOnly

<html><head></head><body><h1>Hello world!**</h1></html>**

- Status line,
- Entêtes,
- Une ligne vide (attention: <CR><LF>),
- Contenu (Optionnel).

La Réponse HTTP

HTTP/1.1 200 OK

Status line,

Protocole + code d'état + message

Les codes d'état décrivent le résultat de la requête. Les plus fréquents :

200 OK Le document a été trouvé et envoyé au client.

301 MOVED PERMANENTLY Redirection permanente (nécessite Location).

302 FOUND

303 SEE OTHER

307 TEMPORARY REDIRECT Différents types de redirection (nécessitent Location).

400 BAD REQUEST Le client a envoyé une requête mal formatée.

403 FORBIDDEN Le document n'est pas accessible.

404 NOT FOUND Le document est inconnu au serveur.

410 GONE Le document n'existe plus.

500 INTERNAL SERVER ERROR Erreur sur le serveur.

503 SERVICE UNAVAILABLE Le serveur est momentanément indisponible.

HTML = HyperText Markup Language

HTML est un *langage de balisage* (*markup language*), inspiré par SGML, pour l'écriture de documents Hypertexte

Balises HTML

```
<tag>
 Mon contenu <tag>plus de contenu</tag>
</tag>
```

Les Balises (tags) délimitent du contenu textuel :

- À chaque balise ouvrante <tag1> correspond une balise fermante </tag1>.
- Une balise peut en contenir d'autres, proprement imbriquées.
- Certaines balises n'ont pas de contenu. Dans ce cas <tag></tag> peut être raccourci en <tag/>

Attributs

- Les balises peuvent avoir des **attributs**, dans la balise ouvrante.

```
<tag attribut1="valeur 1"
     attribut2='valeur 2'
     attribut3=valeur3>
  Contenu
</tag>
```

- Les valeurs des attributs sont contenues entre guillemets simples, doubles ou aucun guillemet ; dans ce dernier cas (à éviter) elles ne doivent pas contenir d'espace.
- Certains attributs sont obligatoires pour certains tags.

Commentaires

- Les commentaires sont écrits entre <!-- et -->

```
<!-- Ceci est un commentaire, ce sera ignoré -->
<tag>Ceci sera interprété par le browser</tag>
```

La structure du document

- Il y a un nombre limité de balises, l'utilisateur **ne peut pas** en inventer.
- Tout document HTML **doit** avoir cette forme

```
<html>
  <head>
    <!--
     Le head contient toutes les informations
     sur le document qui ne sont pas du contenu
     -->
    <title>Un titre</title>
  </head>
  <body>
    <!-- Le body contient le vrai Hypertexte -->
  </body>
</html>
```

Ce qui va dans le <head>

- Le **<head>** contient tout ce qui concerne le document, mais qui n'en fait pas partie. Voici quelques unes de ses balises plus importantes.
  - **<title>:** Le titre du document. Obligatoire.
  - **<script>:** Code pour le scripting côté client (JavaScript, VBScript, etc.).
  - **<style>:** Directives d'affichage (CSS, etc.).
  - **<meta>:** Meta-informations sur le document (langue, encodage, etc.).
  - **<link>:** Documents reliés (flux de news, favicons, etc.)

Ce qui va dans le <body>

Le **<body>** contient le vrai contenu. Voici une petite sélection de balises:

- **<section>, <nav>:** Structure du document.
- **<header>, <footer>, <aside>, <address>:** Structure d'une section.
- **<p>, <h1>, …, <h6>:** Un paragraphe de texte, un titre de premier niveau, …, un titre de sixième niveau.
- **<a>**: Une ancre, c.-à-d. un lien vers d'autres contenus.
- **<img>, <audio>, <video>, <object>, <svg>:** Inclusion de médias
- **<form>, <input>,** …: Interaction avec l'utilisateur.
- **<table>, <tr>, <td>,** …: Tableaux.
- **, :** Texte présente différemment (par ex., gras ou italique)
- ** , :** Balises avec aucune signification (importantes pour faire le lien avec style et scripts).

Attributs

Les attributs jouent plusieurs rôles. Voici les plus communs:

- **id:** Assigne un identifiant à une balise. Doit être unique.
- **class:** Assigne une balise à une classe (pour regroupement logique).
- **src:** Indique où trouver les ressources externes.
- **href:** HyperReference. Utilisé pour faire des liens aux ressources externes.
- **style:** Pour ajouter des directives de style.
- **title:** Donne plus d'informations sur une balise.
- **onclick, onload, onmouseover, …:** ***Event hooks*** pour les scripts.
- **data-*:** Attributs définissables par les utilisateurs (depuis HTML5).

-
  <li data-animal-type="bird">Owl
  <li data-animal-type="fish">Salmon
  <li data-animal-type="spider">Tarantula

Suivez les standards

- En s'efforçant de rendre plus facile ou plus puissant la programmation web, les développeurs des browsers ont introduit au fil du temps des tonnes de balises et règles d'interprétation *non standard*.
- Ceci a finalement amené à des **mauvaises pratiques** en programmation web, et à des **incompatibilités**.
- Le W3C essaye d'imposer les standards du Web, aussi à travers ses outils de **validation**.
- **Servez-vous en :**
  - http://validator.w3.org/
  - http://jigsaw.w3.org/css-validator/
  - http://validator.w3.org/mobile/
  - http://validator.w3.org/checklink

Déclarer le langage

l faut déclarer le langage et la variante dans laquelle vous écrivez vos documents. Un document HTML5 proprement déclaré commence comme ceci (façon préférée dans ce cours)

```
<!DOCTYPE html>
<html>
    <head>...</head>
    <body>...</body>
</html>
```

Ou comme ceci (avec l'encodage XHTML)

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>...</head>
    <body>...</body>
</html>
```

Attention à l'imbrication !

Bon

```
<p>
  <strong>Gras <em>et même italique</em></strong>
</p>
```

Mal

```
<p>
  <strong>Gras <em>et même italique</strong></em>
</p>
```

Bonne pratique : déclarer l'encodage

Par exemple (préférez l'encodage Unicode)

```
<head>
  <meta charset="utf-8" />
  ...
</head>
```

ou comme ceci

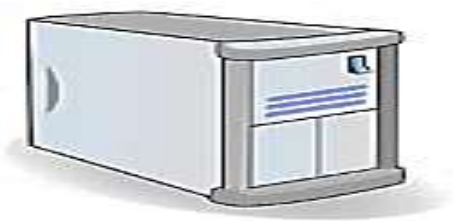
```
<head>
  <meta charset="iso-8859-1" />
  ...
</head>
```

Formulaires HTML


```
<form method="POST" action="http://www.captcha.net/user">
  First name:  <input type="text" name="first">  <br>
  Last name:  <input type="text" name="last">  <br>
  Gender: Male  <input type="radio" name="sex" value="M">
          Female  <input type="radio" name="sex" value="F"><br>
  e-mail:  <input type="email" name="email">  <br>
  <input type="submit" value="Subscribe">
</form>
```

First name: Alan
Last name: Turing
Gender: Male ◉ Female ○
e-mail: alan@gchq.gov.uk
Subscribe

```
POST /user HTTP/1.1
Host: www.captcha.net
...
first=Alan&
last=Turing&
sex=M&
email=alan@gchq.gov.uk
```



www.captcha.net

Contrôles des formulaires

La balise **<input>** représente presque tous les contrôles. Le choix se fait à travers l'attribut **type**.

- **Champs de texte :** <input type="**text**" name="clef">

tapez quelque chose

- **Boutons radio:**

<input type="**radio**" name="choice" value="choice-1">A

<input type="**radio**" name="choice" value="choice-2">B

A B

- **Checkbox:**

<input type="**checkbox**" name="check1" value="check-1"> C

<input type="**checkbox**" name="check2" value="check-2">D

C D

- **Password** <input type="**password**" name="mot_de_passe">

••••••••••

- **Fichier** <input type="**file**" name="fichier">

Choisissez un fichier Aucun fichier choisi

- **Email** (depuis HTML5, vérifie qu'il y a un @)

<input type="**email**" name="courriel">

- **Submit** <input type="**submit**" value="Send data">

Send data

Contrôles des formulaires (suite)

- **Button:** <input type="button" value="Click me!">
- **Image:** <input type="image" src="like.svg" alt="Like!">

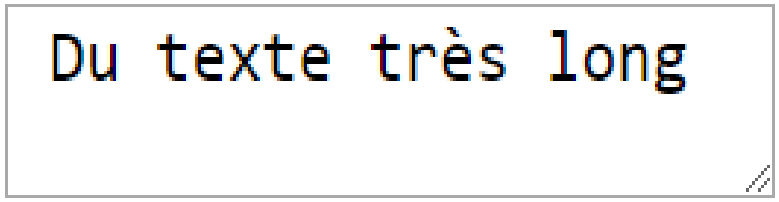
Remarque:

D'autres nouveaux types sont définis dans HTML5 (pour la plus part, pas encore bien supportés): **date**, **time**, **number**, **range**, **color**, **tel**, **url**, …

Autres contrôles

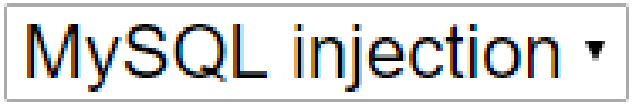
- **Text area:**

```
<textarea name="texte_long">
    Du texte très long
</textarea>
```



- **Selection list:**

```
<select name="liste">
    <option value="M">MySQL injection</option>
    <option value="X">XSS</option>
    <option value="C">CSRF</option>
</select>
```



Validation des formulaires

- Attribut **required**: prévient si pas rempli

```
<input type="text" required>
```

- Attribut **pattern**: compare l'entrée à une regexp

```
<input type="text" pattern="[0-9]{6}">
```

- Attribut **placeholder**: donne une suggestion

```
<input type="text"
placeholder="tapez quelque chose">
```

- Attribut **novalidate**: désactive toutes les validations précédentes.
- Des validations plus compliquées doivent être faites en **Javascript**.

Les Feuilles de style.

- **CSS = Cascading Style Sheets:** un langage pour l'expression de directives d'affichage.
- **Cascading** veut dire que plusieurs feuilles de style peuvent s'appliquer à un document, le résultat est calculé d'après des règles de précédence en cascade.

```
p.lead {
  font-weight: bold;
  font-family: "Gill Sans MT", GillSans, sans-serif;
  padding: 2pt;
}

p.lead:first-letter {
  font-size: 200%;
}

p.lead em {
  font-variant: small-caps;
  font-style: normal;
}
```

Lorem ispsum *DOLOR SIT* amet

Les Feuilles de style

- **La syntaxe CSS**
- **Règles CSS**

```
selector {property: value; property: value; ...}
```

- **Commentaires**

```
/* Ceci est la seul façon de faire des commentaires en CSS
   (en effet, // n'introduit pas un commentaire)       */
```

- **At-rules**

```
@import "unautrestyle.css"; /* Importe dans la feuille courante */
@media screen;              /* S'applique seulement à un affichage
                               sur écran (par ex., pas à l'impression) */
```

Les Feuilles de style

- **Où va le style ?**

```
<html>
  <head>
    <!-- Ceci s'applique à tout le document -->
    <link rel='stylesheet' href='sheet.css' type='text/css' />
    <style>
      body {font-family: Arial;}
    </style>
  </head>
  <body>
    <div>
      <!-- Ceci s'applique seuelement dans ce div -->
      <style scoped>
        p {color:blue;}
      </style>
      <!-- Ceci s'applique seulement à ce paragraphe
           La syntaxe CSS y est limitée -->
      <p style="font-weight:bold">...</p>
    </div>
  </body>
</html>
```

- **Le document et le style**

```
<style>
  .lerouge {color:red;}
  div.lerouge {background-color:yellow;}
  #lenoir {color:black;}
  div p {font-style:italic;}
</style>
...
<p class="lerouge">Premier</p>
<div class="lerouge">
  <p id="lenoir">Deuxième</p>
  <p>Troisième</p>
</div>
```

Premier

Deuxième

Troisième

Référence

Cours Applications web et sécurité

http://defeo.lu/aws/