



Applications Web

Cours 1: Introduction et rappels



Plan

- Web statique vs. Web dynamique
- Principe des applications web
- Protocole HTTP
- Bibliographie



Terminologie: serveur

- **Serveur**: un ordinateur disposant d'un certain nombre de ressources qu'il met à disposition d'autres ordinateurs (**clients**) via le réseau.
- Types de serveurs:
 - serveur web
 - serveur d'application
 - ...



Terminologie: serveur web

- Programme s'exécutant sur une machine reliée à internet
- Protocole HTTP: répond aux requêtes des clients (**navigateur web**) et les traite
- Retourne des pages **HTML**



Web statique

- Web 1.0
 - HTML
 - CSS



HTML

- Langage des pages web




```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<title>Google</title>

</head>
<body bgcolor=#ffffff text=#000000 link=#0000cc vlink=#551a8b alink=#ff0000 topmargin=3 marginheight=3>

<center>
<br clear=all id=lcpd>

<br><br>
<form action="/search" name=f>
<table cellpadding=0 cellspacing=0>
  <tr valign=top>
    <td width=25%>&nbsp;</td>
    <td align=center nowrap>
      <input name=hl type=hidden value=fr> <input maxlength=2048 name=q size=55 title="Recherche Google" value="">
      <br>
      <input name=btnG type=submit value="Recherche Google"> <input name=btnI type=submit value="J'ai de la chance"></td>
    <td nowrap width=25%>
      <font size=-2>&nbsp;&nbsp;&nbsp;<a href=/advanced_search?hl=fr>Recherche avancée</a>
      <br>&nbsp;&nbsp;&nbsp;<a href=/preferences?hl=fr>Préférences</a>
      <br>&nbsp;&nbsp;&nbsp;<a href=/language_tools?hl=fr>Outils linguistiques</a></font>
    </td>
  </tr>
  <tr>
    <td align=center colspan=3>
      <font size=-1><span style="text-align:left">Rechercher dans :
        <input id=all type=radio name=meta value="" checked><label for=all> Web </label>
        <input id=lgr type=radio name=meta value="lr=lang_fr"><label for=lgr> Pages francophones </label>
        <input id=cty type=radio name=meta value="cr=countryFR"><label for=cty> Pages : France </label>
      </span></font></td>
  </tr>
</table>
</form>
<br><br>
<font size=-1><a href="/intl/fr/ads/">Programmes de publicité</a> - <a href="/services/">Solutions d'entreprise</a> - <a href="/intl/fr/
/about.html">À propos de Google</a> - <a href=http://www.google.com/ncr>Google.com in English</a></font><p><font size=-2>&copy;2008 Goo
gle</font></p>
</center>

</body>
</html>
```

En-tête

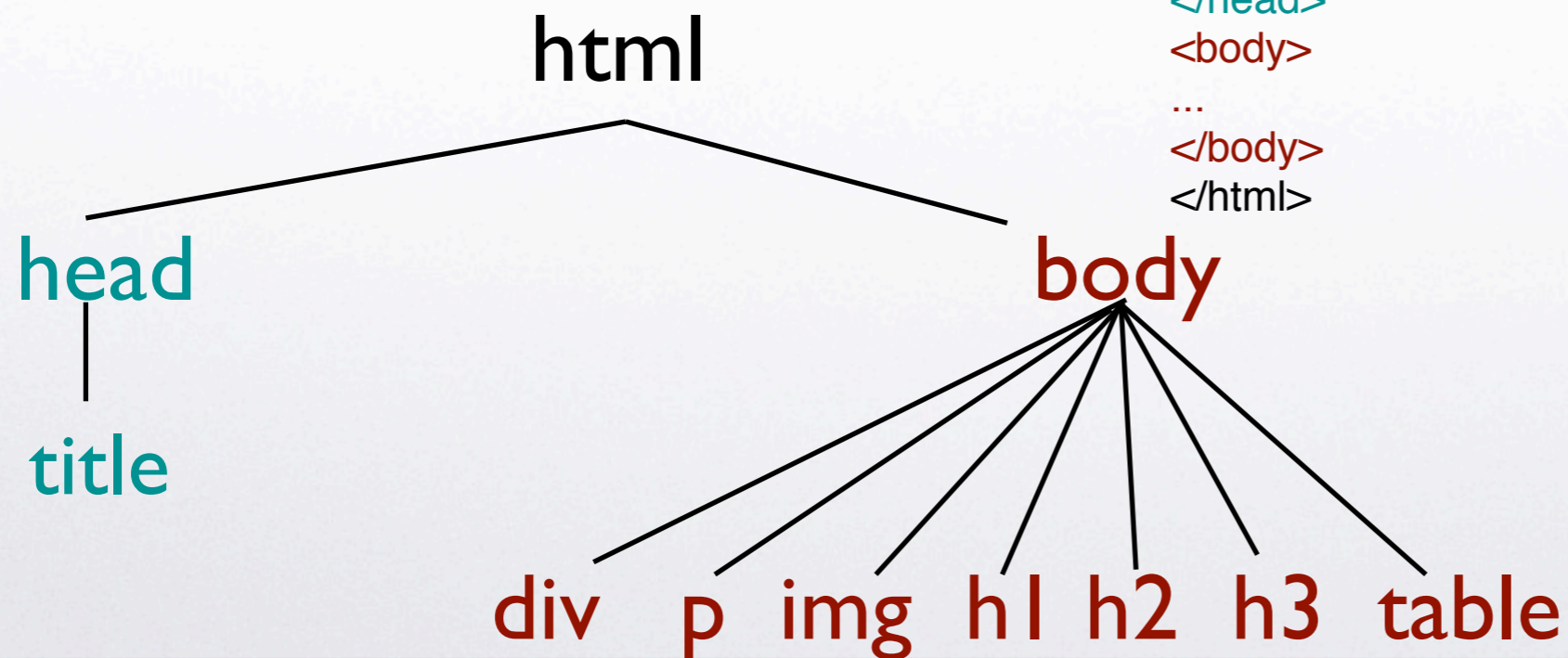
Corps



HTML

- Structure d'une page HTML

```
<?xml version="1.0" encoding="iso-8859-1"?>  
<!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">  
  <head>  
    <title>Page 1</title>  
    ...  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```





HTML

- Une page HTML est une ensemble d'éléments,
- Un élément est composé de :
 - balise ouvrante + contenu + balise fermante
 - `<title>Titre de page</title>`
- Les balises peuvent avoir des attributs:
 - ``



HTML

- Editable avec un éditeur de texte :-) mais il existe des éditeurs spécialisés:
 - Amaya, DreamWeaver, ...
- HTML est un standard (<http://www.w3.org/TR/xhtml111>) les pages doivent être valides:
 - Valideurs en ligne: W3C, Valet, ...
 - <http://validator.w3.org>, <http://valet.webthing.com>
 - Librairies de validation/nettoyage: Tidy, OpenSP, ...
 - <http://tidy.sourceforge.net>





CSS

- Cascading Style Sheets
 - présentation des pages
 - un style par défaut dans le navigateur
 - feuilles de styles, dans des fichiers CSS

home.css

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<link rel="stylesheet" type="text/css" href="home.css" />

<style type="text/css">
/**/
  div.spot-image img {
    margin-bottom: 20px;
  }
/**/
</style>
...
</head>
```

```
html, body {
  background: #fff;
  color: #000;
}
html, body, h2, h3, h4, div, p, ul, li, input {
  font-family: Arial, Helvetica, sans-serif;
}
a:link {
  background: #fff;
  color: #037;
}
a:visited {
  background: #fff;
  color: #636;
}
a:active {
  background: #ffc;
  color: #900;
  text-decoration: none;
}
a img { color: #fff; } /* hide the border in Netscape 4 */
@media all { /* hide from Netscape 4 */
  a img { color: inherit; /* undo the rule above */
  border: none; /* hide from Netscape 4 */
}
}
```

CSS

texte noir
sur fond blanc

définit les fonts
des éléments

le texte de #logo
et de #slogan est
centré

le texte de
l'élément #slogan
est bleu sur fond
blanc

```
html, body {  
  background: #fff;  
  color: #000;  
}  
  
html, body, h2, h3, h4, div, p, ul, li, input {  
  font-family: Arial, Helvetica, sans-serif;  
}  
  
#logo, #slogan {  
  text-align: center;  
  margin: 0;  
  padding: 0;  
  margin-bottom: 0.3em;  
}  
  
#slogan {  
  background: #fff;  
  color: #4d8cba;  
  font-family: Times, "Times New Roman", serif;  
  font-style: italic;  
  font-weight: bold;  
  margin-top: -6px;  
}
```



Leading the Web to Its Full Potential...

[Activities](#) | [Technical Reports](#) | [Site Index](#) | [New Visitors](#) | [About W3C](#) | [Join W3C](#) | [Contact W3C](#)



CSS

- **Avantages des styles externes:**
 - uniformité sur tout le site
 - changement de présentation facile
 - adaptation facile (impression, accessibilité, ..)



CSS

- Editable avec un éditeur de texte,
 - Editeurs spécialisés: *CSSEd*, *DreamWeaver*, ...
 - Validation: <http://validator.w3.org>
 - Extensions firefox:
 - *CSSViewer*,
 - *Web Developer Toolbar*





Web dynamique I

- Exécution par le client de scripts ou d'applications:
 - Scripts
 - Applets



Scripts

- Exécutés par le navigateur
 - les scripts ont accès (lecture et écriture) au contenu de la page qui les appelle.
 - MAIS pas aux données stockées sur le serveur, et ils ont des restrictions de sécurité (pas d'accès aux fichiers de la machine locale ...)
 - On utilise les scripts exécutés par le navigateur pour traiter des actions utilisateurs : click où passage sur une zone de la page ..., quand la réaction demandée ne nécessite pas d'informations stockées sur le serveur.
 - *Exemples* : convertisseurs, calculette; vérifications de base du remplissage d'un formulaire...



JavaScript

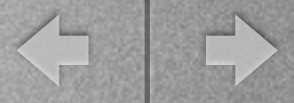
- Langage de script avec une syntaxe proche de java
- Exemple:

Convertisseur Francs
Euros

Francs 98.39

Euros 15

conversion



JavaScript

Définition des fonctions javascript (balise <script>)

Le formulaire est composé de 2 champs *fr* et *eu*, on remplit un champs et on soumet (pas d'action car on ne fait pas

A l'événement `onClick` la fonction `cfe` est appelée

```
<html>
<head>
  <title>Page 1</title>
  <link rel="stylesheet" media="screen" type="text/css"
    title="Design1" href="examRatMar06.css" />
  <script language="javascript">
    function cfe(lef) { // .... on le verra tout a l'heure ]
  </script>
</head>
<body>
<h1> Convertisseur Francs Euros </h1>
<form class=regle method="get">
  <p>
  Francs
  <input type="text" name="fr" size="20" />
  </p>
  <p>
  Euros
  <input type="text" name="eu" size="20" />
  </p>
  <input type="button" onClick="cfe(this.form)"
    value=" conversion " />
</form>
</body>
</html>
```




JavaScript

```
<script language="javascript">
function cfe(lef) {
  tdc = 6.55957;
  var ffr = lef.fr.value;
  var eeu = lef.eu.value;
  if (ffr.length>0) {
    eeu = ffr/tdc;
    lef.eu.value = eeu.toFixed(2);
  }
  else {
    ffr = eeu*tdc;
    lef.fr.value = ffr.toFixed(2);
  }
}
</script>
```

la fonction appelée par le bouton *conversion*

ffr: valeur du champs fr
eeu: valeur dur champs eu

calcul de la valeur en eu si le champs fr est non vide



Applets

- Applications Java exécutées par le navigateur
 - restrictions équivalentes aux scripts
 - pas d'accès au serveur
 - pas d'accès aux fichiers sur la machine cliente
 - nécessite des plug-ins JAVA
 - MAIS possibilité d'écrire des applications complexes, ou d'utiliser des applications existantes



Applet

- Le serveur rend disponible:
 - une classe (ou un ensemble de classes) JAVA.
 - une page HTML référénçant la classe (balise <applet>)
 - *Exemple:*

SimpleApplet.html

```
<html>
<head>
<title> A Simple Program </title>
</head>
<body>
<p>
The box is displayed using a simple applet : </p>
<applet code="SimpleApplet.class" width="150" height="50" />
</applet>
</body>
</html>
```

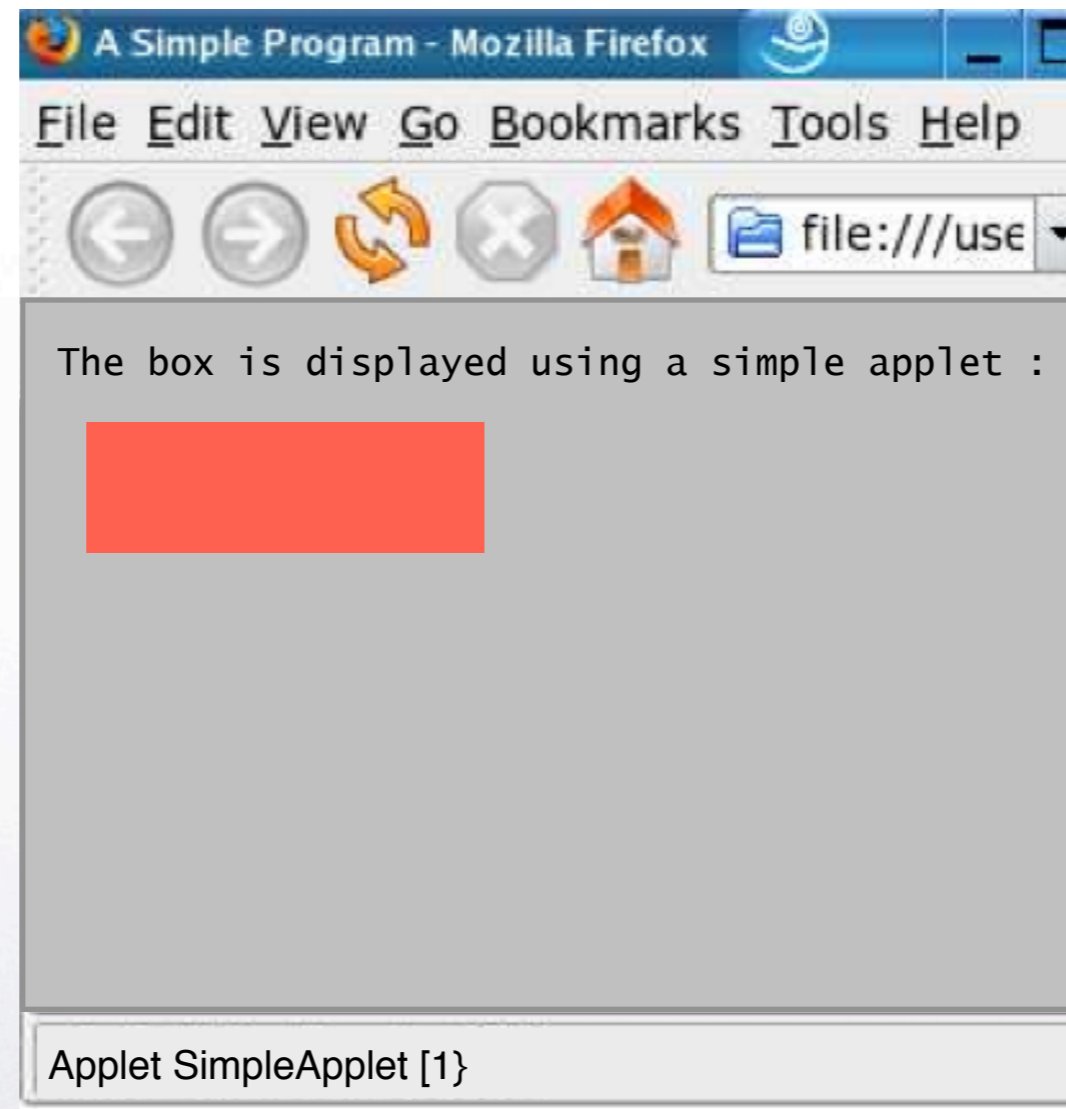
SimpleApplet.java

```
import java.applet.Applet;
import java.awt.Color;

public class SimpleApplet extends Applet{
    public void init(){
        this.setBackground(Color.RED);
    }
}
```



Applet





Web dynamique II

- Exécution par le serveur
 - Orienté présentation
 - pages statiques
 - pages dynamiques
 - Orienté service



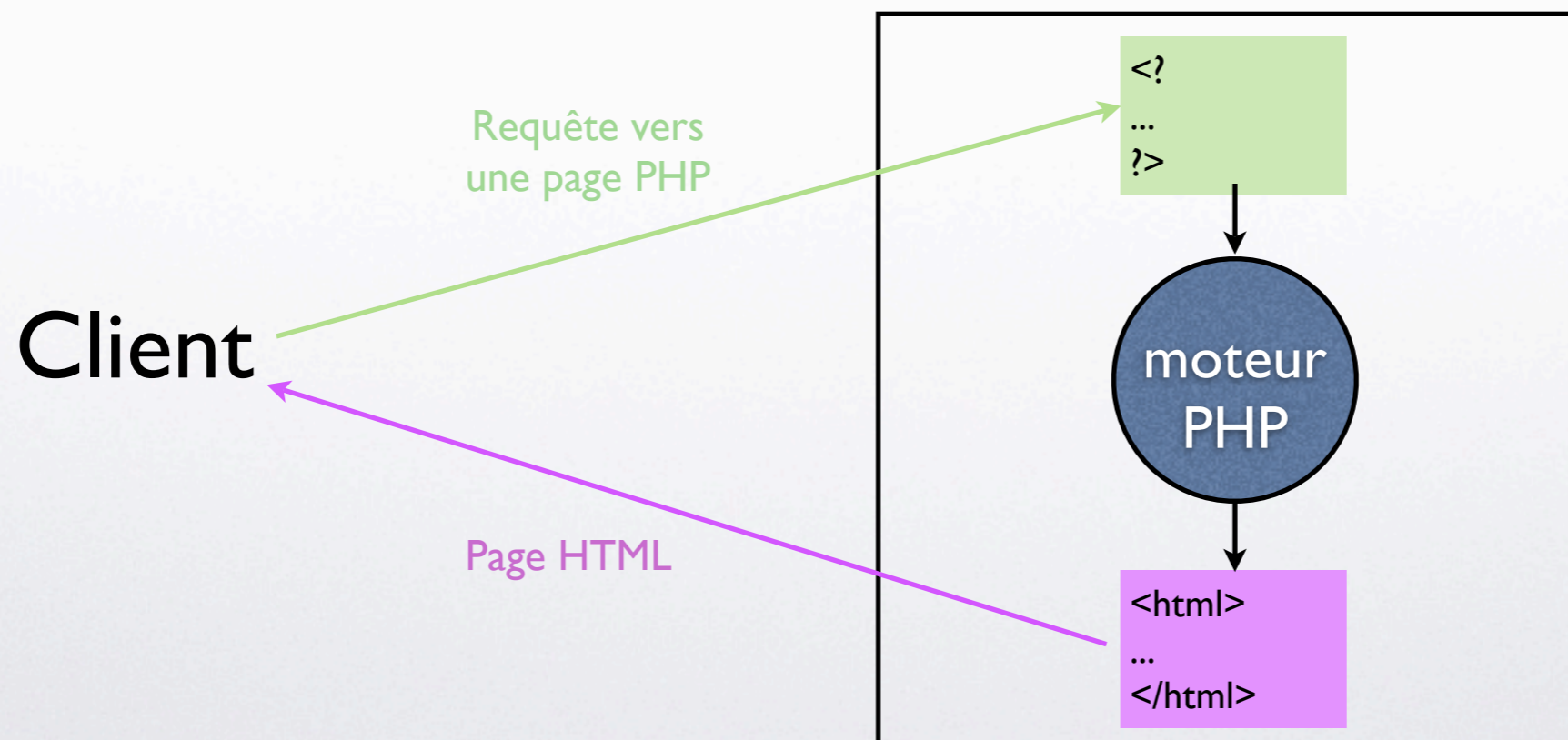
Orienté présentation

- Il s'agit d'applications qui génèrent des pages contenant différents types de langage: html, sgml, xml, ... en réponse à des requêtes.
- Les pages générées peuvent être *statiques* (pages html)
- Les pages générées peuvent être *dynamiques*: pages contenant du code exécuté sur le serveur



Orienté présentation

- URL pointant sur une page contenant des scripts PHP (ASP) ayant pour rôle de construire "à la volée" des pages HTML. La page pointée fait appel à un interpréteur PHP qui convertit son contenu.





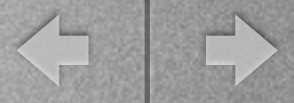
Orienté service

- Une application Web orientée service implémente un point d'accès à un service via le Web. Ce service est souvent invoqué par une application orientée présentation
- Ce sont typiquement des services métiers. On parle de transactions B-to-B



HTTP

- Le protocole
- Les méthodes de base
- Gestion des sessions
- Clients et serveurs
- Bibliographie

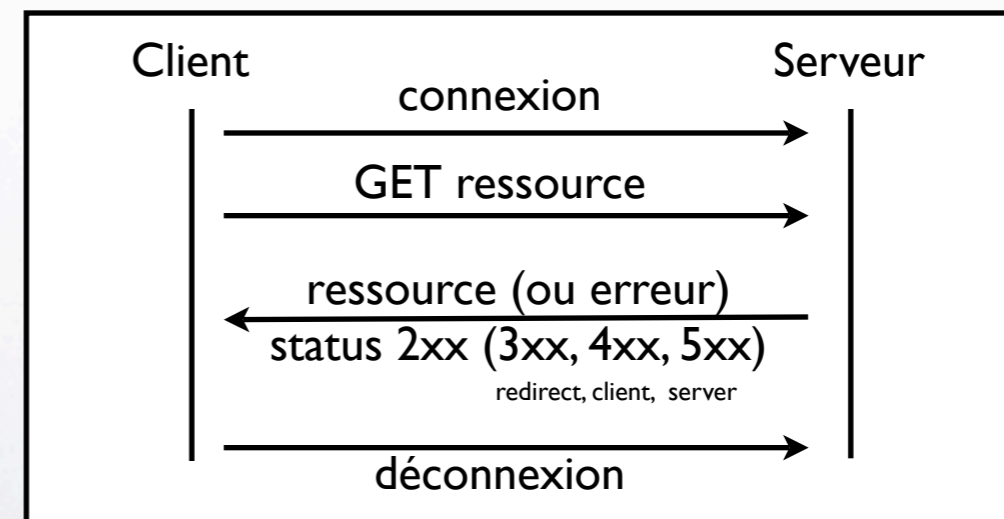


Le protocole HTTP

- HTTP: HyperText Transfert Protocol
 - RFC 1945, RFC 2616
 - protocole de rapatriement de ressources et de soumission de formulaires

- Principe (HTTP/1.0)

- connexion,
- demande de ressource (GET),
- renvoi de la ressource ou d'une erreur
- déconnexion



- Le dialogue peut être plus complexe:
 - authentification, optimisations (KeepAlive en HTTP/1.1)



Dialogue HTTP

- Protocole en mode texte

- telnet www.unice.fr 80

- Types de dialogue

- Récupération de ressources ➡ GET
- Soumission de formulaires ➡ GET ou POST
- Envoi de ressources ➡ PUT, DELETE, LINK, UNLINK
- Gestion de cache/proxy ➡ HEAD



Format de requêtes

- Du client au serveur:

Méthode <URI> HTTP/<version>

[<champs>: <valeur>]

[<tab> suite de la valeur si > 1024]

ligne blanche

[corps de la requête (POST)]

- Exemple de requête GET

GET /index.html HTTP/1.0

Accept: www/source

Accept: text/html

Host: www.w3.org

User-Agent: Mozilla/5.0 (X11; U; Linux i686; fr-FR) Firefox/2.0

- Exemple de requête POST

POST /form HTTP/1.0

Accept: www/source

Accept: text/html

User-Agent: Mozilla/5.0 (X11; U; Linux i686; fr-FR) Firefox/2.0

Content-Length: 24

name1=value1&

name2=value2



Méthodes de requête (i)

- **GET**
 - obtenir les informations et les données référencées par l'URI
- **HEAD**
 - obtenir les informations concernant l'URI
- **POST**
 - envoie de données au serveur (les données sont contenues dans la requête après l'en-tête et une ligne vide)
- **PUT**
 - enregistrement du corps de la requête à l'URI
- **DELETE**
 - suppression des données référencées par l'URI



Méthodes de requête (ii)

- **OPTIONS**
 - options de communication disponibles
- **TRACE**
 - renvoie la requête telle quelle (debug)
- **CONNECT**
- **Extensions WebDAV**
 - PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK
- **Extensions HTTP/U et HTTP/MU**
 - NOTIFY, ...(UPnP)



Format de réponse

- Du serveur vers le client

HTTP/<Version> <Status> <Commentaire Status>

Content-Type: <Type MIME du contenu>

[<Champ>: <Valeur>]

[<tab> suite de la valeur si >1024]

Ligne vide

Données

- Exemple de réponse

HTTP/1.1 200 OK

Content-Type: text/html

Content-Length: 2345

Date: Sat, 23 Feb 2008 15:36:29 GMT

Server: Apache/2.2.4 (Ubuntu)

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

<html><head> ...



Status des réponses

Code de réponse donné par le serveur au client

- **100-199 Informationnel**

- 100 : Continue (le client peut envoyer la suite de la requête), ...

- **200-299 Succès de la requête client**

- 200: OK, 201: Created, 204 : No Content, ...

- **300-399 Redirection de la Requête client**

- 301: Redirection, 302: Found, 304: Not Modified, 305 : Use Proxy, ...

- **400-499 Requête client incomplète**

- 400: Bad Request, 401: Unauthorized, 403: Forbidden, 404: Not Found

- **500-599 Erreur Serveur**

- 500: Server Error, 501: Not Implemented, 502: Bad Gateway, 503: Out Of Resources(Service Unavailable)



Entêtes HTTP

- **Général**
 - communes au client et au serveur
 - Cache-Control, Connection, Date, MIME-Version, Pragma, Transfert-Encoding, Upgrade, Via
- **Requête client**
 - formats des données, paramètres
 - Accept, Accept-Language, Authorization, Cookie, Host, Referer, User-Agent, ...
- **Réponse serveur**
 - informations du serveur
 - Accept-Range, Age, Public, Retry-After, Server, Set-Cookie, WWW-Authenticate...
- **Entité**
 - informations sur les données
 - Allow, Content-Base, Last-Modified, Content-Length, Content-Type, Refresh, ...



Exemples

- Récupération d'une page (GET)

```
GET /index.html HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/jpg
User-Agent: Toto/0.0

<enter>
```

<- Requête

Réponse ->

```
HTTP/1.1 200 OK
Date: Sat, 23 Feb 2008 16:17:07 GMT
Server: Apache/2.2.4 (Ubuntu) DAV/2 PHP/5.2.3-1ubuntu6.3
Last-Modified: Sat, 19 Jan 2008 08:02:12 GMT
ETag: "42de90-3123-cafd0900"
Accept-Ranges: bytes
Content-Length: 12579
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//
EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>
```




Exemples

- Soumission d'un formulaire (GET)

```
GET /form?cursus=miage&cours=appliweb HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/jpg

<enter>
```

<- Requête

Réponse ->

```
HTTP/1.1 200 OK
Date: Sat, 23 Feb 2008 16:19:07 GMT
Server: Apache/2.2.4 (Ubuntu) DAV/2 PHP/5.2.3-1ubuntu6.3
Last-Modified: Mon, 21 Jan 2008 07:12:12 GMT
Accept-Ranges: bytes
Content-Length: 1579
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//
EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>
```



Exemples

- Soumission d'un formulaire (POST)

```
POST /form HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/jpg
```

```
cursus=miage&
cours=appliweb
```

<- Requête

Réponse ->

```
HTTP/1.1 200 OK
Date: Sat, 23 Feb 2008 16:19:07 GMT
Server: Apache/2.2.4 (Ubuntu) DAV/2 PHP/5.2.3-1ubuntu6.3
Last-Modified: Mon, 21 Jan 2008 07:12:12 GMT
Accept-Ranges: bytes
Content-Length: 1579
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//
EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>
```




Encodage des param.

- Paramètres au format `x-www-form-urlencoded`
 - *i.e.* sur 7-bits et pas de caractères spéciaux
- Espace : `+`
- Caractères spéciaux et accentués : `%code-ascii` (`à` → `%e0`)
- **Exemple:** pages en cache sur google (recherche: *MIAGE M1 Nice*)
<http://209.85.129.104/search?q=cache:e6Db8a3z2MMJ:portail.unice.fr/jahia/1313+MIAGE+M1+Nice&hl=fr&ct=clnk&cd=1&gl=fr&client=firefox-a>
- Bibliothèque d'encodage/décodage disponibles
 - En Java:
 - `java.net.URLEncoder`
 - `java.net.URLDecoder`



Sessions

- Dans une application web, le client a besoin de faire différentes requêtes au serveur:
 - Webmail
 - Site de e-commerce
- Une **session** est l'exécution d'une application web pour un utilisateur donné. C'est un espace de travail limité dans le temps et associé à un espace de stockage d'informations propre à l'utilisateur.



Sessions

- HTTP ne gère pas la notion de sessions:
- Pas d'informations sur les requêtes précédentes d'un client
- Pas pratique pour nombre d'applications ayant besoin de gérer des sessions.

Comment implanter les sessions?



Sessions

- **Création par le serveur d'identifiant de session fourni au client, qui le renvoie à chaque requête**
 - un identifiant de session défini l'état et la durée de validité d'une session
- **Plusieurs méthodes pour les échanges d'identifiants entre le client et le serveur:**
 - ré-écriture d'URL
 - input HIDDEN dans les formulaires
 - cookies



Sessions: ré-écriture d'URL

- L'identifiant de session est encodé dans l'URL des pages retournées par le serveur:

URL originale **identifiant de session**
`http://localhost/webapp/example;jsessionid=323aed39902`

- Différents codage possibles:

<code>http://server/servlet/Rewritten</code>	URL originale
<code>http://server/servlet/Rewritten/123</code>	information <i>extra-path</i>
<code>http://server/servlet/Rewritten?sessionId=123</code>	ajout de paramètre
<code>http://server/servlet/Rewritten;\$sessionId\$123</code>	spécifique au serveur

- Chaque technique à ces avantages et ces inconvénients



Sessions: input HIDDEN

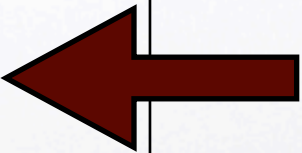
- L'identifiant de session est "caché" dans une entrée HIDDEN des formulaires retournés par le serveur:

- Exemple:

```
<form method="post" action="/cgi-bin/select">
  <input type="checkbox" name="art123">
  Java™ Servlet Programming, O'Reilly
  ..
</form>
```

réponse de /cgi-bin/select ->

```
<form method="post" action="/cgi-bin/validate">
<INPUT TYPE="hidden" NAME="SessionID" VALUE="54109848932">
  Identifiant: <INPUT TYPE="text" NAME="login">
  Adresse: <INPUT TYPE="text" NAME="mail">
  N° Client: <INPUT TYPE="text" NAME="numclient">
  ..
  <INPUT TYPE="hidden" NAME="Language" VALUE="French">
</FORM>
```



- Inconvénients:
 - Pas de persistance de l'ID côté client
 - Ambiguïté en cas d'utilisation de "Back"



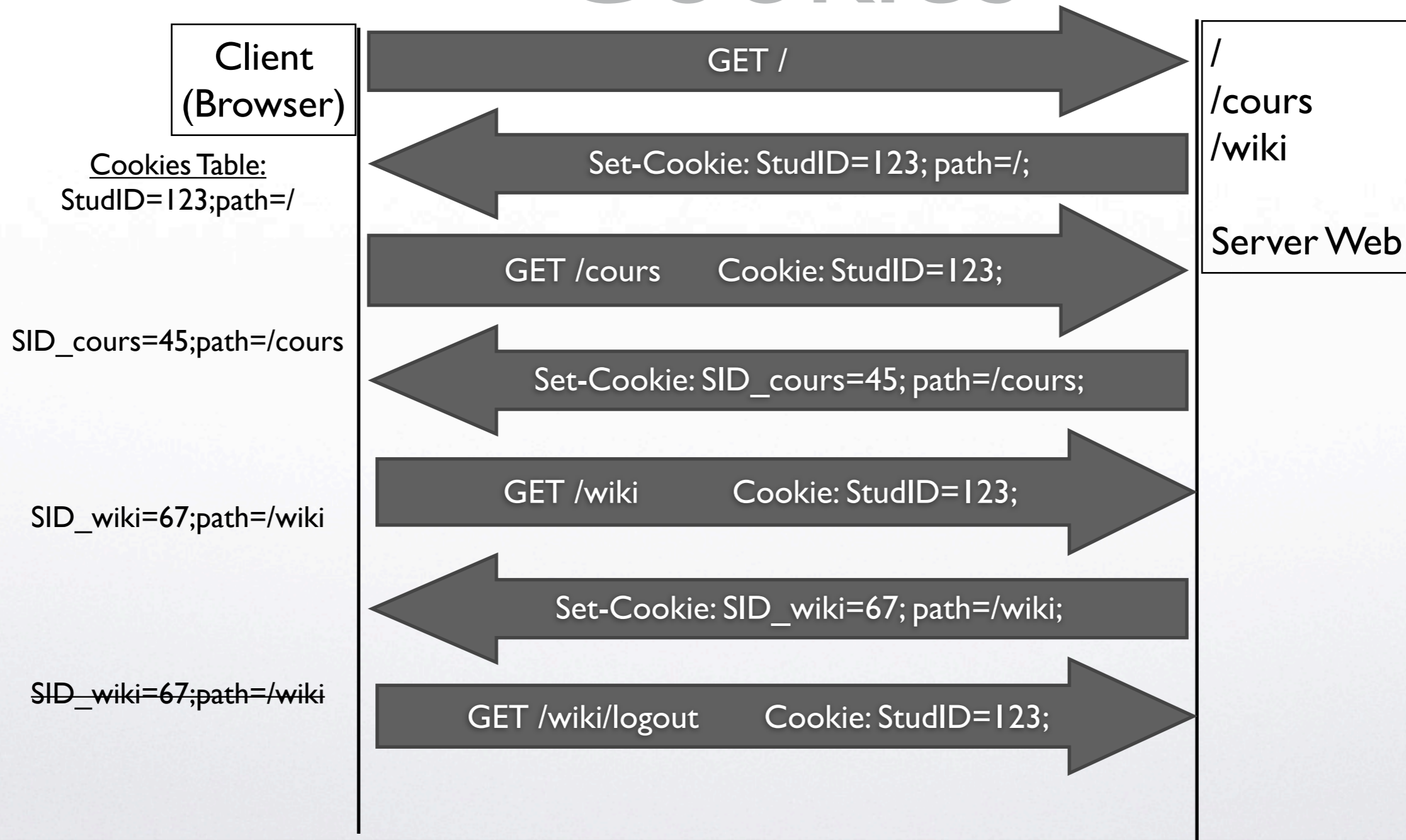
Sessions: cookies

- Ensemble d'informations stockées sur le client.
(Accessible par les scripts dans une page HTML)
- Envoyé dans les entêtes des requêtes (Cookie:) et des réponses (Set-Cookie:)
- Composé d'un ensemble de chaînes décrivant l'état d'une session:
 - NAME=<valeur>;
 - expires=<date>;
 - path=<chemin>; /, /foo, /foo/bar ou /foo/bar.html
 - domain=<hote>; books.amazon.fr
- Inconvénients
 - Limitations de taille (4ko par cookie, 20 cookies / domain et 300 / client)

```
Nom : JSESSIONID
Contenu : BADCC81AED2A837049E5E2A419F63E74
Hôte : www.arteradio.com
Chemin : /
Envoi pour : Tout type de connexion
Expire : à la fin de la session
```



Cookies





Bibliographie

- Java™ Servlet Programming:
 - <http://www.unix.org.ua/oreilly/java-ent/servlet/index.htm>
- Session Management:
 - <http://www.informit.com/articles/article.aspx?p=332851>
- Tutorials SUN:
 - <http://developers.sun.com/jscreator/fr/learning/tutorials/index.jsp>
- HTTP:
 - <http://en.wikipedia.org/wiki/HTTP>
 - <http://www.w3.org/Protocols/rfc1945/rfc1945>
 - <http://www.ietf.org/rfc/rfc2068.txt>