

Déroulé du module

- ▶ Rappels sur le protocole http, le requêtage dynamique et les formulaires
- ▶ PHP
- ▶ 8 séances de TD/TP et 8 séances de CM
- ▶ 1 contrôle écrit lors du dernier CM + 1 TP noté

Protocole HTTP et formulaires

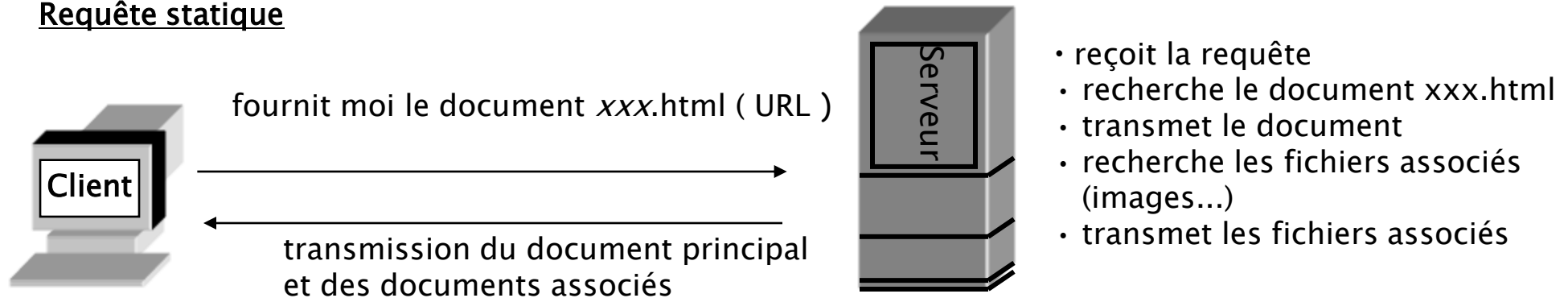
Sommaire

- ▶ **Architecture Web**
- ▶ Le protocole HTTP
- ▶ Formulaires HTML

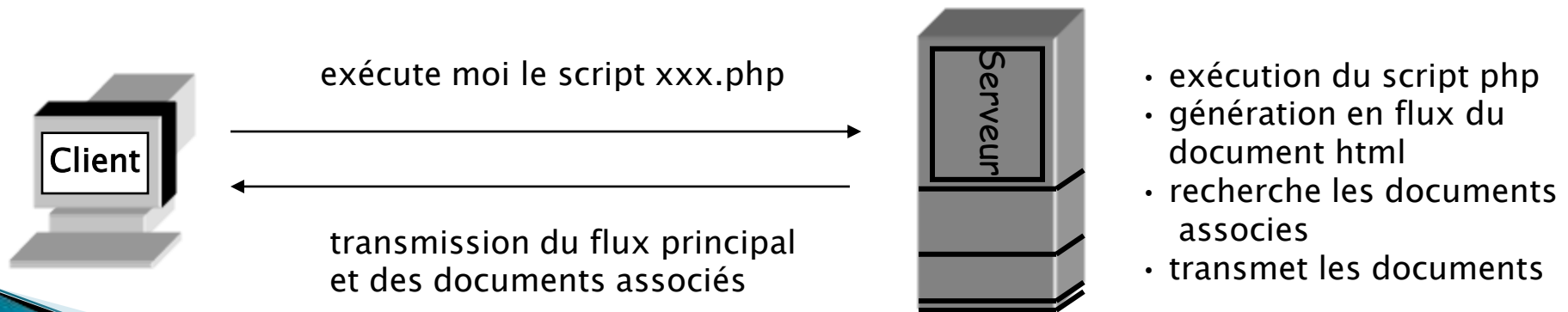
Architecture web : fonctionnement (principe général)

Reçoit des requêtes HTTP provenant d'un navigateur

Requête statique



Requête dynamique



Rôle du serveur HTTP

Le serveur HTTP a plusieurs "missions"

- ▶ Ecouter sur le réseau les requêtes émanant des différents clients
- ▶ Rechercher sur son disque les fichiers demandés et les renvoyer
- ▶ Lancer l'exécution d'un script (CGI,ASP,PHP...)
- ▶ Filtrer l'accès aux informations suivant une adresse IP, un mot de passe, ...
- ▶ Envoyer des messages d'erreurs en cas de requêtes défectueuses
- ▶ Gérer un nombre de connexions maximum
- ▶ Lancer différents processus pour gérer les réponses
- ▶ etc ...

Le rôle du navigateur

- ▶ Affiche les informations HTML fournies par le serveur :
 - Texte, images, formulaires de saisie, son, video, applets java
- ▶ Exécute les scripts "client" (Javascript)
- ▶ Emet des requêtes vers un serveur.
- ▶ Navigation et Affichage :

Sommaire

- ▶ Architecture Web
- ▶ **Le protocole HTTP (RFC 1945)**
- ▶ Formulaires HTML

Hyper Text Transfert Protocol

- ▶ HTTP est le protocole utilisé pour communiquer entre un client et un serveur Web.
- ▶ Les principales "commandes" échangées sont :
 - **GET** : Pour demander un fichier au serveur
 - C'est la requête la plus souvent utilisée par les navigateurs
 - Les paramètres éventuels sont envoyés dans l'URL
 - **HEAD** : Elle permet de ne récupérer que le header d'un document
 - Robots de recherches, serveurs proxy, etc ...
 - **POST** : Elle permet d'envoyer des données vers le serveur séparément de l'URL (plus de 255 caractères)

Syntaxe d'une requête

```
METHODE URL VERSION
EN-TETE : Valeur
. . .
EN-TETE : Valeur
Ligne vide
CORPS DE LA REQUETE
```

► Voici donc un exemple de requête HTTP:

```
GET http://www.univ-mlv.fr HTTP/1.0
Accept : text/html
If-Modified-Since : Saturday, 15-January-2000 14:37:11 GMT
User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)
```

Source : CCC

Syntaxe d'une réponse HTTP

Réponse HTTP = ensemble de lignes envoyées au navigateur par le serveur.

Elle comprend:

- ▶ **Une ligne de statut** : ligne précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif.

Elle comprend trois éléments séparés par un espace:

- La version du protocole utilisé
- Le code de statut
- La signification du code

- ▶ **Les champs d'en-tête de la réponse**: ensemble de lignes facultatives → donner des informations supplémentaires sur la réponse et/ou le serveur.
- ▶ **Le corps de la réponse**: il contient le document demandé

Source : CCC

HTTP : Requête client

- 1 - Le client contacte le serveur sur un port données (80 par défaut), puis lui envoie une requête HTTP :

GET /index.html HTTP/1.1

- 2 - Le client envoie un header optionnel pour informer le serveur de sa configuration et des documents qu'il accepte

User-Agent: Mozilla/3.01Gold (WinNT; I)

Accept: image/gif, image/jpeg, */*

- 3 - Le client envoie alors éventuellement des données, essentiellement dans le cas d'une requête utilisant la méthode POST

HTTP : Réponse Serveur

- 1 - Le serveur envoie une "status line" contenant trois champs: la version de HTTP employée, un code de réponse et une information "lisible" du status

HTTP/1.1 200 OK

- 2 - Le serveur envoie des informations à propos de lui-même et du document (headers) :

Date: Fri, 20 Sep 1998 08:17:58 GMT

Server: NCSA/1.5.2

Last-modified: Tue, 17 Sep 1998 08:34:58 GMT

Content-type: text/html

Content-length: 2482

- 3 - Ensuite, les données du fichier (et des fichiers attachés) sont envoyées

Sommaire

- ▶ Architecture Web
- ▶ Le protocole HTTP
- ▶ **Formulaires HTML**

Utilisation de formulaires

- ▶ Les formulaires permettent d'insérer dans une page HTML, une interface de saisie de données.
- ▶ La création de formulaires comporte deux stades:
 - création en HTML de l'interface "statique" proprement dite,
 - écriture de la procédure qui sera appelée à la validation.
- ▶ On utilise à l'intérieur d'un formulaire 3 types de composants:
 - des INPUT: bouton, champ texte, etc.
 - des SELECT: liste à sélection
 - des TEXTAREA: zone de texte multilignes

FORM

<FORM

ACTION = url

L'url désigne un script qui se trouve sur le serveur HTTP

ACTION="myScript.php"

L'url désigne une URL de type mailto

ACTION="mailto:eppstein@univ-mlv.fr"

METHOD = **GET** | **POST**

Désigne la méthode utilisée pour envoyer les informations au serveur

GET : Les informations seront placées dans l'URL désignant le cgi à activer (QueryString)

POST : Les informations sont envoyées séparément (dans une deuxième instruction HTTP).

ENCTYPE = application/x-www-form-urlencoded,
text/plain

Détermine le mécanisme utilisé pour encoder les données >

</FORM>

<INPUT

<INPUT type = TEXT	Un champ de saisie de texte.
PASSWORD	Idem mais caractères sous forme de *.
CHECKBOX	Case à cocher
RADIO	Pour avoir une option à choix multiple.
FILE	Utilisé pour permettre de lister des fichiers
RESET	Bouton utilisé pour remettre le formulaire dans l'état initial
BUTTON	Un bouton
SUBMIT	Un bouton pour soumettre le contenu du formulaire (ACTION) au serveur.
IMAGE	Idem submit mais avec une image
HIDDEN	Input "caché"

Les autres attributs de INPUT

NAME="*inputName*"

- Utiliser pour identifier le nom du composant de la propriété soumise au serveur

VALUE="Valider"

- Utiliser pour initialiser le champ, ou donner un nom aux boutons (reset et submit)

SRC="/gig/icon.gif"

- Une URL pour désigner l'image du bouton SUBMIT graphique.

SIZE=10 MAXLENGTH=7

Exemples :

```
<input type="text" name="champ1">
```

```
<input type="button" name="bouton1" value="Cliquez ici">
```

La sélection dans une liste : **SELECT**

- Pour proposer à l'utilisateur un choix unique ou multiple dans une liste d'item

<SELECT

NAME=*name* Utiliser pour identifier le composant côté serveur

SIZE=100 Nombre d'item visibles

MULTIPLE > Pour une sélection multiple

<OPTION

VALUE=*value* Valeur envoyée au serveur

SELECTED> Rend l'item sélectionné

</SELECT>

La zone de texte : TEXTAREA

- Pour afficher une zone de texte multiligne

<TEXTAREA

NAME=*name* Utiliser pour identifier le
composant côté serveur

ROWS=4 Nombre de lignes

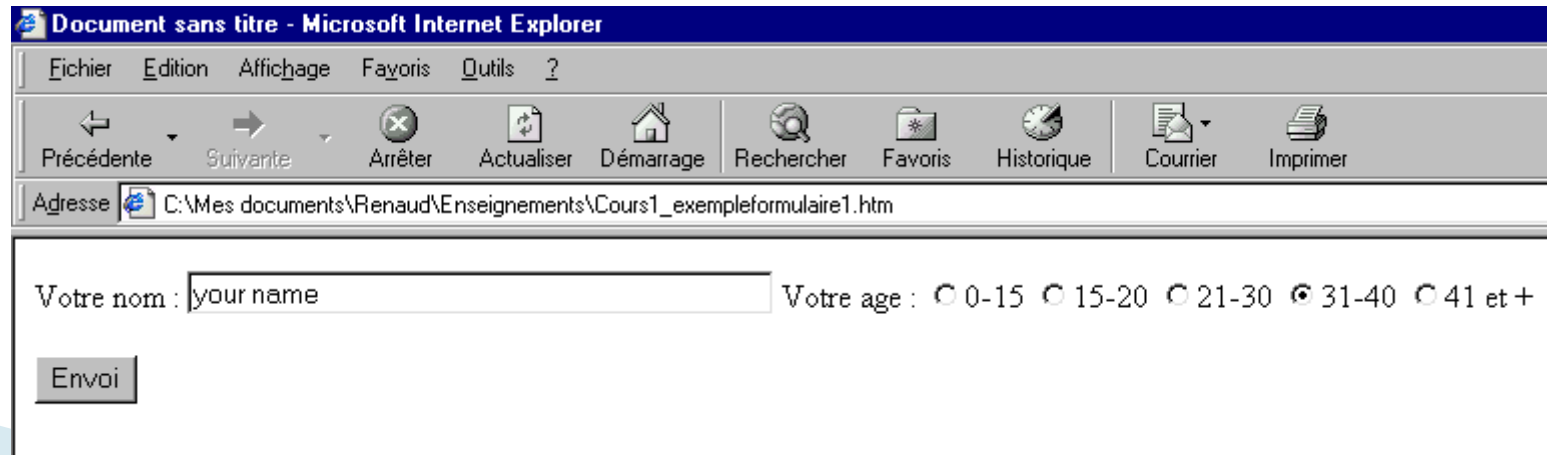
COLS=10 > Nombre de colonnes

... Le texte affiché dans la zone de texte ...

</TEXTAREA>

Exemple

```
<FORM ACTION="http://www.formulaire.com/toto.php" METHOD=GET >
  Votre nom : <input type=text size=40 name=user value="your name">
  Votre age : <input type=radio name=age value="0">0-15
               <input type=radio name=age value="1">15-20
               <input type=radio name=age value="2">21-30
               <input type=radio name=age value="3" checked>31-40
               <input type=radio name=age value="4">41 et +<BR>
  <input type=hidden name=customerid value="c2415-345-8563"><BR>
  <input type=submit value="Envoi">
</FORM>
```



Récupération des paramètres d'un formulaire

Fichier test.php

Entrez votre nom :

```
<FORM action="valide.php">  
<INPUT type="text" name="champ1">  
<INPUT type="submit" value="Cliquez ici">  
</FORM>
```

Fichier valide.php

