

M1105 - Cours n°5

Formulaires



1

I - Client / Serveur (rappel)

Principe de base : Client/Serveur

Machine Client

(ex : votre PC)

+

un logiciel « client web »

(ex : Firefox, Chrome, Safari, IE,...)

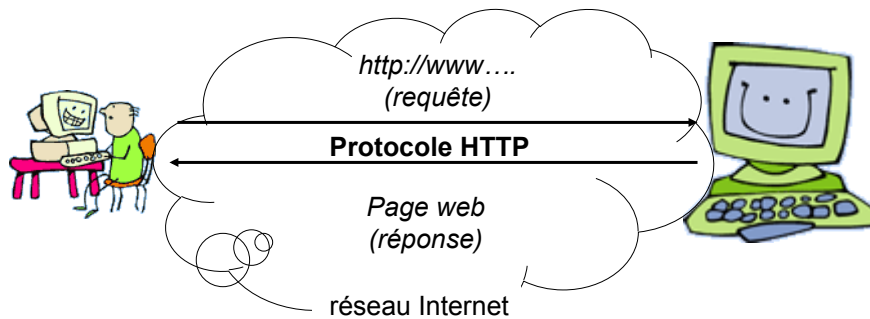
Machine Serveur

(ex : www.iut2.upmf-grenoble.fr)

+

un logiciel « serveur web »

(ex : Apache, IIS)



2

Pages HTML statiques vs dynamiques

- Pages statiques:
 - contenu fixé à l'avance => manipulation de nombreuses pages pour la maintenance
 - aucun moyen d'action pour l'utilisateur pour sélectionner et/ou transmettre des informations au serveur => pas de « personnalisation » des pages selon l'utilisateur

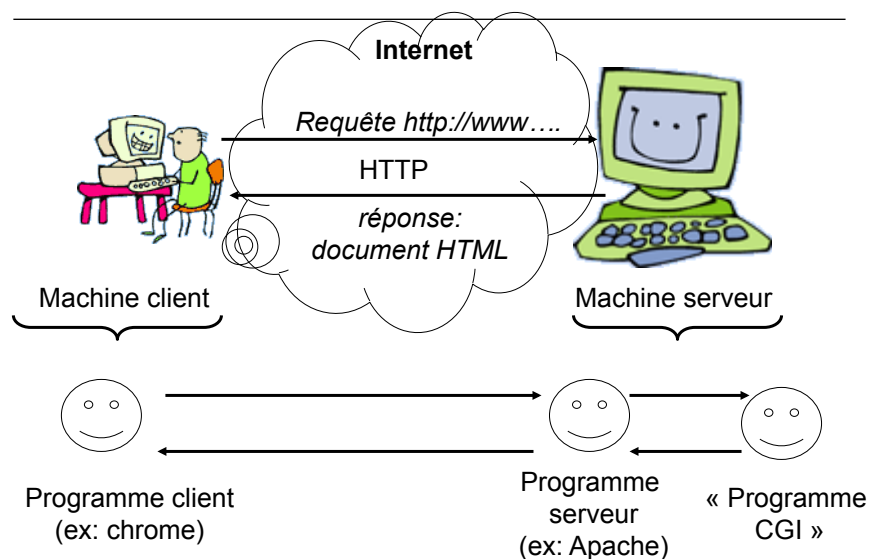
=> technique de base pour pallier ces inconvénients:

Interface **CGI** (*Common Gateway Interface*)

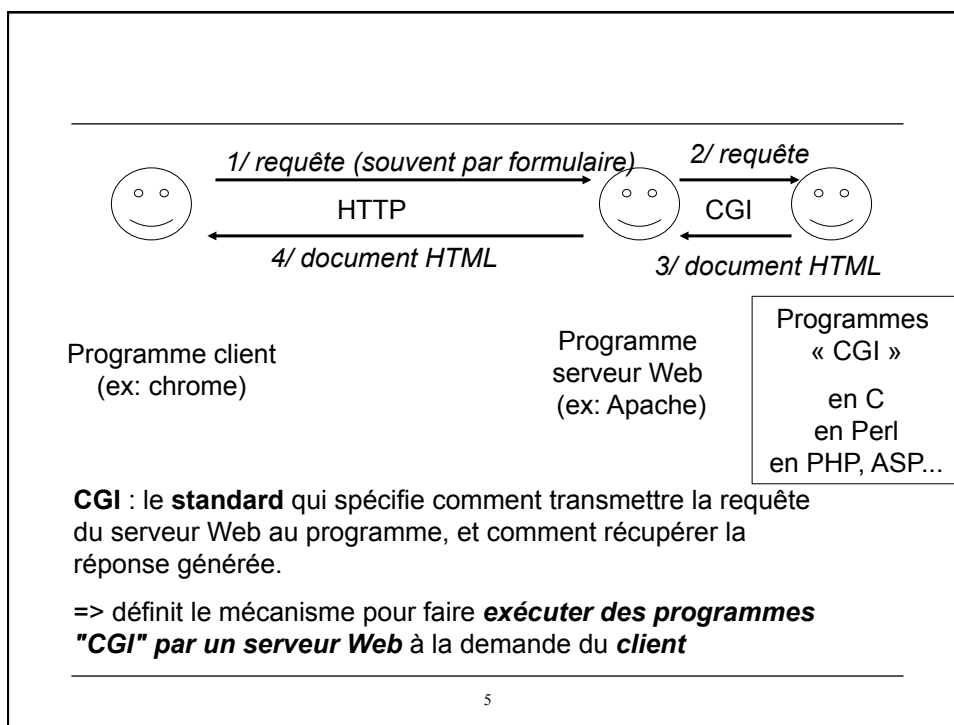
=> Pour des sites web dynamiques : créer des pages dynamiquement en fonction du contenu d'une base de données

3

Principe CGI



4



5

HTTP et CGI

HTTP : le client « réclame » simplement le fichier qu'il cherche:

HTTP + CGI : relation client/serveur plus élaborée :

- le fichier spécifié ne doit pas être retourné tel quel
- mais c'est un programme qui doit être exécuté
- c'est le résultat de l' exécution du programme qui est retourné au client
- des paramètres peuvent être passés au programme.

6

Langages de scripts

Langages de scripts "côté serveur" :

- C, C++, ADA,...
- Perl, JSP, ASP, **PHP**...

pour :

- exécuter une application
- collecter des données de formulaire
- consulter, mettre à jour,... des données d'une BD

PHP (*Hypertext Preprocessor*) :

- langage de scripts généraliste
 - Open Source
 - spécialement conçu pour le développement d'applications web
 - peut être intégré facilement au HTML
-

7

Limitations

- La technologie CGI requiert le lancement d'un nouveau processus à chaque requête
⇒ charge de traitement
 - Une solution est l'installation (installation de PHP par exemple) directement dans le serveur HTTP sous forme de module
 - C'est le cas avec le serveur web Apache: PHP installé en tant que "module Apache"
-

8

II- Formulaires

Les formulaires : le moyen simple de passer des paramètres/valeurs à un programme qui les traitera

- Un formulaire est une zone de saisie
- Le formulaire est intégré à une page HTML,
- Il permet à l'utilisateur de saisir des valeurs;
- Ces valeurs sont traitées par un programme externe (un programme PHP par exemple)

Le langage HTML propose un ensemble de balises pour définir des champs de saisie

9

Comment créer un formulaire?

- Dans une page HTML, on établit une zone de saisie (le **formulaire** ou **form** en anglais) en utilisant l'élément **form**
- On définit la méthode à employer pour transmettre au serveur l'information recueillie dans les champs du formulaire : la méthode **post** ou la méthode **get** (la différence est au niveau du passage des paramètres)
- On écrit un programme externe (par exemple un programme PHP) qui devra traiter les informations recueillies.

```
<form method="post" action="traite.php"> ... </form>
```

```
ou <form method="get" action="traite.php">... </form>
```

10

Les possibilités « de base »

1. Zones de saisie de texte:

- 1 seule ligne (ex: Google)
- plusieurs lignes

2. Les choix:

- Boutons radio (1 seul choix possible)
- Liste déroulante (1 seul choix possible)
- Cases à cocher (plusieurs choix possibles)

11

Zone de saisie texte: 1 seule ligne

`<input type="text" ...`

```
<form method="get" action="traite.php">
  <fieldset>
    <legend>Vos coordonnées:</legend>

    <label for="nom">Votre nom: </label>
    <input type="text" name="nm" id="nom" />
    <br />

    <label for="prenom">Votre prénom: </label>
    <input type="text" name="pn" id="prenom" />
    <br />
  </fieldset>

  <p>
    <input type="submit" value="Envoyez" />
    <input type="reset" value="Raz" />
  </p>
</form>
```

Vos coordonnées:

Votre nom: Votre prénom:

Explications

- 2 zones de saisie texte (`<input type="text" ...>`)
 - pour chaque zone, un label (`<label>`)
 - Les 2 zones de saisies sont regroupées en un ensemble (`<fieldset>`) auquel on donne un titre (`<legend>`)
- un bouton "reset" (`<input type="reset" ...>`)
- un bouton "submit" (`<input type="submit" ...>`)
- Le programme qui traite les valeurs saisies par l'utilisateur est un **programme php** => **traite.php**.
- Il y a donc **2 paramètres** passés au programme (**nm, pn**) qui contiendront les valeurs saisies (attribut **name** de la balise `<input>`)

13

`<input type="text" ...` Exemple d'attribut supplémentaire

- Auto-complétion ou d'auto-suggestion : **datalist**

```
...
<label for="browsers">Votre navigateur préféré: </label>
<input type="text" name="br" id="browsers" list="listBr" >
<datalist id="listBr">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

...

L'informatique et vous:

Votre navigateur préféré:

Envoyez

- Internet Explorer
- Firefox
- Chrome
- Opera
- Safari

<input type="text" ... variantes et attributs supplémentaires

Variantes:

```
<input type= "password" ...
<input type= "email" ...
<input type= "url" ...
<input type= "tel" ...
```

Attributs supplémentaires:

```
<input type="text" ... size="xx" autofocus required
placeholder="xxxx" ...
```

✓ voir le TP

15

Plus généralement

```
<form method= "..." action="...">
  <fieldset>
    <legend>...</legend>
    .....
  </fieldset>

  <fieldset>
    <legend>...</legend>
    .....
  </fieldset>

  ...
  <p>
    <input type="submit" value="..." />
  </p>
</form>
```

16

1 seul choix: bouton radio

<input type="radio" ...

```

...
<fieldset>
  <legend>Votre fruit préféré:</legend>
  <input type="radio" name="choix" value="pomme" id="pom" />
  <label for="pom">Les pommes ? </label>
  <br />

  <input type="radio" name="choix" value="poire" id="poi" />
  <label for="poi">Les poires ? </label>
  <br />

  <input type="radio" name="choix" value="peche" id="pec" />
  <label for="pec">Les pêches ? </label>
  <br />
</fieldset>

```

Votre fruit préféré: _____

- ☐ Les pommes ?
☐ Les poires ?
☐ Les pêches ?

17

1 seul choix: Liste déroulante

<select ...

```

...
<fieldset>
  <legend>Que préférez-vous ?</legend>
  <select name="animal">
    <optgroup label="Chien">
      <option value="labrador">Labrador</option>
      <option value="boxer">Boxer</option>
      <option value="berger">Berger allemand</option>
    </optgroup>
    <optgroup label="Chat">
      <option value="persan">Persan</option>
      <option value="siamois">Siamois</option>
    </optgroup>
  </select>
</fieldset>

```

- Chien
- Labrador
 - ✓ Boxer
 - Berger allemand
- Chat
- Persan
 - Siamois

18

Les choix multiples: cases à cocher type= "checkbox"

```
...
<fieldset>
<legend>Vos villes préférées: (plusieurs choix possibles) :</legend>
  <input type="checkbox" name="gre" id="grenoble" />
  <label for="grenoble">Grenoble</label>
  <br />

  <input type="checkbox" name="par" id="paris" />
  <label for="paris">Paris</label>
  <br />

  <input type="checkbox" name="bre" id="brest" />
  <label for="brest">Brest</label>
</fieldset>
...
```

Vos villes préférées: (plusieurs choix possibles) :

☐ Grenoble

☐ Paris

☐ Brest

19

HTML5: de nombreux types pour <input>

<input type="..." />

button, checkbox, color, date, datetime, datetime-local, email, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week

✓ voir le TP

20

HTML5: de nombreux attributs

Nouveaux attributs pour <form> et <input>:

- <form ... >:
 - autocomplete, novalidate
- <input ... >:
 - autofocus, autocomplete, checked, list, min, max, multiple, placeholder, required, size, step,...

👉 voir le TP

21

III - Passage des valeurs saisies et traitement

<form method= ".." **action="traite.php"**>

- Les valeurs saisies sont transmises au serveur Web (Apache par exemple)
 - Puis, elles sont transmises au programme PHP pour traitement
 - Elles sont transmises sous forme d' une seule chaîne de caractères
 - ⇒ une suite de couples *nomvariable=valeur* séparés par &
- var1=val1&var2=val2&var3=val3...**

Exemples de chaîne: (exemple du 1^{er} formulaire)

nm=lagaffe&pn=gaston

🔍 .../traite.php?nm=lagaffe&pn=gaston

22

<form **method= "??"** action="traite.php">

2 méthodes pour transférer les paramètres entre le client et le programme de traitement:

- **Méthode GET:** les paramètres sont encodés par le client dans l'URL à la suite de l'adresse au programme (ici, traite.php)

http://www-etu-info/~pestys/traite.php ?nm=lagaffe&pn=gaston

https://www.google.fr/# q=gateau+au+chocolat

- **Méthode POST:** l'URL ne contient plus la chaîne,
http://www-etu-info/~pestys/traite.php
les paramètres sont ensuite transmis par le serveur au script sur son entrée standard
-