Séance 4

Caroline Desprat et Catherine Comparot

25/05/2018

Manipulation base de données et formulaires

Interactions avec l'utilisateur

Fondée (entre autres) sur la validation de formulaire HTML

- provoque l'envoi de données vers le serveur
- permet au serveur de générer une page à partir de nouvelles informations en provenance de celui-ci

1. Syntaxe HTML
2. Interactions avec le serveur

Les formulaires

Syntaxe HTML

Dans le document HTML (ex: `form.html`)

```
<html>
<head> … </head>
<body>
  <h1>Veuillez saisir les informations suivantes</h1>
  <form action="" method=""> <!--form-->
      <label for="nom">Nom :</label> <!--label-->
      <input type="text" size="20" id="nom"/><br /> <!--input text-->
      <label for="prenom">Prenom :</label>
      <input type="text" size="20" id="prenom" /><br />
      <label for="mail">Adresse email :</label>
      <input type="email" size="22" id="mail"/><br />
      <input type="submit" value="Envoyer" /> <!--input submit-->
  </form>
</body>
</html>
```

Syntaxe HTML - visuel

Veuillez saisir les informations suivantes

Nom :

Prenom :

Adresse email :

Envoyer

La balise `form`

```
<form action="traitement" method="choix_methode">
      <!--mettre ici les élements-->
</form>
```

- `traitement` correspond à l'adresse du script qui effectuera le traitement des données saisies (ex: `traitement.php`)
- `choix_methode` correspond au type de la requête HTTP utilisée pour effectuer le transfert des données pour leur traitement. Deux possibilités : `GET` ou `POST`.

Les balises du formulaire

La balise de saisie `input`

Créer un contrôle interactif qui permet à l'utilisateur de saisir des données.

```
<input type="text" size="10" id="prenom" value="caroline"/>
<input type="date" size="20" id="nom" placeholder="votre date de naissance"/>
```

- `type` correspond au type de l'`input`. La valeur `text` permet de saisir un champ texte simple. Autres types de valeurs : `number`, `color`, `checkbox`, `radio`, `date`, `url`, `file`, `month`, `password`, `submit`, `range` ou `time`.
- `size` : longueur visible du champ (pas de la valeur).
- `value` : valeur par défaut du champ
- `placeholder` : indication, destinée à l'utilisateur, pour ce qui peut être saisi

Voir Input (MDN , compatibilité)
Voir aussi la balise Textarea (MDN)

Les balises du formulaire

La balise de saisie `input` - visuel

Input classiques

type “text”

caroline

type “date”

jj/mm/aaaa

type “email”

un email

Textarea

Vous pouvez écrire quelque chose ici.

Envoyer

Les boutons du formulaire

Bouton d'envoi :

1. **Verifier le format des données saisies au clic.**
2. Transmettre les données suivant l'attribut `action` spécifié dans la balise `form`.

```
<input type="submit" value="Envoyer" />
```

Bouton de réinitialisation

Reinitialise toutes les valeurs du formulaire aux valeurs par défaut.

```
<input type="reset" value="Annuler" />
```

Bouton "libre"

```
<input type="button" value="Afficher une fleur" />
```

Les balises du formulaire

La balise de sélection `select`

```
<form>
  <label for="enseignement">
  enseignement: </label>
  <select name="nom" id="enseignement">
      <option value="informatique" >Informatique</option>
      <option value="maths" selected="selected">Mathématique</option>
      <option value="lv">Langue vivante</option>
  </select> <br />
  <input type="submit" value="Envoyer" />
</form>
```

Organisation du formulaire

Objectif: faciliter les regroupements graphiques et la lisibilité

```
<form action="">
  <fieldset><legend>Mentions obligatoires</legend>
     Nom : <input type="text" size="20" /><br />
     Prénom : <input type="text" size="17" /><br />
     Adresse email : <input type="text" size="22" /><br />
  </fieldset>

  <fieldset> <legend>Mentions facultatives</legend>
      Sexe :
      <input type="radio" name="sexe" />Masculin
      <input type="radio" name="sexe" />Feminin<br />
      Vos suggestions :<br />
      <textarea cols="25" rows="4"></textarea>
  </fieldset>
  <p> Bouton d'envoi : <input type="button" value="Envoyer" /> </p>
</form>
```

Organisation du formulaire - Visuel

Mentions obligatoires

Nom :

Prénom :

Adresse email :

Mentions facultatives

Sexe : Masculin Feminin

Vos suggestions :

Bouton d'envoi : Envoyer

Comment faire pour que le serveur puisse traiter les données saisies dans le formulaire?

Interactions avec le serveur

Côté client : le formulaire

Le document `connexion.html`, dans la balise `body` :

```
<form action="validation_formulaire.php" method="post">
  <p>Nom: <input type="text" size="30" name="nom" /></p>
  <p>Prenom: <input type="text" size="20" name="prenom" /></p>
  <p><input type="submit" name="btAction" value="OK" />
    <input type="reset" name="btAnnuler" value="Annuler" /></p>
</form>
```

Rappel

- `validation_formulaire.php` correspond au fichier PHP qui effectuera le traitement des données saisies (attribut `action`).
- `POST` correspond au type de la requête HTTP utilisée pour effectuer le transfert des données (attribut `method`).

Donner un **nom à chaque élément du formulaire** : ce nom permet ensuite de retrouver les valeurs saisies par l'utilisateur

Visuel

`connexion.html`

Nom:

Prenom:

OK Annuler

Côté serveur : traitement du formulaire

Le formulaire précédent est traité par `validation_formulaire.php` :

```
...
<body>
        <?php
            if (empty($_POST["nom"]) or empty($_POST["prenom"])) {
                    echo "Vous devez saisir un nom et un prénom !";
            }
            else {
                $nom = $_POST["nom"];
                $prenom = $_POST["prenom"];
                echo "Bonjour $prenom $nom";
            }
        ?>
</body>
```

Côté serveur : traitement du formulaire - visuel

Rappel du formulaire

Nom: Dupont

Prenom: Marie

OK Annuler

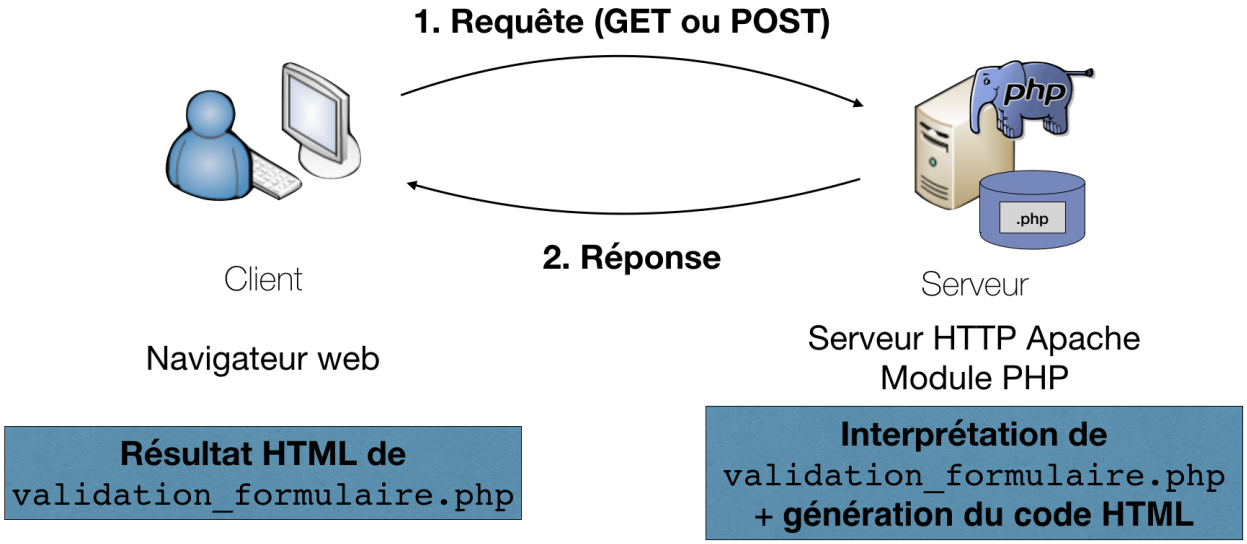
Si le formulaire est rempli le serveur retourne

Bonjour Dupont Marie

Sinon:

Vous devez saisir un nom et un prénom !

Cycle (simplifié)



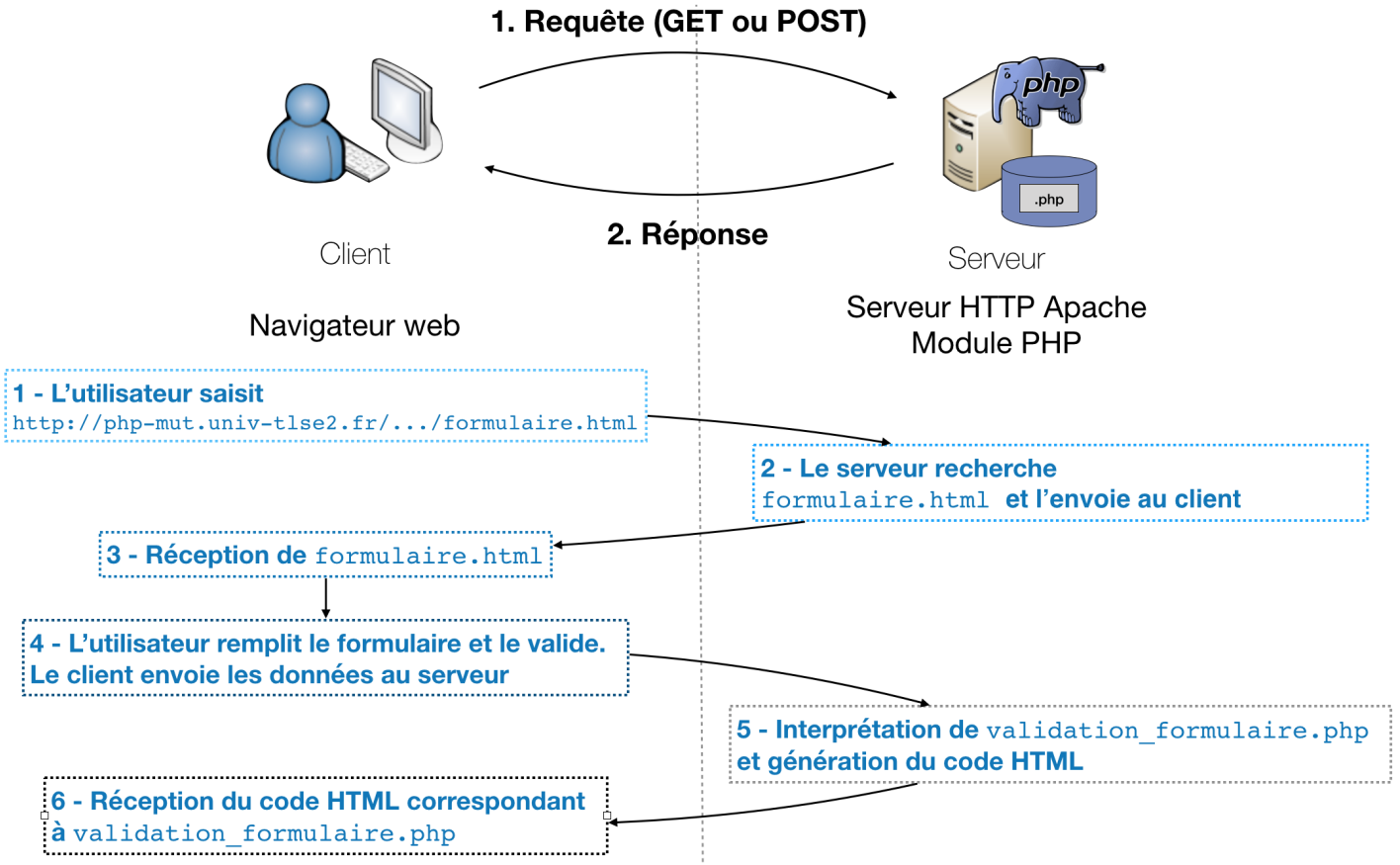
1. Le navigateur rassemble les informations saisies dans le formulaire (nom et prénom) et les envoie par une méthode `POST` (ou `GET`) au serveur

2. Le serveur interprète `validation_formulaire.php` avec les données reçues et génère un flux HTML résultat renvoyé au navigateur

Cycle (complet)

1. L'utilisateur se rend à l'adresse http://mi-phpmut/…/formulaire.html
2. Le serveur recherche `formulaire.html`
3. Le navigateur client récupère `formulaire.html`
4. L'utilisateur remplit le formulaire et le valide. Le client envoie les données au serveur
5. Le serveur interprète `validation_formulaire.php` et génère le code HTML résultat renvoyé au navigateur
6. Le navigateur reçoit et affiche le code HTML correspondant à `validation_formulaire.php`

Cycle complet - visuel



PHP : Variables globales permettant l'échange client / serveur

`$_POST[…]` ou `$_GET[…]`

- Variables globales permettant de récupérer les valeurs transmises au programme PHP par la requête client.
- Hachages dont les clés correspondent aux **noms** donnés dans le formulaire à chaque élément du formulaire, et les valeurs à la valeur saisie par l'utilisateur pour chacun de ces éléments.

Mise en pratique : TP

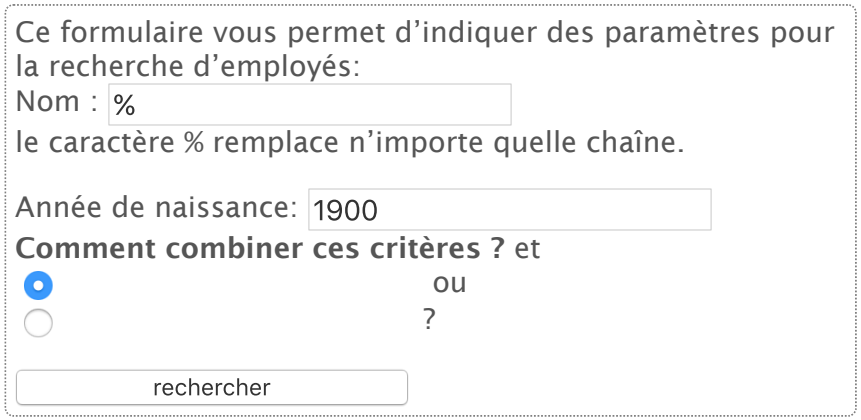
Faire l'exercice 1.

Formulaire d’interrogation

Exemple - Formulaire d'interrogation

`formulaire2.html`

```
...
<body>
  <form action="exemple2search.php" method="get">
    Ce formulaire vous permet d'indiquer des paramètres pour
    la recherche d'employés:
    <p>
    Nom : <input type="text" size="20" name="fnom" value="%"><br />
    le caractère % remplace n'importe quelle chaîne.</p>
    <p> Année de naissance:
        <input type="text" name="fnaissance" value="1900"><br />
    <b> Comment combiner ces critères ? </b>
    et <input type="radio" name="fcomb" value="et" checked="checked">
    ou <input type="radio" name="fcomb" value="ou">?
    </p>
    <input type=submit value='rechercher'>
  </form>
</body>
```



Exemple - Script PHP associé au formulaire

`exemple2search.php` avec une table `Employes [Id,Nom,Prenom,Naissance]`

```
<html>
<head>...</head>
<body>
<?php
  require("connect.php");

  // on récupère les valeurs du formulaire
  $vnom = $_GET['fnom'];
  $vnaissance = $_GET['fnaissance'];
  $vcomb = $_GET['fcomb'];

  echo "Requête : <b>Nom=$vnom et naissance=$vnaissance";
  echo " et combinaison=$vcomb</b><br/>";

  // création de la requête
  if ($vcomb == "et")
      $requete = "select * from Employes where Nom like '$vnom'
                and Naissance=$vnaissance";
  else
      $requete="select * from Employes where Nom like '$vnom'
                or Naissance=$vnaissance";


  $connexion = mysqli_connect(SERVEUR,NOM,PASSE);
  mysqli_select_db($connexion,BASE);

  // exécution et affichage du résultat
  $resultat = mysqli_query($connexion,$requete);

  while ($emp = mysqli_fetch_object ($resultat)){
    echo "$emp->Id, de nom $emp->Nom $emp->Prenom, né le
          $emp->Naissance<BR>\n";
  }
 ?>
</body>
</html>
```

Les champs du formulaires sont dans le tableau **`$_GET`** (méthode GET spécifiée dans le formulaire HTML)

Formulaire de mise à jour

Exemple - Formulaire de mise à jour

insertion.html

```
<html>
<head>
  <title>formulaire de mise à jour</title>
</head>
<body>
  Insertion dans la table employes:  <br/>
  <form action="exemple3update.php" method="post">
    nom: <input type="text" size="20" name="fnom"><br>
    prénom: <input type="text" size="20" name="fprenom"><br>
    année naissance: <input type="text" size="4" name="fnaissance"><br>
    <input type="submit" value="insérer">
  </form>
</body>
</html>
```

Insertion dans la table employes:
nom:
prénom:
année naissance:
insérer

Exemple - Script PHP associé au formulaire de mise à jour

`exemple3update.php`

```
<html>
<body>
<?php
  require("connect.php");

  //récupération des variables
  $vnom = $_POST['fnom'];
  $vprenom = $_POST['fprenom'];
  $vnaissance = $_POST['fnaissance'];

  $connexion = mysqli_connect(SERVEUR,NOM,PASSE);
  mysqli_select_db($connexion,BASE);

  $requete="insert into employes(nom,prenom,naissance)
            values ('$vnom','$vprenom',$vnaissance)";

  $resultat = mysqli_query($connexion,$requete);
  if ($resultat){
    echo "<br>la requête a été effectuée.";}
  else {
    echo "la requête n'a pas pu être effectuée
          pour la raison suivante:".mysqli_error($connexion); }
?>
</body>
</html>
```

Lignes de sélection

```
<form>
  Enseignement:
  <select name="nom" >
      <option value="informatique" selected="selected">Informatique</option>
      <option value="maths">Mathématique</option>
      <option value="lv">Langue vivante</option>
  </select> <br />
  <input type="submit" value="Envoyer" />
</form>
```

Choix multiples

On veut récupérer plusieurs sélections faites par l'utilisateur côté client :

```
<form method="post" action="manger.php">
  <select name="dejeuner[]" multiple>
      <option value="jambon">Sandwich au jambon</option>
      <option value="poulet">Sandwich au poulet</option>
      <option value="thon">Sandwich au thon</option>
      <option value="gruyère">Sandwich au gruyère</option>
  </select>
<input type="submit" name="soumettre">
```

Côté serveur, en PHP, `$_POST['dejeuner']` est un tableau contenant les valeurs sélectionnées :

```
foreach ($_POST['dejeuner'] as $choix){
    echo "Vous avez pris un sandwich au $choix. <br />";
}
```

Formulaire validé par lui-même

Formulaire validé par lui-même

Dans le premier exemple, lorsque l'utilisateur s'est trompé ou s'il veut faire une nouvelle saisie, il doit revenir en arrière en utilisant la touche back du navigateur

Il est souvent plus judicieux d'afficher le formulaire et le résultat sur une même page.

Exemple **toutenun.php** 1/2

```
...
<body>
    <div>
        <?php
            if ($_SERVER["REQUEST_METHOD"] == "POST") {
                // Est appelé par la méthode POST, donc en validation du formulaire
                if (empty($_POST["nom"])or empty($_POST["prenom"])) {
                    echo "Vous devez saisir un nom et un prénom !";
                }
                else {
                    $nom = $_POST["nom"];
                    $prenom = $_POST["prenom"];
                    echo "Bonjour $prenom $nom";
                }
            }
        ?>
... (suite)
```

Exemple **toutenun.php** 2/2

```
        <form action="toutenun.php" method="post">
            <p><label for="nom">Nom :</label>
            <input type="text" size="30" name="nom" />
            </p>
            <p><label for="prenom">Prénom :</label>
            <input type="text" size="20" name="prenom" />
            </p>
            <p><input type="submit" name="btAction"
                value="OK" />
            <input type="reset" name="btAnnuler"
                value="Annuler" />
            </p>
        </form>
    </div>
</body>
...
```

Mise en pratique

TP

Faire les exercices 1, 2 et 3

Références

Manuel de référence

- http://www.php.net/manual/fr/

Pour avoir des exemples

- http://www.phpfrance.com/
- http://www.manuelphp.com