

TP-Un formulaire en php

On dépose le dossier (décompressé) dans le répertoire `www/myapp` de UwAmp

Ce dossier contient 4 fichiers

- Un fichier html (`index.html`)
- Un fichier php (`reponse.php`)
- Un fichier de style (`style.css`)
- Un fichier image (`ico.png`)

Ouvrir ce dossier (avec Brackets pour accéder au code) et depuis le serveur pour voir la page dans le navigateur

Objectif :

Dans ce TP on construit un formulaire dans une page web et on traite le formulaire avec le langage php.

Rappel :

- Le html permet de créer le formulaire (et est interprété par le navigateur)
- Le php permet de traiter les informations (et est interprété par le serveur)

Ce qui fait qu'il est indispensable de déposer ces fichiers sur un serveur...

Code de base d'un formulaire

Dans une page html classique, l'introduction de ce code permet la création d'un formulaire :

Le formulaire est introduit par la balise HTML `<form>`.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <link rel="icon" type="image/png" href="ico.png"/>
5      <meta charset="utf-8">
6      <title>IHM-Web </title>
7      <link rel="stylesheet" href="style.css">
8
9  </head>
10
11 <body>
12     <h1> TP formulaire avec du php</h1>
13
14     <form method="post" action="reponse.php">
15         <!-- On place ici les différents contrôles du formulaire -->
16     </form>
17
18
19
20
21
22 </body>
23 </html>
24
25
```

- **L'attribut "method"** : ici on utilise POST, on peut également utiliser GET.
 - **GET** est la méthode par défaut, il ne faut pas l'utiliser si le formulaire contient des informations sensibles car elles apparaissent en clair dans l'URL.
 - **POST** offre une meilleure sécurité car les informations n'apparaissent pas dans l'URL.
- **L'attribut "action"** : il contient le nom du script php qui sera appelé lorsque le formulaire sera validé par l'utilisateur.

Les différents contrôles

On construit notre formulaire avec des "contrôles", en voici une liste non exhaustive

Pour chaque contrôle **l'attribut "name"** définit le nom de la variable qui contiendra l'information saisie par l'utilisateur

Texte	<code><input type="text" /></code>
Bouton radio	<code><input type="radio" name="etat" value="on" checked/></code>
Case à cocher	<code><input type="checkbox" name="accept" id="accept" checked="checked" /></code>
Bouton	<code><input type="submit" value="Submit" /></code>
Liste déroulante	<code><select name="section">...</select></code>
Slider	<code><input name="slider" type="range" min="0" max="20" step="1" value="5" /></code>
Zone de texte	<code><textarea name="memo" rows="5" cols="30">Contenu dans la zone de texte</textarea></code>

Vous trouverez sur le web pléthore de sites sur la balise [<form>](#).

Exemple de formulaire

📄 Compléter la page index.html avec le code ci-dessous

```
13 <hr>
14 <form action="reponse.php" method="POST"><!-- Début du formulaire -->
15 <!-- Liste déroulante -->
16 <select name="titre">
17 <option>Mlle</option><option selected>Mme</option><option>M.</option>
18 </select>
19 <!-- Zones de texte -->
20 <label for="nom"> Nom : </label><input type="text" name="nom" id="nom" placeholder="Entrez votre nom" /><br/><br/>
21 <label for="prenom">Prénom : </label><input type="text" name="prenom" id="prenom" placeholder="Entrez votre prénom" /><br/><br/>
22 <label for="annee"> Votre année de naissance : </label><input type="text" name="annee" placeholder="2019" /> <br/><br/>
23 <label for="identifiant"> Donnez un identifiant : </label><input type="text" name="identifiant" placeholder="id" /> <br/><br/>
24 <label for="mdp"> Donnez un mot de passe : </label><input type="password" name="mdp" placeholder="mot de passe" /> <br/><br/>
25 <!-- Bouton radio -->
26 <label for="sexe">Sexe : </label>
27 <input type="radio" name="sexe" id="sexe" value="H" checked /> Masculin
28 <input type="radio" name="sexe" value="F" /> Féminin
29 <br/><br/>
30 <!-- Case à cocher -->
31 <input type="checkbox" name="debutant" checked="checked" /> Je débute en PHP<br/><br/>
32 <!-- Bouton -->
33 <input type="submit" name="btOK" value="Envoyer">
34 </form> <!-- Fin du formulaire -->
35 <hr>
```

📄 Enregistrer et actualiser la page dans le navigateur.

Vous devriez voir :

Le code précédent affiche le formulaire mais ne fait aucun traitement....

TP formulaire avec du php

Mme ▼ Nom :

Prénom :

Votre année de naissance :

Donnez un identifiant :

Donnez un mot de passe :

Sexe : ☒ Masculin ☐ Féminin

☒ Je débute en PHP

Traitement du formulaire

Dans un premier temps, nous allons faire afficher les informations reçues par le script php.

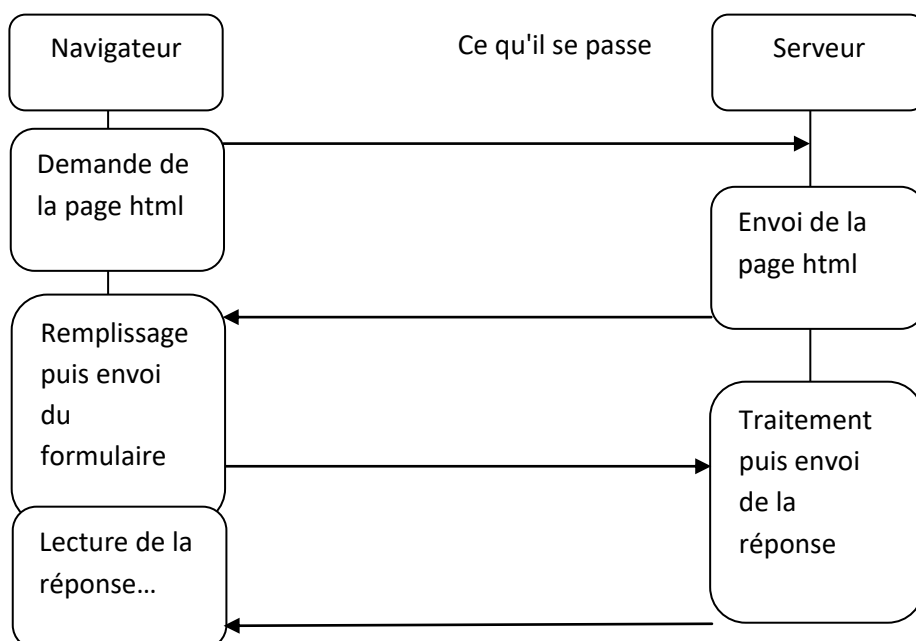
📁 Dans le fichier reponse.php, écrire le code suivant:

Le code php s'écrit entre les balises : `<?php.....?>`

```
1 <!doctype html>
2 <html>
3 <head>
4     <link rel="icon" type="image/png" href="ico.png"/>
5     <meta charset="utf-8" />
6 <title>
7     Test Formulaire PHP
8 </title>
9 </head>
10 <body>
11 <?php
12     if($_POST) {
13         echo 'Contenu de la variable $_POST : >';
14         print_r($_POST);
15     }
16 >?
17 </body>
18
19 </html>
20
```

📁 Enregistrer, et dans le navigateur renseigner le formulaire puis envoyer

Vous devriez voir un affichage sous forme d'une liste (array) où pour chaque variable (entre crochet) on associe la valeur



TP-Un formulaire en php

Nous allons maintenant créer une réponse du serveur plus élaborée...

Pour accéder à la valeur d'une variable en particulier, on utilise: `$_POST['nom de la variable']`

Pour faire afficher la valeur d'une variable, on utilise :

```
<?php echo $_POST['nom de la variable'] ?>
```

Il y a deux méthodes pour réaliser un affichage des données :

- On met du code html dans du code php :

```
<?php
    echo "<p>le code html entre guillemets " . $_POST['nom'] . " de nouveau du code html </p>";
    echo "<br/>encore du html " . $_POST['prenom'] . " Du code html <b>là aussi</b>";
?>
```

- ➔ Chaque instruction se termine par un point-virgule
- ➔ Le code html est encapsulé dans des guillemets
- ➔ echo pour faire afficher sur la page ce qui suit...
- ➔ Les variables php se délimitent par des points: `$_POST['nom']`.

- On met du code php dans du code html:

```
<h1>Bonjour, <?php echo $_POST['prenom']?> <?php echo $_POST['nom'] ?></h1>
```

```
<h2>Votre mot de passe est: <?php echo $_POST['mdp'] ?> </h2>
```

- ➔ On écrit du code html classique, et pour une variable php on l'encapsule dans les balises `<?php....?>`

Dans l'exemple de traitement ci-dessous, nous utiliserons la première méthode

📁 Dans la page `reponse.php` écrire le code suivant (le code est commenté)

```
<?php
// Si la variable nom n'est pas vide
if($_POST['nom']!=''){
    //On fait afficher les informations transmises au serveur
    echo "<br/><br/> <h1>Bonjour " . $_POST['titre'] . " " . $_POST['prenom'] . " " . $_POST['nom'] . " </h1><br/>";
    echo "<h2> Votre identifiant est :". $_POST['identifiant'] . " </h2>";
    echo "<h2> Votre mot de passe est :". $_POST['mdp'] . " </h2><br/>";
    // Si on a coché H ou F la variable $mot se différencie
    if($_POST['sexe']=='H') {
        $mot = "débutant";
    }
    else {
        $mot = "débutante";
    }
    //Si on a coché la case débutant, on affiche une phrase qui change selon la valeur de $mot
    if(isset($_POST['debutant'])){
        echo "<h2> Comme vous êtes " . $mot . " C'est une bonne idée de commencer à apprendre à programmer en PHP !</h2><br/><br/>";
    }
    // Si la variable annee n'est pas vide on affiche la page wikipédia de l'année
    if(isset($_POST['annee'])){
        echo " <h1>Voilà les faits importants de votre année de naissance :</h1> <br/><br/>";
        $url="https://fr.wikipedia.org/wiki/" . $_POST['annee']; // on stocke l'url de la page dans une variable
        echo "<iframe src='".$url.' ' width='1500px' height='800px' >"; // on affiche la page de wikipédia dans notre réponse
    }
}
?>
```

(isset (...)) permet de savoir si une variable n'est pas vide....)

(\$mot=... stocke une valeur dans la variable \$mot)

📁 Tester la bonne communication entre le serveur et le navigateur

📄 **À faire** : Réaliser une seconde page (reponse2.php) en utilisant la seconde méthode.

Et la sécurité...

Le code ci-dessus a le gros défaut de "faire confiance" à l'utilisateur en ne vérifiant pas les informations reçues. En effet, un utilisateur mal intentionné peut par exemple à la place de son nom mettre du code HTML ou JavaScript ce qui représente une faille de sécurité très importante.

Taper par exemple : `<script> alert('Site Hacké !!!'); </script>` à la place de votre nom et valider le formulaire.

Pour éviter ce désagrément, on peut faire en sorte que les balises html et php ne soient pas exécutées par le serveur avec la fonction : `strip_tags()`

Par exemple : Dans la page reponse.php modifiez chaque variable comme celle-là :

```
.strip_tags($_POST['prenom']).
```

📄 "Sécuriser" toutes les variables et tester

Ce n'est qu'une façon de faire (pas très jolie), il y a bien d'autres failles de sécurité. ..
Lorsqu'en terminale nous verrons comment lier un formulaire à une base de données, nous serons amenés à parfaire cette sécurité...

Travail à rendre :

Élaborer une page web(index.html) avec sa réponse (reponse.php) avec un formulaire que vous imaginerez

(en vous inspirant de ce TP et de ce que vous trouverez sur le web...(php, html))

Idée de projet : Création d'un 'mini' site web interactif