TP – Mon Premier Vrai Formulaire

Vous allez pouvoir créer des pages PHP afin de créer des formulaires et les traiter. Pour faire fonctionner les pages, vous devez déposer les fichiers sur votre serveur PHP : le sous répertoire www pour WAMP et httdocs pour XAMPP.

Pour ce TP, Utilisez le repository (à créer si besoin) /php/decouverte-php/S1/tp-form/ et publiez-le sur GITHUB en public.

Ce dossier contient 4 fichiers

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

Trouvez bien entendu un fichier ico pour rendre la page plus jolie.

# Créer le 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>.

* L'attribut "method" : ici on utilise POST, on peut également utiliser GET.
* L'attribut "action" : il contient le nom du script php qui sera appelé lorsque le formulaire sera validé par l'utilisateur.

**Quelle méthode ?**

**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.  
La sécurité reste toute relative car cela est envoyé « en clair ».

**Les différents contrôles**

On construit notre formulaire avec des « contrôles ». Pour chaque contrôle l'attribut "name" définit le nom de la variable qui contiendra l'information saisie par l'utilisateur

Vous trouverez sur le web pléthore de sites sur la balise <form>.

Créez le formulaire suivant.

Graphical user interface

Description automatically generated

# Traiter le 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 :

if ($\_POST)

{

**echo** "contenu de post";

**print\_r**($\_POST);

}

Pour tester : 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.

Voilà ce qui se passe

Diagram

Description automatically generated

# Améliorer l’affichage

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['nomDeLaVariable']; ?>

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>";

?>

Attention :

* 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 le code PHP dans le 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.  
On peut également taper

<?=$nomVariable;?>

**A faire :**

* Affichez les différentes informations saisies de manière élégante
* Gérez l’affichage en cas d’une saisie vide dans le cas du formulaire
* Utilisez la deuxième méthode, c’est la bonne pratique

# Un peu de 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['nom']);

A faire :

* Sécuriser l’ensemble du formulaire