e-catalyst

Au cœur de vos attentes. En réponse à vos besoins

Exercice

Formulaires

Consignes de l'exercice

- Dans cet exercice, un fichier html fourni contient un formulaire HTML utilisant la méthode get et envoyant la variable fname au script PHP index.php
- Vous allez compléter la première ligne de code pour afficher la variable fname envoyée au script à la suite de l'instruction echo
- Une fois que vous avez testé ces modifications, vous allez modifier le formulaire html pour que celui ci utilise la méthode post
- Vous allez modifier par la suite le fichier index.php pour afficher la variable fname envoyée via la méthode post

Théorie du cours

Une utilisation très courante de PHP est de récupérer les données envoyées par l'utilisateur afin d'effectuer diverses opérations utilisant ces données.

Pour récupérer les données, on utilisera les variables super globales $_GET et $_POST, si le formulaire utilise la méthode get pour envoyer les données, on utilisera $_GET en PHP pour les récupérer, si le formulaire utilise la méthode post, on utilisera $_POST pour récupérer ces données.

Une alternative est d'utiliser la super globale $_REQUEST qui va récupérer les données quelque soit la méthode utilisée par le formulaire.

- Il est très important dans le cas où php va traiter des donnée venant de l'extérieur de garder à l'esprit l'aspect sécurité : La règle générale est de considérer que toutes données envoyées par l'utilisateur est potentiellement malveillante.

Il existe de nombreuses façons en PHP de sécuriser les échanges client/serveur, nous aborderons dans de prochains exercices.

- $_GET,$_POST et $_REQUEST retournent un tableau , ces trois variables sont des super globales, ceci veut dire qu'elles sont accessibles à tout moment et n'importe où dans votre code.

- $_GET est une liste de variable passées à votre programme via la barre d'adresse du navigateur

- $_POST est une liste de variable passées à votre programme via la méthode POST du protocole HTTP

Quand faut il utiliser GET :

Les variables passées via la méthode GET sont visibles par tout le monde, il ne faut donc pas envoyer d'informations sensibles via cette méthode.

Il existe également une limitation dans la quantité de donnée pouvant être envoyée via get

L'avantage de GET est qu'il est possible pour l'utilisateur de mettre en marque page la page envoyée contenant les données envoyées dans l'url.

Quand utiliser POST :

Les données envoyées via POST sont invisibles pour les utilisateurs, il n'y a pas de limite au niveau de la quantité de données pouvant être envoyée.

C'est donc la méthode à utiliser pour envoyer des mots de passe, des fichiers etc...

Exemple utilisation $_GET en PHP :

Un formulaire envoie le champ suivant : nom ayant pour valeur "Marcel"

En PHP, pour récupérer la valeur, on procédera de la sorte :

$data = $_GET['nom'];

$data va contenir "Marcel"

Le fonctionnement est similaire lorsqu'on utilise $_POST ou $_REQUEST

Liens vers la documentation php.net :

- http://php.net/manual/fr/reserved.variables.request.php
- http://php.net/manual/fr/reserved.variables.get.php
- http://php.net/manual/fr/reserved.variables.post.php

Considérations sur la sécurité :

Une bonne façon de sécuriser les données reçues par l'utilisateur est d'utiliser la fonction filter_var()

Cette fonction accepte en parametre des filtres prédéfinis permettant de "nettoyer" ou valider les données reçues.

Exemple:

$maVar = "dutext<$@";

$maVarNettoyée = filter_var($maVar,FILTER_SANITIZE_STRING);

Testez ce code pour voir ce que $maVar affiche.

Liens vers la documentation sur php.net :

- http://php.net/manual/fr/function.filter-var.php
- http://php.net/manual/fr/filter.filters.sanitize.php
- http://php.net/manual/fr/filter.filters.validate.php