

Transmission des données d'une page à une autre

Introduction

- Les pages d'un site web sont liées entre elles, et il peut y avoir des informations qui doivent être partagées entre certaines pages web.
- Il existe trois manières de transmettre les données entre pages web :
 1. Transmission via un URL
 2. Transmission via un formulaire
 3. Transmission via des variables de session (A voir plus tard)

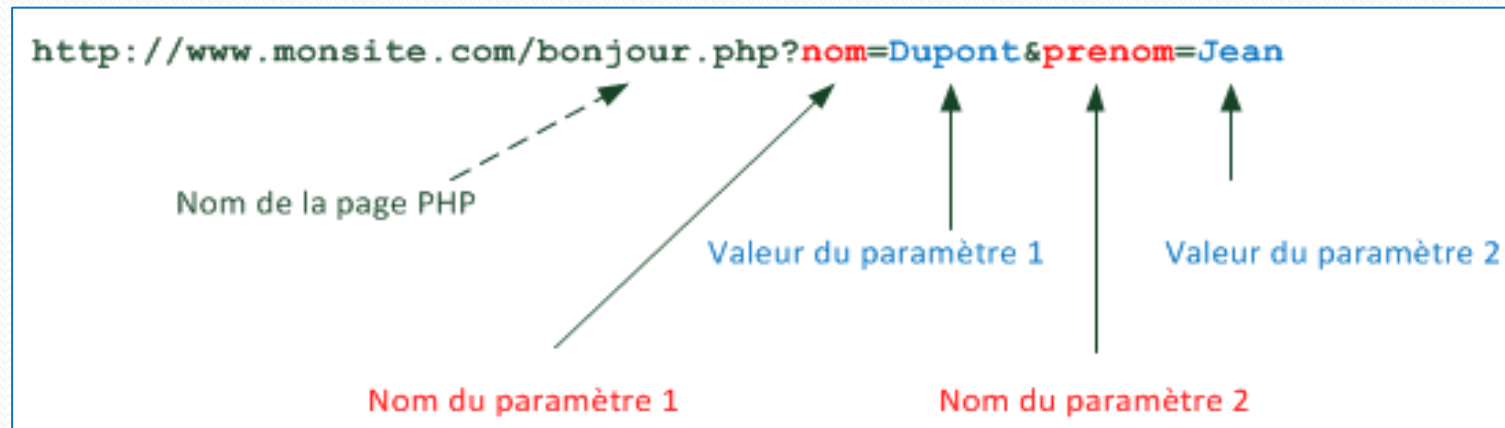


Transmettre des données via un URL

- L'URL signifie **Uniform Resource Locator**. Elle sert à représenter une adresse sur le web.
- Pour transmettre des données via l'URL, il existe plusieurs manières, une des manières c'est d'utiliser des **liens**.
- Donc il faut savoir:
 1. Comment passer des informations de la page 1 vers la page 2 via l'URL
 2. Comment récupérer ces informations dans la page 2

Transmettre les données via URL

- Les données apparaissent dans l'URL de cette façon :



Transmettre les données via URL

Syntaxe générale :

```
<?php  
echo "<a href=page2.php?parametre1=valeur&parametre2=valeur >  
Aller vers la page suivante </a>";  
?>
```

- Exemple dans **Page1.php** :

```
<?php  
echo "<a href=page2.php?p1='Mouhamed'&p2=16 >Aller vers la  
page suivante </a>";  
?>
```

Récupérer les données dans la page destination

- Pour récupérer les données transmises dans l'URL, on se sert d'un **tableau associatif** prédéfinis par PHP: **\$_GET**
- Syntaxe générale :

```
<?php  
echo $_GET['parametre1']." ". $_GET['parametre2'];  
?>
```

- Exemple **Page2.php** :

```
echo "Le nom",$_GET['p1']." agé de" .$_GET['p2']. " ans" ;
```

Transmettre les données via un formulaire

- Le formulaire est un moyen qu'on utilise pour donner à l'utilisateur la possibilité de saisir des informations pour les traiter par la suite.

Rappel sur les formulaire en HTML

- Pour définir un formulaire, on utilise la balise **form**:

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

- L'attribut **action**: définit la page de destination (celle qui va traiter les données du formulaire)
- L'attribut **method**: permet de définir la méthode de soumission des informations du formulaire, il y'a deux méthodes **get** et **post**.

La méthode de transfert des données dans un formulaire

- Si on utilise la méthode **get**, les informations vont être transmises via l'**URL**, les informations seront stockées dans un tableau associatifs **\$_GET**. Les informations seront visibles dans l'URL .
- Si on utilise la méthode **post**, les informations vont être stockées dans le tableau associatif **\$_POST**. Et les informations ne seront pas visibles par l'utilisateur.

La méthode GET

- Le principal inconvénient de la méthode GET est **l'absence de protection des données**. Les paramètres URL envoyés sont non seulement visibles par tous dans la barre d'adresse du navigateur, mais sont également accessibles par les utilisateurs d'où la possibilité de modifier des valeurs et provoquer des erreurs et des problèmes dans le site.

La méthode POST

- La méthode POST écrit les paramètres URL dans la **requête HTTP** pour le serveur. Les paramètres ne sont donc pas visibles pour les utilisateurs.
- Lorsqu'il s'agit de transmettre des données sensibles au serveur, par exemple un formulaire d'inscription avec nom d'utilisateur et mot de passe, la méthode POST permet de garder la **confidentialité** nécessaire. Les données ne sont pas mises en cache et n'apparaissent pas dans l'historique de navigation.

Transmettre les données via un formulaire

- Dans ce cas, la page1.php doit contenir un formulaire

```
<form method="post" action="page2.php">  
  
<input type="text" name="Nom">  
<input type="text" name="Prenom">  
<input type="submit" value="Valider">  
  
</form>
```

Récupérer les données du formulaire

- Dans page 2.php

```
<?php
if ( isset($_POST ['Nom'] ) and isset ( $_POST ['Prenom'] ) ) {

echo "Bonjour". $_POST ['Nom']. "  ". $_POST ['Prenom'];  

}  

?>
```

Rq: si on avait utilisé get dans l'attribut méthode, dans ce cas il fallait écrire **\$_GET['Nom']** et **\$_GET['Prenom']**

Redirection vers une autre page

- Pour se rediriger d'une page à une autre sans lien et sans formulaire on peut utiliser la fonction **header()**.
- Cette fonction envoie des **entêtes HTTP** au serveur. Pour effectuer une redirection, il faut envoyer l'entête **Location** : suivi de **l'URL de la nouvelle page**.

Exemple de redirection

- Dans la page 1.php on écrit le code suivant pour se rediriger vers la page2.php:

```
header("Location:page2.php");
```

Transmettre les données via l'en-tête http

- On peut transmettre des valeurs lorsque on se redirige vers une autre page.
- **Exemple:**

```
header("Location:page2.php?p1=mohamed&p2=12");
```

Ou

```
$x=12;
```

```
$y=mohamed;
```

```
header("Location:page2.php?p1=$y&p2=$x");
```