

# Exercice 1

## 1.

*form1.php*

```
<?php include 'html5_page.php'; begin_html('Formulaire Saisie'); ?>
<br/><br/>
<h3>Formulaire de saisie</h3>
<form method="post" action="form2.php">
<p>Prenom : <input type="text" name="prenom"></p>
<p>Nom : <input type="text" name="nom"></p>
<p><input type="submit" value="Valider"></p>
</form>
<?php end_html(); ?>
```

*form2.php*

```
<?php include 'html5_page.php'; begin_html('Formulaire Données'); ?>
<!-- page PHP affichant les donnees saisies dans le formulaire
precedent -->
<h3>Affichage des variables provenant du formulaire</h3>
<p>Votre prenom est : <?php echo $_POST["prenom"] ?> </p>
<p>Votre nom est : <?php echo $_POST["nom"] ?></p>
<?php end_html(); ?>
```

## 2.

La méthode *POST* est généralement utilisée pour envoyer des informations (du client vers le serveur).

## 3.

### a)

La balise `<form>` permet de générer un formulaire en html, pouvant contenir plusieurs champs à remplir.

### b)

L'attribut `method` permet de définir comment (par quelle méthode) seront envoyées les données.

### c)

L'attribut `action` définit vers quelle page la requête doit être envoyée.

### d)

L'élément `input` de type `text` est un champ d'entrée où l'utilisateur doit entrer des données de type *texte*.

### e)

L'élément `input` de type `submit` est un bouton qui effectuera la requête une fois actionné.

**f)**

La valeur de la variable `$_POST["prenom"]` après l'exécution de `form2.php` est la valeur entrée par l'utilisateur dans `form1.php`, dans l'élément `input` ayant pour attribut `name="prenom"`.

## Exercice 2

**1.**

```
<?php include 'html5_page.php'; begin_html('URL envoi de données'); ?>
<br><br>
<h3>Lien comportant des variables</h3>
<p>
  <a href="querystring2.php?prenom=Paul&nom=Durand&pays=France">
    Paul Durand France
  </a>
</p>
<?php end_html(); ?>
```

```
<?php include 'html5_page.php'; begin_html('Récupération Données URL'); ?>
<!-- Recuperation des variables passees dans l'URL de la page -->
<br><br>
<p>
  La variable <strong>prenom</strong> contient :
  <?php echo $_GET["prenom"] ?>
</p>
<p>
  La variable <strong>nom</strong> contient : <?php echo $_GET["nom"] ?>
</p>
<p>
  La variable <strong>pays</strong> contient : <?php echo $_GET["pays"] ?>
</p>
<?php end_html(); ?>
```

**2.**

La méthode *GET* est généralement utilisée pour récupérer des informations (du serveur vers le client).

**3.**

**a)**

La partie de l'URL appelée *query* ou *querystring* est la partie à la fin, elle commence par un `?` après l'adresse de la page.

**b)**

Le caractère permettant de séparer les paires `nom=valeur` formant une *query* est **&**.

**c)**

Les lignes 5 à 7 de *querystring1.php* constituent un lien qui redirige vers la page *querystring2.php* avec 3 paires de paramètres, `prenom=Paul` , `nom=Durand` et `pays=France` .

**f)**

Lors de l'exécution de *querystring2.php*, la valeur de `$_GET["prenom"]` est `Paul` .

## Exercice 3

### 1.

*cookie1.php*

```
<?php
    $compteur = isset($_COOKIE['compteur']) ? $_COOKIE['compteur'] + 1 : 1;
    setcookie("compteur",$compteur);
?>
<?php include 'html5_page.php'; begin_html('URL envoi de données'); ?>
<!-- code HTML après l'envoi des Cookies -->
Vous avez effectué
<?php echo($compteur . ($compteur==1 ? " visite " : " visites "));?>
sur cette page
<?php end_html(); ?>
```

### 2.

**a)**

`$_COOKIE` est une variable superglobale, sous forme de tableau, contenant les références aux cookies HTTP.

**b)**

Au démarrage d'un script PHP, la variable `$_COOKIE` ne contient rien.

**c)**

- Ligne 2

On vérifie si le cookie existe déjà, si oui on l'incrémente de 1, sinon on le définit à 1.

- Ligne 3

On actualise le cookie dans le navigateur.

- Ligne 8

On affiche le contenu du cookie `$compteur` , qui contient le nombre de visites de la page.

### 4.

**a)**

Lors de la première exécution de *cookie1.php*, la variable `$_COOKIE['compteur']` ne contient rien.

**b)**

Lors de la première exécution de *cookie1.php*, le nom du cookie créé est `compteur` et sa valeur est `1`.

**c)**

Lors de la deuxième exécution, la variable contient la valeur `2`.

## Exercice 4

**1.**

*cookie2.php*

```
<?php //soumission du formulaire, setcookie si demandé
if (isset($_POST['remember']) && isset($_POST['nom'])) {
    setcookie("nom", $_POST['nom'], time() + 90 * 86400);
    // cookie pour pré-remplir le formulaire la prochaine fois,
    // expire dans 90 jours
}??
<?php include 'html5_page.php'; begin_html('Récupération Données URL'); ?>
comment
// Si $_POST contient des valeurs le formulaire a été soumis sinon c'est le 1er
affichage de la page il faut afficher le formulaire.
<?php if (isset($_POST['nom'])): ?>
    Bienvenue <b><? echo $_POST['nom'] ?></b>
<?php else: ?>
<form method="post"> // attribut manquant ?
    Indiquer votre nom :
    <input type="text" name="nom"
    value=""> <br>
    <input type="checkbox" name="remember" checked="checked" value="cookie">
    Retenir mon nom <br>
    <input type="submit" value="OK">
</form>
<?php endif; ?>
<?php end_html(); ?>
```

**2.**

**a)**

Lors de la première exécution, `$_POST['nom']` n'est pas définie.

**b)**

Le code HTML qui sera écrit dans la sortie standard lors de la première exécution est le code compris entre la ligne **15** et la ligne **22**.

### c)

Dans les lignes 2 et 3, le test renverra `vrai` et un cookie sera créé, avec comme nom `nom`, comme valeur `Toto`, avec une date d'expiration de 90 jours.

### d)

Lors de la deuxième exécution du script, le code HTML qui sera écrit dans la sortie standard sera `Bienvenue <b>Toto</b>`.

## 3.

*cookie2.php*

```
<?php //soumission du formulaire, setcookie si demandé
if (isset($_POST['remember']) && isset($_POST['nom'])) {
    setcookie("nom", $_POST['nom'], time() + 90 * 86400);
    // cookie pour pré-remplir le formulaire la prochaine fois,
    // expire dans 90 jours
}??
<?php include 'html5_page.php'; begin_html('Récupération Données URL'); ?>
comment
// Si $_POST contient des valeurs le formulaire a été soumis sinon c'est le 1er
affichage de la page il faut afficher le formulaire.
<?php if (isset($_POST['nom'])): ?>
    Bienvenue <b><? echo $_POST['nom'] ?></b>
<?php else: ?>
<form method="post"> // attribut manquant ?
    Indiquer votre nom :
    <input type="text" name="nom"
    value="<? isset($_COOKIE['nom']) ? echo $_COOKIE['nom'] : "" ?>"> <br>
    <input type="checkbox" name="remember" checked="checked" value="cookie">
    Retenir mon nom <br>
    <input type="submit" value="OK">
</form>
<?php endif; ?>
<?php end_html(); ?>
```

## Exercice 5

### 1.

*session.php*

```
<?php
session_start();
if (!isset($_SESSION['count'])) {
    $_SESSION['count'] = 1;
```

```

} else {
    $_SESSION['count']++;
}
?>
<?php include 'html5_page.php'; begin_html('Session'); ?>
<h2>Script 5.5 : Session</h2>
<?php
echo 'Vous avez vu cette page ', $_SESSION['count'], ' fois<br>';
if ($_SESSION['count'] == 4) {
    unset($_SESSION['count']);
}
?>
<a href="">Recharger de la page</a>
<?php end_html(); ?>

```

## 2.

- Ligne 1

Crée une session / restaure la session retrouvée trouvée sur le serveur.

- Ligne 2

On vérifie si la variable `$_SESSION['count']` est définie.

- Ligne 3

On définit la variable `$_SESSION['count']` à `1`.

- Ligne 5

On incrémente la variable `$_SESSION['count']` de `1`.

- Ligne 12

On affiche le nombre de fois où la page a été vue dans la même session.

- Ligne 13

On vérifie si la session a été ouverte 4 fois.

- Ligne 14

On "détruit" la variable `$_SESSION['count']`.