
	Activité pratique	
	PHP	
	découverte des langages du WEB	

Objectifs

- Découvrir les bases du langage PHP
- Découvrir la notion de script
- Apprendre à réaliser script simple
- Hébergement d'un fichier PHP sur un serveur
- Appréhender les méthodes GET et POST (HTTP)

Support d'activité

Fichiers : Sujet de l'activité

Logiciels : éditeur de texte (Sublime Text) ; navigateur WEB (Google Chrome) ; Client FTP (Filezilla), serveur personnel (voir brief utilisation du serveur personnel du début de formation).

Introduction

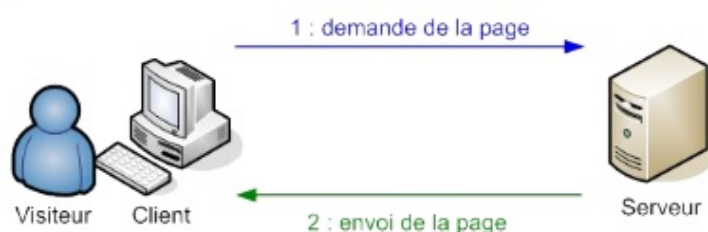
Il est intéressant de pouvoir échanger des informations en interaction avec une page WEB. Par exemple, on pourrait vouloir transmettre des données à une page pour qu'elle les affiche. On pourrait aussi avoir à demander des informations à un utilisateur pour les traiter par la suite (formulaire).

Nous arrivons cependant aux limites du langage HTML car il faut ensuite pouvoir analyser les informations que le visiteur a saisies... et cela ne peut pas se faire en langage HTML. Comme nous allons le voir, le traitement des résultats doit s'effectuer dans un autre langage, par exemple **le PHP**.

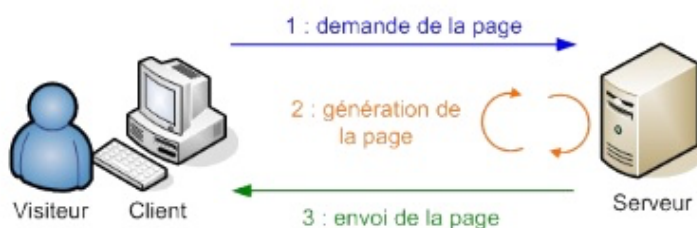
Dans ce TP, nous allons voir comment il est possible de transmettre des données à une page WEB et comment les traiter.

Différence entre HTML et PHP

□ HTML



□ PHP



Lorsque l'on veut transmettre des données à une page WEB, on peut le faire de deux façons : via l'**URL** de la page, ou via un **formulaire**.

Via l'URL

Une des manières les plus simples consiste à faire transiter les données par l'adresse URL (**méthode GET**). Par exemple :

<http://www.test.com/index.html?name1=value1&name2=value2>

- http indique le protocole utilisé (Hyper Hext Hransfer Hrotocol)
- www.test.com est l'adresse du serveur WEB sur lequel le site est stocké
- Index.html est le nom de la page cherchée sur ce serveur
- Le caractère « ? » est utilisé pour indiquer le début des variables envoyées.
- Le caractère « & » permet de séparer les variables envoyées.
- Le caractère « = » permet d'affecter une valeur à chaque variable

Remarques :

✚ Avec cette méthode, le nombre de variables envoyées est illimité bien que la longueur totale de l'URL ne doive pas dépasser 256 caractères. Il faut donc opter pour cette méthode lorsque de petites quantités de données doivent transiter.

✚ Cette méthode n'est pas sécurisée. Elle ne doit alors jamais être utilisée pour faire transiter un mot de passe ou une information sensible.

Autre exemple :

`http://www.monsite.com/bonjour.php?nom=Dupont&prenom=Jean`

Il est donc possible de faire transiter des données entre deux pages :



Remarque : une page PHP peut afficher du contenu HTML. D'ailleurs, un document .html et .php peuvent contenir le même contenu HTML. En revanche, le **PHP** est un langage **SERVEUR** alors que le **HTML** est un langage **CLIENT**. Autrement dit, le PHP doit obligatoirement être exécuté sur un serveur WEB.

Nb : toutes les explications de la méthode GET sont à savoir par cœur.

- Ouvrir un nouveau fichier texte avec Sublime Text, puis insérer le code suivant. Sauvegardez le code sous le nom : ap12.php sur votre ordinateur.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Titre</title>
  </head>

  <body>
    <p>Bonjour

    <?php echo $_GET['prenom'] . ' ' . $_GET['nom']; ?> !

  </p>
</body>
</html>
```

- Ouvrir le fichier ap12.php dans Google Chrome. Vous devriez obtenir le résultat suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Titre</title>
  </head>

  <body>
    <p>Bonjour

    <?php echo $_GET['prenom'] . ' ' . $_GET['nom']; ?> !

  </p>
</body>
</html>
```

⇒ Q1 : Expliquer pourquoi le script PHP ne se lance pas.

- Déplacer le fichier ap12.php dans le répertoire public_html de votre serveur personnel. Pour lancer le fichier, il faut taper son URL dans le client WEB en complétant par les paramètres requis.

⇒ Q2 : proposer un exemple d'URL à rentrer dans Google Chrome pour tester le script.

- Un utilisateur pourrait envoyer un URL avec des variables vides (ex : ?nom=&prenom=) ce qui pourrait causer des problèmes dans la suite du script. Pour vérifier si une variable a été renseignée ou non, on utilise la fonction `isset` :

```
1 <?php
2 if (isset($_GET['prenom']) AND isset($_GET['nom'])) // On a le nom et le prénom
3 {
4     echo 'Bonjour ' . $_GET['prenom'] . ' ' . $_GET['nom'] . ' !';
5 }
6 else // Il manque des paramètres, on avertit le visiteur
7 {
8     echo 'Il faut renseigner un nom et un prénom !';
9 }
10 ?>
```

- On peut également faire des opérations sur des variables :

```
$var_a = $_GET['a'];
$var_b = $_GET['b'];

echo $var_a + $var_b ;
```

- On peut aussi tester les variables :

```
if ($var_a > $var_b)
    echo "a is bigger than b";
```

⇒ Q3 : proposer un script permettant de calculer l'IMC d'un utilisateur qui rentrerait dans l'URL :

- Son poids
- Sa taille
- Son genre (homme ou femme)

Le Sript affiche sur la page WEB :

- L'IMC de l'utilisateur
- Le sexe de l'utilisateur
- L'interprétation de son IMC (<https://www.foodspring.fr/imc>)

Joindre au compte rendu une capture d'écran du code réalisé ainsi que du résultat produit.

Via un formulaire

La deuxième méthode pour envoyer des données à une page WEB est l'utilisation de formulaires.

- Commencer par créer deux nouveaux fichiers texte nommés « formulaire.html et cible.php », puis insérer le code de base d'une page html dans chacun d'eux (voir activité pratique 1.1). Sauvegardez ces fichiers dans le dossier « AP12 - PHP » de WampServer.

Lorsque les informations sont demandées à un utilisateur, il est plus commode de passer par les formulaires. Pour insérer un formulaire sur la page WEB, il faut utiliser les balises html suivantes `<form>` (ne pas copier ce code).

```
1 <p>Texte avant le formulaire</p>
2
3 <form>
4   <p>Texte à l'intérieur du formulaire</p>
5 </form>
6
7 <p>Texte après le formulaire</p>
```

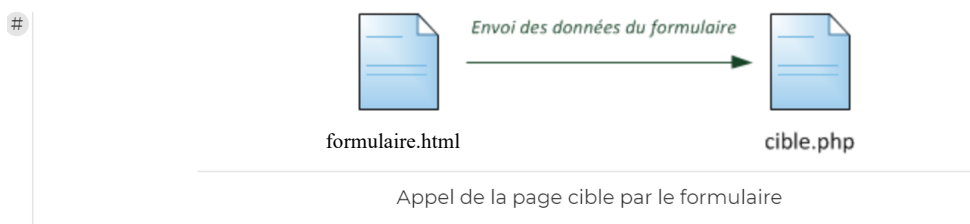
La balise `form` est constituée de deux attributs :

1. **Action** : on y met l'adresse du script qui va traiter les informations
2. **Method** : cet attribut indique par quel moyen les données vont être envoyées (**GET** ou **POST**)
 - a. `method="get"` : les données transiteront par l'URL, comme on l'a appris précédemment. On pourra les récupérer grâce à l'array `$_GET`. Cette méthode est assez peu utilisée car on ne peut pas envoyer beaucoup d'informations dans l'URL (256 caractères max).
 - b. `method="post"` : les données ne transiteront pas par l'URL, l'utilisateur ne les verra donc pas passer dans la barre d'adresse. Cette méthode permet d'envoyer autant de données que l'on veut, ce qui fait qu'on la privilégie le plus souvent. Néanmoins, les données ne sont pas plus sécurisées qu'avec la méthode GET, et il faudra toujours vérifier si tous les paramètres sont bien présents et valides.

La cible

L'attribut `action` sert à définir la page appelée par le formulaire. C'est cette page qui recevra les données du formulaire, et qui sera chargée de les traiter.

Imaginons le schéma de la figure suivante.



Dans le fichier formulaire.html, entrer le code suivant entre les balises `<body>` :

```
1 <p>
2     Cette page ne contient que du HTML.<br />
3     Veuillez taper votre prénom :
4 </p>
5
6 <form action="cible.php" method="post">
7 <p>
8     <input type="text" name="prenom" />
9     <input type="submit" value="Valider" />
10 </p>
11 </form>
```

Puis dans le fichier cible.php, entrer le code suivant entre les balises `<body>` :

```
1 <p>Bonjour !</p>
2
3 <p>Tu t'appelles <?php echo $_POST['prenom']; ?> !</p>
4
5 <p>Si tu veux changer de prénom, <a href="formulaire.html">clique ici</a> pour revenir à la page formulaire.php.</p>
6
7
```

- Testez le script en lançant la page formulaire.html et essayez de comprendre le cheminement des données.

⇒ Q4 : Changer la méthode du formulaire en get puis tester à nouveau la page formulaire.html. A l'aide du message d'erreur affiché, que faut il changer dans le script cible.php ? Faire le changement nécessaire.

⇒ Q5 : Comment sont transmises les données avec cette méthode ? Faites une capture d'écran pour illustrer.

- Dans le formulaire, on peut également créer une liste déroulante :

La figure suivante est une liste déroulante.

Dans quel pays habitez-vous ?

France

France

Espagne

Italie

Royaume-Uni

Canada

Etats-Unis

Chine

Japon

Une liste déroulante

On utilise le code HTML suivant pour construire une liste déroulante :

```
# 1 <select name="choix">
2   <option value="choix1">Choix 1</option>
3   <option value="choix2">Choix 2</option>
4   <option value="choix3">Choix 3</option>
5   <option value="choix4">Choix 4</option>
6 </select>
```

Ici, une variable `$_POST['choix']` sera créée, et elle contiendra le choix qu'a fait l'utilisateur. S'il a choisi « Choix 3 », la variable `$_POST['choix']` sera égale au `value` correspondant, c'est-à-dire `choix3`.

- On peut également créer des cases à cocher :

La figure suivante représente une série de cases à cocher.

☐ Frites
 ☐ Steak haché
 ☐ Epinards
 ☐ Huitres

Cases à cocher

On utilisera le code suivant pour afficher des cases à cocher :

```
1 <input type="checkbox" name="case" id="case" /> <label for="case">Ma case à cocher</label>
```


Là encore, on donne un nom à la case à cocher via l'attribut `name` (ici : « case »). Ce nom va générer une variable dans la page cible, par exemple `$_POST['case']` .

- Si la case a été cochée, alors `$_POST['case']` aura pour valeur « on ».
- Si elle n'a pas été cochée, alors `$_POST['case']` n'existera pas. Vous pouvez faire un test avec `isset($_POST['case'])` pour vérifier si la case a été cochée ou non.

Si vous voulez que la case soit cochée par défaut, il faudra lui rajouter l'attribut `checked="checked"` . Par exemple :

⇒ Q6 : Proposez une page html et un cript PHP permettant de calculer et d'afficher l'IMC d'une personne. La page html doit contenir un formulaire permettant de recueillir les informations suivantes :

- Le poids : avec une entrée de texte simple
- La taille : avec une entrée de texte simple
- Le genre : avec au choix une liste déroulante ou des cases à cocher proposant les choix « femme » et « homme ».

Le Script php lié au formulaire affiche :

- L'IMC de l'utilisateur
- Le sexe de l'utilisateur
- L'interprétation de son IMC (<https://www.foodspring.fr/imc>)
- Il devra permettre de revenir au formulaire.

Joindre au compte rendu une capture d'écran du code réalisé ainsi que du résultat produit.

Bonus

A l'aide du lien suivant, créer un script PHP permettant de stocker les résultats d'un formulaire d'une dizaine de questions dans un fichier texte.

<https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/913492-lisez-et-ecrivez-dans-un-fichier>