

TP 3

Des pages dynamiques : premiers pas en PHP

Internet et Outils (IO2)

Février 2014

1 Préambule

L'affichage d'une page web, en résumé :

1. votre navigateur contacte le serveur qui héberge la page
(ici le serveur est `http://pams.script.univ-paris-diderot.fr/`),
2. le serveur envoie au navigateur du code HTML,
3. le navigateur interprète et affiche le contenu du HTML.

La description HTML envoyée au navigateur est un “produit fini”, figé. On appelle ce type de pages des pages **statiques**.

Le but de ce TP est de produire au contraire des pages **dynamiques**, qui vont varier en fonction du contexte. Il existe différents langages qui permettent l'écriture de telles pages¹. Vous utiliserez ici le langage PHP. PHP² est un langage de script inspiré du shell, réputé simple d'utilisation.

L'écriture d'une page dynamique suit la stratégie suivante :

- vous écrivez une page HTML partielle,
- vous placez dans les “trous” des programmes PHP,
- vous déposez le tout sur le serveur (dans votre `public_html`).

Ensuite, quand un navigateur envoie une requête au serveur pour accéder à votre page, il se passe ceci :

1. les morceaux de code PHP sont exécutés par le serveur (chaque fragment de PHP génère du code HTML),
2. le code HTML produit est intégré dans le HTML principal, la page HTML ainsi reconstruite est envoyée au navigateur,
3. le navigateur interprète et affiche ce qu'on lui a envoyé. La production par PHP du HTML est transparente pour le client.



Comme d'habitude, une partie du travail de votre TP consiste à chercher les bonnes informations sur les ressources en ligne. Outre vos sources habituelles, vous pouvez consulter la documentation officielle de PHP <http://fr.php.net/manual/fr> (en français).

-
1. En fait n'importe lequel le permet en théorie, mais certains sont spécialement dédiés à cette tâche.
 2. Dont le nom signifiait à l'origine Personal Home Page.

2 Le langage PHP : petit guide de survie

PHP est un langage impératif dont certains aspects syntaxiques vous sont déjà familiers. En particulier, les instructions sont séparées par des point-virgules ; et les blocs sont enfermés dans des accolades {}. Les programmes PHP sont délimités par une balise spéciale qui s'ouvre avec `<?php` et se ferme avec `?>`. Pour produire du texte, PHP fournit l'instruction `echo`.

```
<?php
    /* Mon premier programme */
    echo "Hello world";
?>
```

Remarquez aussi le commentaire entre `/*` et `*/`.

Il existe plusieurs manières d'interpréter un programme PHP. Dans cette section, on utilisera l'interprète en ligne de commande `php`, qui prend en argument un fichier PHP et renvoie sur la sortie standard le résultat.

Exercice 1 — *Hello world* Recopiez le code du Hello world précédent dans un fichier `hello.php`, et lancez la commande `php hello.php`. Admirez.

2.1 Chaînes de caractères

Ce qui est affiché/généré par `echo` est une chaîne de caractères. Les chaînes sont définies à l'aide de guillemets simples ou doubles. Si vous souhaitez définir un guillement dans votre chaîne, utilisez le backslash (`\`). On peut concaténer deux chaînes grâce à l'opérateur "point" (`.`). Par exemple, les chaînes suivantes sont strictement identiques :

```
"Un voleur, " . "des valises"
```

```
"Un voleur, des valises"
```

2.2 Variables

En PHP vous pouvez créer des variables dans lesquelles stocker des valeurs pour vous en servir par la suite. À noter cependant, les noms de variables en PHP commencent toutes par le symbole `$`.

PHP est un langage faiblement et dynamiquement typé, ce qui signifie que PHP essaye de convertir automatiquement au type correct en fonction de la valeur stockée par la variable et du contexte. Le type des variables étant dynamique, on ne le déclare pas à leur création. Ainsi on définira des variables par ce genre d'appel très simple :

```
$x = 16;
$x = "bonjour";
$x = True;
```

Observez comment le type de `$x` passe d'entier à booléen de la première à la troisième ligne. Cela peut sembler un avantage, la contrepartie est qu'il est plus facile de faire des erreurs à l'exécution. De plus, la sémantique³ des règles de conversion de types de PHP fait appel à des arcanes définies uniquement par l'implémentation et hautement non uniformes, pouvant conduire à des comportements erratiques et non prédictibles.

La syntaxe `$` est inspirée du shell, et PHP partage avec ce dernier le mécanisme de substitution dans les chaînes :

3. Or lack thereof. L'auteur original de PHP le confesse : « There was never any intent to write a programming language [...] I have absolutely no idea how to write a programming language, I just kept adding the next logical step on the way. »

```
$x = 42;
echo "The answer is $x\n";
echo 'The answer is in $x\n';
```

Exercice 2 — Substitutions Évaluez le code précédent (dans un fichier `subst.php` et en rajoutant la balise `<?php ... ?>`). Que constatez-vous ?

2.3 Fonctions

Les fonctions en PHP permettent de factoriser du code, amenant à une meilleure lisibilité et à un maintien plus aisé. La syntaxe pour écrire une fonction est la suivante :

```
function toto($arg_1, $arg_2, /* ..., */ $arg_n)
{
    echo "Une fonction.\n";
    return 42;
}
```

Exercice 3 — Deux et "DEUX" font 5 Écrivez une fonction `add` prenant deux arguments `$x` et `$y` ayant le comportement suivant : `add(1, 2)` affiche `1 + 2 = 3`. Testez la fonction avec des entiers, des chaînes et un mélange des deux.

On vous recommande d'utiliser les fonctions au maximum dans la suite des exercices.

2.4 Tableaux

Concernant les tableaux, à nouveau ce typage faible permet de faire des choses dont vous n'avez pas l'habitude dans d'autres langages. Ainsi un même tableau peut contenir à la fois des entiers, des chaînes, etc.

```
$tableau = array();
$tableau[0] = "toto";
$tableau[2] = "titi";
$tableau[1] = 42;
```

Remarquez que l'on ne spécifie pas la taille d'un tableau en PHP, en effet PHP change automatiquement la taille des tableaux pour qu'il y ait toujours de la place pour des nouveaux éléments.

C'est même un peu plus compliqué que cela, en PHP vous avez plus de liberté : les clés du tableau (la valeur passée entre crochets `[]`) peuvent être de n'importe quel type. On parle de tableaux associatifs. Ainsi, le tableau de l'exemple précédent peut être complété comme suit :

```
$tableau["cle"] = "valeur";
$x = "cle";
echo $tableau[$x];
```

Ce programme affichera la chaîne de caractères `"valeur"`.

Le tableau précédent peut aussi être défini directement par

```
$tableau = array(0=>"toto", 2=>"titi", 1=>42, "cle"=>"valeur");
```

Attention, la conversion de type automatique à la volée peut vous jouer des tours. D'après vous, que fait le code suivant ?

```
$tableau = array(0=>"toto", "0"=>"titi", "00"=>"tutu");
```

Exercice 4 — Tableaux Évaluez les différents morceaux de codes qu’on vient de vous présenter, en mettant des `echo` à des endroits bien choisis pour observer le contenu des tableaux. N’hésitez pas à expérimenter avec des clefs de types différents.

Exercice 5 — Accesseur Écrivez une fonction contenu prenant un tableau `$tab` et une clef `$clef` en arguments et affichant “La valeur associée à `X` est `Y`” où `X` et `Y` sont les valeurs `$clef` et `$tab[$clef]`. Réécrivez l’exercice précédent en utilisant cette fonction au lieu d’utiliser `echo`.

3 Mélanger PHP et HTML

Les programmes PHP s’intègrent dans du code HTML grâce à la balise spéciale précédemment citée `<?php ... ?>`.

```
<?php
/* Un programme PHP qui fabrique du HTML */
echo
```

4 Des pages dynamiques

Exercice 7 — *Ma première page dynamique*

La fonction `date` de PHP permet de récupérer la date courante. Par exemple les appels de fonction `date('l')`, `date('j')`, `date('F')` et `date('Y')` retournent une chaîne de caractères contenant respectivement le jour de la semaine, le jour du mois, le mois et l'année.

1. Allez jeter un œil sur le manuel aux autres informations que peut donner `date`.
2. Créez une page `date.php` contenant le titre “Service de date en temps réel” et un paragraphe annonçant la date courante.
3. Comme vous pouvez le remarquer, la date est affichée en anglais. À l'aide de tableaux associatifs, créez des “dictionnaires” traduisant les jours et les mois de l'anglais au français et servez-vous en pour afficher maintenant la date en français convenable.

5 Des pages interactives

Pour l'instant nos pages PHP ne sont pas capables d'interagir avec l'utilisateur. Objectif ici : que le client (votre navigateur) puisse passer des paramètres au serveur, et que le serveur puisse se servir de ces paramètres dans la construction de la page HTML qu'il lui renvoie. Deux techniques, appelées GET et POST, coexistent pour ceci. Nous nous concentrerons cette semaine sur la première. Le prochain TP reprendra, avec les formulaires, la description des deux techniques.

Le passage de paramètres en mode GET se fait par l'url. Ainsi, au lieu d'avoir une simple adresse

```
page.php
```

on ajoute un certains nombres de couples clé-valeur de la manière suivante

```
page.php?cle1=valeur1&cle2=valeur2&...&clen=valeurn
```

Notez le `?` séparant l'adresse de la page des paramètres, et le caractère `&` séparant les différents couples clé-valeur.

Exercice 8 — *Google is watching you*

1. Faites une recherche avec Google et observez dans la barre d'adresse le passage des paramètres correspondant à votre recherche.
2. Mettez un marque-page sur la page des résultats, et constatez que vos critères de recherche sont conservés.

Récupérer les valeurs fournies dans l'URL PHP peut accéder à ces couples clé-valeur à travers un tableau associatif spécial nommé `$_GET`. Par exemple, si vous demandez l'URL

```
nom.du.serveur.com/page.php?cours=I02&note=16
```

alors le serveur remplit le tableau `$_GET` de la façon suivante

```
$_GET["cours"] = "I02";  
$_GET["note"] = "16";
```

avant d'exécuter le script `page.php` (remarquez bien les guillemets autour du 16, qui indiquent une chaîne de caractères).

Exercice 9 — *I can GET now*

1. Créez une page PHP qui prend les paramètres `ville`, `film`, `matiere` et affiche une phrase (en français) indiquant dans quelle ville vous habitez, ainsi que votre film favori et votre matière favorite.
2. Testez votre page PHP.

3. Recommencez en utilisant une fonction `affiche_message($ville, $film, $matiere)` qui affiche la phrase.
4. Recommencez en utilisant une fonction `message($ville, $film, $matiere)` qui au lieu d'afficher le message, le renvoie sous forme de chaîne de caractères.

Parcourir un tableau Nous aurons besoin dans le prochain exercice d'une autre construction de PHP, qui permet d'itérer sur le contenu d'un tableau : l'instruction `foreach`, dont l'utilisation est la suivante

```
foreach ($tableau as $cle => $valeur) {
    ...
}
```

Le code placé entre les deux accolades sera exécuté une fois pour chaque couple `$cle => $valeur` dans le tableau `$tableau`. On utilise la clé et la valeur courantes grâce aux variables `$cle` et `$valeur`. Ainsi le code suivant

```
$tableau = array(0=>"zero", 1=>"un", "deux"=>2);
foreach ($tableau as $c => $v) {
    echo $c . " " . $v . "\n";
}
```

affichera

```
0 zero
1 un
deux 2
```

Remarquez qu'un retour à la ligne est indiqué par la chaîne `"\n"`, encore une fois comme en JAVA.

Exercice 10 — Vous voyez le tableau ! Écrivez une fonction qui prend en argument un tableau associatif de PHP et génère un tableau HTML listant le contenu du tableau dans deux colonnes clefs / valeurs. Testez-la.

Vous pourrez astucieusement la découper en une première fonction qui génère les balises externes `<table>...</table>` et une autre qui génère chaque ligne `<tr>...</tr>`.

Exercice 11 — GET on the scene

1. Écrivez une page `echo.php` qui présente dans un tableau HTML tous les couples clé-valeur qui lui sont passés par GET. Vous utiliserez pour ce faire la fonction définie à l'exercice précédent.
2. Comme d'habitude, n'oubliez pas de tester et de valider !
3. Au fait, comment valider du PHP ? Appelez votre encadrant de TP pour en discuter.

6 Ma page perso en PHP

Exercice 12 — Page personnelle en PHP - version 0

1. Créez une page d'accueil `bonjour.php` qui contient (par exemple) votre nom.
2. Ajoutez à vos deux pages un menu contenant des liens vers toutes les pages du site (y compris la page courante). Utilisez `<nav>` !

Fonction include Supposons que vous vouliez ajouter une nouvelle page à ce site. Il faut d'abord la créer, puis lui intégrer le menu, et mettre à jour les menus de **toutes les autres pages**. Faire ceci à la main est laborieux, et source d'erreur. Une solution à ce problème est la fonction `include` de PHP, qui permet d'inclure le contenu d'un fichier PHP dans un autre avant son exécution.

Exercice 13 — Page personnelle en PHP - version 1

3. Extrayez le menu de l'une de vos pages, et recopiez-le dans un fichier `menu.php` (sans même mettre d'en-tête).
4. Dans chacune des pages du site, effacez le menu en question et utilisez la fonction `include` pour inclure le menu.
5. Ajoutez une nouvelle page et voyez comme il est plus simple de mettre à jour le menu.

Pseudo frame Nous allons maintenant faire un vrai site dynamique qui prend un paramètre `page`. Ce type d'architecture s'appelle pseudo frame. L'idée est la suivante : tout le contenu est réparti dans des petits fichiers `bonjour.php`, `date.php`, etc., correspondant aux différentes options du menu et qui ne contiennent que le code qui doit s'afficher.

Ensuite, les liens du menu pointeront vers des adresses du type `index.php?page=bonjour` ou `index.php?page=date`. Un accès au tableau `$_GET` va nous permettre de récupérer la valeur de la clé `page` qui sera soit `bonjour` soit `date`, etc., et on pourra afficher le contenu correspondant via `include`.

Exercice 14 — Page personnelle en PHP - version 2

6. Réécrivez votre site de cette façon. Vous aurez donc un fichier PHP pour le menu, un pour chacune des différentes pages du site, et un fichier central `index.php` qui affichera les différents morceaux en fonction du choix de l'utilisateur.
7. Essayez d'accéder à la page `index.php` sans préciser de paramètre `page`. Que constatez-vous ?

Conditions On aimerait que par défaut, en cas de demande sans paramètre de la page `index.php`, la page `bonjour` s'affiche. Il faudra pour cela utiliser un branchement conditionnel, qui fonctionne en PHP comme en JAVA :

```
if (condition1) {  
    ...  
} else if (condition2) {  
    ...  
} else {  
    ...  
}
```

Exercice 15 — Page personnelle en PHP - version 2'

8. Modifiez votre page `index.php` pour obtenir ce comportement par défaut. Vous pourrez utiliser la fonction `isset`, qui permet de savoir si une variable a une valeur ou non.

Cette utilisation du passage de paramètres par GET peut permettre à des visiteurs malveillants d'inclure n'importe quoi dans votre page `index.php`, y compris du code malveillant ou du code accédant à des fichiers supposés protégés de votre site. Nous allons donc pour finir limiter l'utilisation de ces paramètres dans notre site.

Exercice 16 — Page personnelle en PHP - version 2' sécurisée

9. Essayez d'afficher dans votre navigateur la page d'adresse suivante : `index.php?page=http://www.facebook.com/login`. Que se passe-t-il ?
10. Créez un tableau PHP contenant les valeurs autorisées pour le paramètre `page`. Par exemple, la clé `bonjour` sera associée à la page `bonjour.php`. Faites en sorte que la page `index.php` ne donne accès qu'aux pages autorisées, et fasse retomber sur la page `bonjour` toutes les requêtes non valides.

7 PHP et formulaires

PHP serait peu pratique si la méthode GET se limitait à passer des paramètres à la main dans la barre d'adresse ou via un ``. Les **formulaires** sont la façon la plus puissante de passer des paramètres GET (ou POST, d'ailleurs) au serveur.

Utilisation des balises `<form>` À la fin du TP précédent nous avons évoqués les formulaires. La balise `<form>` possède deux attributs dont on ne s'était pas occupés : `action` et `method`.

- `action` désigne la page PHP à qui la requête va être envoyée,
- `method` vaut soit "GET" soit "POST", et correspond à la façon de transmettre les données lorsque l'on clique sur le bouton Submit du formulaire.

À l'intérieur de la balise `<form>`, les balises `<input name=...>` indiquent quelles données sont passées à la page. Ainsi, avec le formulaire suivant

```
<form action="page.php" method="GET">
  <input type="text" name="champ"/>
  <input type="submit" value="Envoyer"/>
</form>
```

lorsque l'utilisateur remplit le champ de texte (par exemple avec le texte "essai") et clique sur le bouton Envoyer, le navigateur va produire une requête de la forme

`http://.../page.php?champ=essai`

(Si on avait choisi `method="POST"` une requête pour la page `page.php` aurait été générée et le couple `champ=essai` aurait été passé de façon cachée à la variable `$_POST`.)

Exercice 17 — Infos sur l'utilisateur

1. Créez une page `infos.html` avec un formulaire simple demandant le nom, l'âge et le sexe de la personne qui veut se connecter et un bouton pour envoyer ces informations. Cliquer sur le bouton appellera une page `echo.php` exploitant ces informations. Elle affichera alors :
Bonjour monsieur ou madame (suivant le cas) suivi du nom et de l'année de naissance.
Exemple : Bonjour monsieur Untel, vous êtes né en 1989.
2. Testez avec les méthodes GET et POST et vérifiez que tous les paramètres sont bien passés.
3. Insérez cette page parmi vos pages de présentation personnelle.