

TP n° 3

Premiers Pas en PHP

En plus du cours, on pourra consulter <http://fr.php.net/manual/fr> (en français).

1 Premiers programmes

Dans un premier temps, nous allons utiliser le php tout seul, sans html et sans serveur.
Un programme basique sera donc

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

Il s'interprétera en utilisant la commande php. Si vous utilisez XAMPP, le programme php se trouve dans le sous-répertoire php/ de votre répertoire d'installation XAMPP.

Exercice 1 Enregistrez ce code sous ex1.php, et testez-le avec `php ex1.php`.

Exercice 2 Que se passe-t-il quand on interprète ce code :

```
$x = 42;
echo "The answer is $x\n";
echo "The answer is $x+5\n";
echo 'The answer is in $x\n';
echo "\n";
echo "TRUE = ", true, "\n";
echo "FALSE = ", false, "\n";
```

2 Mélanger PHP et HTML

Exercice 3 On se place maintenant dans le répertoire `public_html`, ou un de ces sous-répertoires.

- Dans un premier temps, créez un fichier `hello.php`, qui reproduit la page ci-dessous en ne mettant que du code HTML.

Premier Test PHP

Bonjour je m'appelle Gérard Dupré. Bienvenue!

- Accédez à cette page via le serveur (adresse `http://pams.script.univ-paris-diderot.fr/...`).
- Maintenant, dans une inclusion PHP, on va déclarer deux variables `$prenom` et `$nom`, initialisées à votre guise. Et, on affiche ces variables à l'endroit approprié avec une autre inclusion PHP. Testez de nouveau. Que se passe-t-il lorsqu'on affiche la page source de la page (avec `Ctrl-U`, par exemple) ?
- On peut ne faire l'inclusion que pour écrire les noms et prénoms ou au contraire pour écrire tout le paragraphe "Bonjour je m'appelle Gérard Dupré. Bienvenue!", balises de paragraphe comprises.
Testez les diverses solutions avec des chaînes avec guillemets simples et doubles.

- Rapatriez au moins deux feuilles de styles d'anciens TP et appelez-les `style1.css`, `style2.css`, ... Maintenant déclarez une variable entière égale à 1, 2, ..., en même temps que le nom et le prénom. Et utilisez cette variable pour choisir quelle feuille de style inclure. Quelle est l'inclusion PHP, la plus petite que vous pouvez trouver ?
- Ajoutez maintenant la date et l'heure courantes dans votre page. (fonction `date`). Regardez dans le cours et dans la doc <http://fr.php.net/manual/fr> pour trouver le bon format. On peut envisager un affichage du genre Date: 12/02/2020, Heure 13:37.

Exercice 4 — On ajoute de l'aléatoire Regardez la documentation de la méthode `rand`.

1. La fonction `rand($x,$y)` renvoie un entier compris entre `$x` et `$y`. Utilisez cette fonction pour créer une fonction `lancer_de()` simulant un lancer de dé. On affichera juste le résultat du lancer de dé, avec éventuellement un petit texte d'accompagnement. Chaque rafraîchissement de la page fera un nouveau lancer.
2. En utilisant de nouveau la fonction primitive `rand($x,$y)`, écrivez une page `meteo.php` qui affiche une des images `meteo/meteo$i.png` (où `$i` est un entier) disponibles sur Moodle dans l'archive `meteo.zip`.
3. À l'aide d'un tableau, donnez une légende pour les pictogrammes : ce sera donc un tableau à deux colonnes : une avec le pictogramme et l'autre avec une explication du genre "Grand soleil", "Pluie", ... (Vous pouvez ne faire que les 3 premières légendes si vous n'avez pas le temps.)

Exercice 5 — Encore de l'aléatoire et retour sur le CSS On veut maintenant faire un tableau de 2×2 (ou 3×3) cases. Toutes les cases vont comporter comme valeur pour l'attribut `class`, `case1`, `case2` ou `case3` qui sera tiré aléatoirement par PHP.

Faites ensuite un CSS qui en plus de donner une taille uniforme aux cases, va choisir la couleur de fond de la case pour chacune des classes. (Choisissez les trois couleurs à votre convenance).

3 Un peu plus sur les variables

Exercice 6 Dans un programme, déclarez 4 variables (`$n`, `$d`, `$s`, `$b`) en leur donnant respectivement les valeurs 3, 3.25, "bonjour" et `true`.

Avec la fonction `echo` affichez leur valeurs. Puis utilisez la fonction `gettype` pour afficher leur types.

Que se passe-t-il si on fait la même chose pour une variable `$x` déclarée sans valeur ou une variable `$y` non déclarée ?

Que devrait-il se passer si on affecte à `$n` la valeur "hello" ? Vérifiez.

Exercice 7 — Débogage de variables Comme d'autres langages de script, PHP utilise un typage dynamique faible pour ses variables. De ce fait, PHP ne nécessite pas de déclaration de type ni d'initialisation pour manipuler des variables. Cette spécificité affectera la sécurité de vos scripts.

Le langage vous fournit plusieurs fonctions pour déboguer les variables à l'exécution :

- `print_r()` affiche des informations à propos d'une variable, de manière à ce qu'elle soit lisible.
- `var_dump()` (et `var_export()`) affiche les informations structurées d'une variable, y compris son type et sa valeur. Les tableaux et les objets sont explorés récursivement, avec des indentations, pour mettre en valeur leur structure.

Testez le script ci-dessous (fichier `debogage_var.php` sur le moodle). Qu'indiquent les valeurs affichées entre parenthèses pour array et string ?

```
<?php
    $booléen=true; // un booléen
    $nbr_i= 10; //un nombre entier
    $nbr_r= 3.141; //un nombre reel
    $str="hello"; //une chaine de caracteres (string)
```

```

echo "Quelques variables affichees avec var_dump() :<br />";
echo "<pre>";
var_dump($booléen); var_dump($nbr_i); var_dump($nbr_r); var_dump($str);
echo "</pre>";echo "<br />";

$jour=array('dimanche','lundi','mardi','mercredi','jeudi','vendredi');
$jour[6]="samedi";// j'avais oublie samedi !
echo "Un tableau avec print_r() :<br />";
echo "<pre>";print_r($jour); echo "</pre>";print_r($jour);
echo "<br /><br />";
echo "Un tableau avec var_dump() :<br />";
echo "<pre>";var_dump($jour); echo "</pre>";
var_dump($jour);
?>

```

4 Boucles et fonctions

Exercice 8 — Fonctions simples

1. Écrire une fonction `ligne` prenant en argument un entier `$n` et un caractère `$c`, et renvoyant une chaîne de caractère composée de `$n` fois le symbole `$c`. Vous pourrez pour cela utiliser les boucles sur les entiers, dont la syntaxe est identique à celle de Java. Par exemple, l'exécution du programme suivant affiche 0123456789 :

```

<?php
for ($i=0; $i<10; $i++)
{
    echo $i;
}
?>

```

2. En utilisant la fonction précédente, écrivez maintenant une fonction qui prend en entrée un entier `$n`, et affiche une pyramide de signes `*` de hauteur `$n`. Par exemple, pour $n = 3$, vous afficherez

```

*
**
***

```

3. Modifier la fonction précédente, de sorte que la pyramide ne soit construite que si l'entier donné en paramètre est plus petit que 10. Elle renverra un message d'erreur dans le cas contraire.
4. En utilisant deux boucles, écrivez une fonction qui prend un entier n inférieur à 10 et affiche pour $n = 3$ par exemple :

```

*
**
***
**
*

```

5. Réécrivez cette même fonction mais en utilisant `foreach` à la place de `for` cette fois-ci. `foreach` est à préférer systématiquement à `for` dans la mesure du possible car, sémantiquement, elle rend mieux compte de la volonté du programmeur qui est souvent d'itérer sur les éléments d'un domaine (par exemple un tableau) qui n'est pas forcément une séquence d'entiers. Elle permet ainsi d'éviter nombre d'erreurs comme l'accès invalide à des éléments d'un tableau (par exemple accéder au 5^{ème} élément d'un tableau à 3 éléments). Vous pourrez utiliser la fonction `range`¹ pour créer un tableau représentant un intervalle d'éléments.

1. <http://php.net/manual/fr/function.range.php>