

TP n°1 PHP

LES BASES

1. Premier programme
2. Affichage
3. Les variables scalaires
4. Les chaînes de caractères
5. Les tableaux
6. Structures de contrôle

Exercice n°1: Premier programme

Rappels :

Les scripts php doivent être interprétés par un *parser* PHP qui se trouve sur un serveur WEB. Vous utiliserez le serveur Apache installé dans la section.

Tous vos scripts devront donc être :

- copiés dans votre espace web personnel (`/home/etudiantsx/votre_login/www`),
- portés l'extension `.php` et
- possédés les droits nécessaires pour y accéder en tant que client anonyme.

Pour plus de clarté, créer un répertoire `tp_php` pour y stocker vos scripts PHP.

Créer le script `hello.php` qui contiendra le source fourni ci-dessous et tester le. Que se passe-t-il si on met l'extension `.html` à ce script ? Commenter.

```
<HTML><HEAD><TITLE>Premier exemple en PHP</TITLE></HEAD>
<BODY><?php echo "<H1>Hello world !</H1>" ; ?></BODY>
</HTML>
```

Généralement, pour tester la présence d'un *parser* PHP et en connaître sa configuration, on réalise le script suivant. Créer le script `test.php` avec le code suivant et tester le.

Quelle est la version du *parser* PHP installé sur le serveur ?

Pourquoi n'est-il pas indispensable d'indiquer les balises HTML habituelles ? Commenter.

```
<?php phpinfo() ; ?>
```

Exercice n°2: Affichage

Il est possible de spécifier une variable de type chaîne de caractères de la façon suivante :

```
$personne = 'M. Smith' ; // est une chaîne de caractères,  
$personne = "M. Smith" ; // est aussi une chaîne de caractères.
```

Dans le deuxième cas, si la chaîne contient des noms de variables, celles-ci seront remplacées par leur valeur :

```
$type = "M. " ;  
$nom = "Smith" ;  
$personne = "$type $nom" ; // donnera "M. Smith«
```

Ceci est régulièrement utilisé dans la fonction **echo**.

Quand on utilise les "... " on doit donc **échapper** certains caractères avec un *backslash* (\) pour pouvoir les afficher comme tels : le dollar \\$, les doubles quotes \« , le backslash \\.

De même, il existe des caractères spéciaux qui nécessitent d'être échappés : nouvelle ligne \n, retour à la ligne \r, tabulation \t.

Ecrire un script **message.php** qui :

- affecte la chaîne de caractère "Hello world !" à la variable `$message`
- affiche le contenu de la variable dans le texte suivant :

La variable `$message` contient la chaîne de caractère "Hello world !"

Exercice n°3: les variables scalaires

Ecrire un script `variable.php` qui vous permet de tester l'utilisation des variables scalaires. Pour cela, vous devez :

- utiliser différents types de variable scalaire,
- afficher leurs contenus,
- afficher leurs types en utilisant les fonction décrites ci-dessous.

Pour déterminer le type d'une variable on peut utiliser les fonctions suivantes :

- `gettype()` : retourne une chaîne décrivant le type de la variable passée en argument (*integer, double, string, array, object, unknown type*),
- `is_long()/is_int()/is_integer()` : retourne `true` si la variable passée en argument est un entier et `false` dans le cas contraire,
- `is_double()/is_float()` : idem mais avec les flottants/doubles,
- `is_string()` : idem mais avec les chaînes de caractères,
- `is_array()` : idem mais avec les tableaux,
- `is_object()` : idem mais avec les objets.

Remarque : Pour faire du débogage, on utilise très souvent la fonction `var_dump()` qui permet de visualiser le contenu d'une variable.

Exercice n°3: suite

Compléter alors le tableau suivant :

Instruction	Type
<code>\$variable = 0;</code>	
<code>\$variable = 12;</code>	
<code>\$variable = 0.0;</code>	
<code>\$variable = 12.0;</code>	
<code>\$variable = "5.0";</code>	
<code>\$variable = `toto`;</code>	
<code>\$variable = "Bonjour tout le monde";</code>	

Exercice n°4: les chaînes de caractères

Lorsqu'une chaîne de caractères est évaluée comme une valeur numérique, les règles suivantes s'appliquent :

- la chaîne est de type flottant/double si elle contient '.', e ou E sinon elle est de type entier ,
- la valeur est définie par la première partie de la chaîne (0 si c'est du texte),
- lorsque la première expression est une chaîne, le type de la variable dépend de la seconde expression.

Ecrire un script `chaîne.php` qui vous permet de tester les exemples suivants :

```
$res = 1 + "4.5" ; // $res = (type: )
$res = 1 + "-1e3" ; // $res = (type: )
$res = 1 + "titi + 149" ; // $res = (type: )
$res = 1 + "149 + titi" ; // $res = (type: )
```

Remarque : l'opérateur de concaténation de chaînes de caractères est le . (point).

Exercice n°5: les tableaux

On peut connaître le nombre d'éléments d'un tableau grâce aux fonctions :

`sizeof()` : retourne le nombre d'éléments d'un tableau, ou

`count()` : retourne le nombre d'éléments d'un tableau s'il existe, 1 si la variable n'est pas un tableau et 0 si la variable n'existe pas.

Chaque tableau entretient un pointeur courant qui sert à naviguer en son sein grâce aux fonctions :

`reset()` : place le pointeur interne sur le **premier élément** et retourne sa valeur,

`current()` : retourne la valeur de l' **élément courant** ,

`next()` : place le pointeur interne sur l' **élément suivant** et retourne sa valeur,

`prev()` : place le pointeur interne sur l' **élément précédent** et retourne sa valeur,

`each()` : retourne la **paire clé/valeur courante** du tableau et avance le pointeur sur l'élément suivant (c'est la seule fonction qui ne retourne pas faux si l'élément vaut 0 ou "").

Un tableau peut être trié en utilisant les fonctions suivantes :

`asort()` / `arsort()` : trie le tableau en ordre croissant/décroissant de **valeurs** ,

`ksort()` / `rsort()` : trie le tableau en ordre croissant/décroissant de **clés** ,

`sort()` : trie le tableau en ordre croissant **clés et valeurs** (on perd la correspondance clé/valeur),

`uasort()` / `uksort()` / `usort()` : trie le tableau de la même façon que les fonctions précédentes (u pour *user*) mais avec une fonction de comparaison fournie par l'utilisateur.

La fonction `in_array()` permet de vérifier la présence d'un élément dans un tableau.

Il existe aussi un certains nombres d'autres fonctions pour travailler avec les tableaux. Elles sont préfixées par `array_` (consulter la documentation).

Exercice n°5.1: Les tableaux

Écrire un script `tableau.php` qui permet de visualiser le contenu de ces différents tableaux :

```
// tableau simple :
$tableau[0] = 2002;
$tableau[1] = "BTS IRIS";
$tableau[] = 11.12;
// tableau à 2 dimensions :
$tab[0][0] = 12;
$tab[0][1] = "tv";
$tab[1][0] = 1245.678;
$tab[1][1] = "bye";
// tableau associatif :
$toto["Nom"] = "le nom de famille de Toto...";
$toto["Age"] = 12;
$toto["Adresse"] = "22 rue des bois fleuri";
```


Exercice n°5.2: Les tableaux `array()`

Écrire un script `array.php` qui permet de tester l'utilisation des tableaux avec `array()` :

```
$prenom = array('robert', 'roger', 'germain', 'fernand');  
// Nous les affichons  
echo $prenom[0].' - '.$prenom[1].' - '.$prenom[2].' - '  
    $prenom[3].'  
<br>';  
$prenom[4] = 'félicien';  
echo $prenom[4] .'  
<br>';  
  
$animaux = array(1 => 'chien', 'chat', 'vache', 'cochon');  
// L'affichage commence cette fois à partir de : [1]  
echo $animaux[1].' - '.$animaux[2].' - '.$animaux[3].' - '  
    $animaux[4];
```

Exercice n°5.3: Les tableaux `array()`

Écrire un script `tabhtml.php` qui permet de tester l'utilisation des tableaux suivants en créant un affichage d'un tableau HTML centré et composé de 2 lignes comportant du texte :

```
// ci-dessous nous stockons les balises qui servent à créer les tableaux HTML
$table = array(
    'D_TABLE' => '<table border="0" cellpadding="0"
cellspacing="2" width="90%" bgcolor="yellow">\n',
    'D_ROWS' => '<tr><td>',
    'F_ROWS' => '</tr></td>\n',
    'F_TABLE' => '</table>\n'
);

// cette fois nous stockons les balises d'alignement HTML
$align = array(
    'D_CENTER' => '<div align=center>\n',
    'F_CENTER' => '</div>\n',
    'D_RIGHT' => '<div align=right>\n',
    'F_RIGHT' => '</div>\n'
);
```

Exercice n°6: Structures de contrôle

Écrire un script `prenom.php` qui permet de visualiser le contenu du tableau `$prenom` de l'exercice précédent à l'aide d'une boucle `for()`.

Écrire un script `individu.php` qui permet de visualiser le contenu du tableau `$individu` à l'aide d'une boucle `foreach()`. On affichera les clés et les valeurs associées.

```
$individu = array(
    'Nom'=>'Dupond',
    'Prenom'=>'Martin',
    'Age'=>18,
    'Nationalité'=>'Français',
);
```

Écrire un script `individus.php` dans lequel vous déclarerez un tableau `$individus` qui contiendra d'autres tableaux `$individu`. Vous réaliserez un affichage ce tableau `$individus`.