TP PHP n°1 : les bases

 <tvaira@free.fr>

Sommaire

Les objectifs de ce tp sont d'être capable de découvrir les bases du langage PHP pour réaliser des scripts serveurs pour un site web.

Il existe de très nombreux sites dédiés au PHP ! Il faut au moins l'accès au manuel (notamment pour les fonctions) : www.php.net/manual/fr/index.php

Les bases

Exercice n°1.1 : génération de code HTML

L'objectif de ce premier exercice est de vérifier que la chaîne client/serveur fonctionne correctement en générant du code HTML à partir d'un script PHP.

Question 1. Créer le script à partir du code source fourni ci-dessous. Quelle est l'extension à donner à ce script ?

```
<html>
   <head>
      <title>Exercice PHP 1.1</title>
   </head>
   <body>
    <?php
      echo "Hello world !";
    ?>
   </body>
</html>
```

Exercice 1.1.a

Question 2. Donner l'URL pour tester ce script dans le navigateur.

Question 3. Afficher le code source de la page à partir du navigateur. Pourquoi le code PHP n'apparaît-il pas ?

On peut tout à fait générer du code HTML avec le script PHP, comme ceci par exemple :

```
<?php
$nom = "Robert"; // en PHP, toutes les variables sont préfixées par un dollar ($)
echo "<h1>TP PHP</h1>\n"; // des balises HTML !
echo "<p>Bonjour $nom !<br />...</p>\n"; // encore des balises HTML !
?>
```

Exercice 1.1.b

Question 4. Vérifier l'exécution de ce script dans un navigateur et le code source de la page ainsi générée. Est-ce que le code HTML généré par le script PHP (balise `P` par exemple) a-t-il été envoyé au client ?

Question 5. Quelle est l'utilité des "$\backslash n$" générés par le script PHP ? Quelle est alors l'utilité des balises `"<br>"` ?

Exercice n°1.2 : des informations sur le serveur

Le langage PHP vous fournit la fonction `phpinfo()` qui affiche de nombreuses informations sur la configuration de PHP : options de compilation, extensions, version, informations sur le serveur, et l'environnement (lorsqu'il est compilé comme module), environnement PHP, informations sur le système, chemins, valeurs générales et locales de configuration, en-têtes HTTP et la licence PHP.


```
<html>
   <head>
      <title>phpinfo.php</title>
   </head>
   <body>
    <?php
      phpinfo();
    ?>
   </body>
</html>
```

phpinfo.php

Question 6. Créer le script `phpinfo.php` à partir du code source fourni ci-dessus. Tester dans un navigateur et identifier la version du moteur PHP que vous utilisez.

Conservez ce script car `phpinfo()` est un bon outil de débogage qui permet d'afficher le contenu de toutes les variables **EGPCS** (Environnement, GET, POST, Cookie, Serveur) que l'on utilise tout le temps en PHP.

Question 7. À partir des informations sur les variables PHP affichées par la fonction `phpinfo()`, donner les noms des variables contenant l'adresse IP du serveur et du client.

Les variables PHP affichées par la fonction `phpinfo()` sont appelées des **variables super-globales** en PHP. Depuis la version 4 de PHP, ces variables pré-définies sont stockées dans des **tableaux superglobaux** tels que `$_GET`, `$_POST` et `$_SERVER`, etc. Pour plus d'informations, lisez la section *superglobals* du manuel PHP : www.php.net/manual/fr/language.variables.predefined.php.

Question 8. Écrire un script PHP qui permet d'obtenir l'affichage suivant :

```
L'adresse IP du serveur : 192.168.52.1
Votre adresse IP est : 192.168.52.2
Votre système : Linux
```

Les adresses IP affichées ci-dessus sont fournies à titre d'exemple.

Exercice n°1.3 : 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 déclararation 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.


```php
<?php
$a = true;
echo "<pre>"; var_dump($a); echo "</pre>"; echo "<br />";

$a = 10;
echo "<pre>"; var_dump($a); echo "</pre>"; echo "<br />";

$a = 3.141;
echo "<pre>"; var_dump($a); echo "</pre>"; echo "<br />";

$a = "hello $a";
echo "<pre>"; var_dump($a); echo "</pre>"; echo "<br />";

$a = 'bonjour $a';
echo "<pre>"; var_dump($a); echo "</pre>"; echo "<br />";

$a = array('dimanche', 'lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi');
// 'dimanche' est l'index 0 de ce tableau
echo "<pre>"; var_dump($a); echo "</pre>"; echo "<br />";
?>
```

Exercice 1.3.a

Question 9. Testez le script ci-dessus. Qu'indiquent les valeurs affichées entre parenthèses pour `array` et `string` ?

Question 10. Finalement de quel type est la variable `$a` ?

Les tableaux PHP peuvent contenir des **clés** de type entier (`integer`) et chaîne de caractères (`string`) en même temps. Dans certains langages, ce type de tableaux sont nommés tableau associatif (ou encore hashage ou *hash*) et permettent une association clé-élément.

Les tableaux PHP peuvent aussi être multidimensionnel.

```php
<?php
echo "Un tableau multidimensionnel affiché avec var_export() :<br />";
$a = array(array("a", "b", "c"),array("d", "e", "f"),array("g", "h", "i"));
echo "<pre>"; var_export($a); echo "</pre>";

$jour = array('dimanche', 'lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi', 'samedi');
$date["jour"] = $jour;
$date["mois"] = array('janvier', 'février', 'mars', 'avril', 'mai', 'juin', 'juillet','aout'
    ,'septembre','octobre','novembre','décembre');

echo "Un tableau associatif affiché avec var_dump() :<br />";
echo "<pre>"; var_dump($date); echo "</pre>";
echo "<br />";

// Une exploitation de ce tableau
$j = date("w");
$m = date("n");
```



```php
$message = "Le ".$date["jour"][$j]." ".date("j")." ".$date["mois"][$m-1]."<br />";
echo "Utilisation d'un tableau associatif :<br />";
echo $message;
?>
```

Exercice 1.3.b

Question 11. Testez le script ci-dessus. En utilisant le manuel PHP pour la fonction `date()`, que permettent de faire les paramètres passés en argument de cette fonction ?

Question 12. Écrire un script PHP qui permet d'obtenir l'affichage (en français) suivant :

```
Bonjour,
On est le samedi 16 février 2013 et il est 11:51:36
```

PHP possède un grand nombre de fonctions ! Il vous faut donc constamment utiliser le manuel PHP. Pour les fonctions concernant les dates et heures : www.php.net/manual/en/ref.datetime.php. Par exemple pour formater des dates dans d'autres langues, utilisez les fonctions `setlocale()` et `strftime()`.

Exercice n°1.4 : les fonctions

Soit la fonction PHP suivante :

```php
// retourne vrai si l'année $annee passée en argument est bissextile sinon faux
function estAnneeBissextile($annee)
{
   $estMultipleDeQuatreCent = ( ($annee % 400) == 0);
   $estMultipleDeQuatre = ( ($annee % 4) == 0);
   $estPasMultipleDeCent = ( ($annee % 100) != 0);

   return ( $estMultipleDeQuatreCent || ( $estMultipleDeQuatre && $estPasMultipleDeCent ) );
}
```

Exercice 1.3.b

La règle des années bissextiles :
Le pape Grégoire XIII a mis au point en 1582 un calendrier, appelé encore aujourd'hui la calendrier grégorien. Il introduit les règles de calcul des années bissextiles : tous les 4 ans donc chaque année divisible par 4 (comme 1992 ou 1996). Mais ces règles ne suffisent pas. En effet, lorsqu'il s'agit de la première année d'un nouveau siècle (par exemple 2000, 1900 ou 2100), cette année doit être divisible non pas par 4 mais par 400. Ainsi 1700, 1800 et 1900 ne sont pas des années bissextiles.

Question 13. Écrire un script qui affiche le nombre d'années bissextiles que vous avez vécu depuis votre naissance en indiquant lesquelles.

Exemple de résultat attendu :

```
Vous avez vécu 13 années bissextiles : 1968 1972 1976 1980 1984 1988 1992 1996 2000 2004
    2008 2012 2016
```


Exercice n°2.1 : passage de paramètres dans l'URL

Cet exercice a pour objectif de montrer comment on récupère des données passées en paramètres de l'*url*. Cette technique est très utilisée dans la réalisation d'applications PHP.

Jusqu'à la version PHP 4.2.0, les paramètres passés dans l'*url* étaient automatiquement connus sous forme de variables globales du script destinataire. Mais depuis la version 4.2.0, ce n'est plus le cas à cause du changement de la valeur par défaut (auparavant à `On` et désormais à `Off`) du paramètre `register_globals` du fichier de configuration `php.ini` du serveur. Ce changement impose de recourir désormais aux tableaux dit superglobaux de PHP (`$_GET[]`, `$_POST[]`, etc ...). Ces variables superglobales sont accessibles de partout dans un script php (ne pas mettre `global`).

Le principe est le suivant :

```
paramètre id dans url : exemple.php?id=4

alors dans le script exemple.php : $_GET['id'] sera égal à 4
```

En phase d'apprentissage ou de débuggage, il est recommandé de faire un `var_dump($_GET);`

```php
<?php
// vérifie que le paramètre year n'est pas vide
if(!Empty($_GET["year"]))
{
   // récupère le paramètre year
   $year = $_GET["year"];

   // Attention les paramètres d'url sont passées sous forme de chaîne de caractères
   echo "<pre>Débogage variable year : "; var_dump($year); echo "</pre>";

   // vérifie que le paramètre year est valide
   if(ctype_digit($year)) // cf. is_numeric()
   {
      echo "est-ce-que $year est bissextile ?<br /><br />";
      // est-ce-que cette année est bissextile ?
      // TODO
   }
   else echo "Paramètre year invalide !<br /><br />";
}
else echo "Paramètre year manquant !<br /><br />";

echo "Essayez avec ces paramètres :<br />";
echo "<a href=\"exercice-2-1-a.php?year=\">vide</a><br />";
echo "<a href=\"exercice-2-1-a.php?year=2000\">2000</a><br />";
echo "<a href=\"exercice-2-1-a.php?year=2007\">2007</a><br />";
echo "<a href=\"exercice-2-1-a.php?year=year\">invalide</a><br />";
?>
```

Exercice 2.1.a

Question 14. Testez le script ci-dessus. Que permettent de faire les fonctions `Empty()` et `ctype_digit()` utilisées dans ce script ?

Question 15. Compléter le script ci-dessus afin qu'il affiche toutes les années depuis votre naissance sous forme de lien avec en paramètre l'année en question et qui affichera si cette année est bissextile ou non. Pour cela, utilisez la fonction `estAnneeBissextile()` écrite à l'exercice 1.4.

Exemple d'affichage attendu :

```
2000 est une année bissextile !

1966  1967  1968  1969  1970  1971  1972  1973  1974  1975
1976  1977  1978  1979  1980  1981  1982  1983  1984  1985
1986  1987  1988  1989  1990  1991  1992  1993  1994  1995
1996  1997  1998  1999  2000  2001  2002  2003  2004  2005
2006  2007
```

Contrainte : Les lignes comportent seulement dix liens (années).

Exercice n°2.2 : traitement de formulaire

Cet exercice a pour objectif de montrer comment on récupère des données envoyées par un formulaire. Cette technique est très utilisée dans la réalisation d'applications PHP.

Le principe est le suivant :

```
<form action="exemple.php" method="POST" name="form">
   <input type="hidden" name="id" value="4">
   <input type="submit" value="Envoyer">
</form>

alors dans le script exemple.php : $_POST['id'] sera égal à 4 .
```

En phase d'apprentissage ou de débuggage, il est recommandé de faire un `var_dump($_POST)` ou `var_dump($_GET)`.

```
<?php
<html>
   <head>
      <title>Exercice 2.2.a</title>
   </head>
   <body>
      <?php
         if(!Empty($_POST["nom"]))
         {
            $nom = $_POST["nom"];
            $heure = date("H");
            if($heure >= 18)
                  $message = "Bonsoir $nom,<br />";
            else $message = "Bonjour $nom,<br />";
            echo $message;
         }
         echo "<br />";
      ?>
```



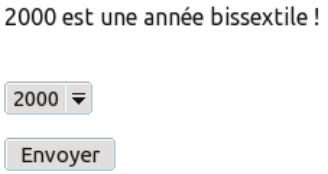
```
      <form action="" method="POST" name="form">
         <input type="text" name="nom" value="">
         <input type="submit" value="Envoyer">
      </form>
   </body>
</html>
?>
```

Exercice 2.2.a

Question 16. Testez le script ci-dessus. Si on change la `method` d'envoi du formulaire (`POST` en `GET`), que faut-il modifier dans le script pour qu'il fonctionne ?

Question 17. Écrire un script qui envoie par un formulaire une année choisie dans une liste déroulante et qui affichera si cette année est bissextile ou non. La liste déroulante contiendra toutes les années depuis votre naissance.

Exemple d'affichage attendu :



Conclusion

Le code *php* se mélange au code HTML et s'exécute côté serveur. PHP est un langage de script libre principalement utilisé pour produire des pages web dynamiques.

Question 18. Proposer une définition de "pages web dynamiques".

L'utilisation de PHP en tant que générateur de pages Web dynamiques est la plus répandue. Elle commence par la récupération de données (*url*, formulaire, ...) en provenance des requêtes du client.

Question 19. Pouvez-vous faire confiance aux données fournies par le client ?

Question 20. Lire l'article fr.wikipedia.org/wiki/php et conclure sur le niveau de fiabilité des applications PHP ?

