Hypertext Preprocessor (PHP)

Jean-Claude Charr

Maître de conférences

IUT NFC

Université de Franche Comté

Description générale

- PHP est un langage de scripts open source conçu pour le développement d'applications web.
- PHP permet d’interagir avec plusieurs systèmes de gestion de bases de données (Oracle, MySQL, etc.)
- Les scripts PHP sont interprétés et exécutés par le serveur web (Apache, IIS (Windows), etc.) et retournent au navigateur du client des pages HTML.
- Les fichiers PHP ont une extension .php et peuvent contenir aussi du texte, des balises HTML et des scripts JS.

Premier exemple

```
<html>
    <body>
        <h2>Premier exemple PHP :<h2>
        <?php
            echo "Hello World";
        ?>
    </body>
</html>
```

Syntaxe

- Un script PHP commence par <?php et termine par ?>
- Les instructions PHP sont séparées par des ;
- Une ligne de commentaire commence par // et les blocs de commentaire sont commencent par /* et terminent par */
- Les variables commencent toujours par un $ et le nom d'une variable ne peut contenir que des caractères alpha-numériques et des _
- Pas de déclaration de variable ou de définition de type
  Ex : $txt= "Hello World";

  $x=1;

Types de données

- PHP propose les types de données suivants : string, int, float, bool, object, NULL et resource.
- Le type d'une variable peut être affiché avec la fonction var_dump().

Exemples :

- $x=2; var_dump($x); // int(2)
- $s="Joe"; var_dump($s); // string(3) "Joe"
- $bool=true; var_dump($bool); // bool(true)
- $t; var_dump($t); // NULL

Changer le type de données (casting)

- Il est possible de convertir le type d'une variable ou d'une valeur en mettant le nouveau type entre parenthèse avant la variable ou la valeur.
- Exemple :

```
$z=(string)2.2;
var_dump($z); // string(3) "2.2"
$z=(float)$z;
var_dump($z); // float(2.2)
```

Opérateurs

- . pour la concaténation de deux chaînes de caractères
- +, -, /, *, %, ++ et -- : opérateurs arithmétiques
- =, +=, -=, /=, *= et %= : opérateurs d'affectations
- ==, !=, <>, >, <, >= et <= : opérateurs de comparaison
- &&, || et ! : opérateurs logiques

Les conditions

- L'instruction if() … elseif() … else

```
if ($d=="Fri")
    echo "Bon weekend!";
elseif ($d=="Sun")
    echo "Bon dimanche!";
else
    echo "Bonne journée!";
```

- L'instruction switch (même syntaxe que JavaScript)

Les tableaux

- Tableau avec index numérique :

```
$cars[5]= "saab";
```

- Tableau avec index associatif :

```
$ages = array("Peter"=>32, "lara"=>30, "Joe"=>34);
$ages['Peter'] = 35;
```

- Tableau multidimensionnel

```
$families = array (
    "Griffin"=>array(
        "Peter",
        "Megan")
) ;
```

Les boucles

- L'instruction while(condition){Instructions à exécuter}
- L'instruction do{Instructions à exécuter} while(condition)
- L'instruction for(initialisation;condition;incrémentation) {instructions à exécuter}
- L'instruction foreach( ){Instructions à exécuter}

```
$x=array("one","two","three");
foreach ($x as $value){
    echo $value . "<br />";
}
```

Parcourir un tableau associatif

```
$car = array("brand"=>"Ford",
"model"=>"Mustang", "year"=>1964);


foreach ($car as $x => $y) {
  echo "$x: $y <br/>";
}
```

Fonctions

- PHP contient beaucoup de fonctions prédéfinies pour manipuler les objets, les chaînes de caractères, etc. Référence : https://www.php.net/manual/fr/funcref.php
- Il est possible de définir de nouvelles fonctions :

```
<?php
    function add($a,$b){
        return $a+$b;
    }
    echo "1 + 16 = " . add(1,16);
?>
```

Arguments et valeur de retour typés

```php
<?php declare(strict_types=1); // typage strict
    function add(float $a, float $b) : float {
        return $a + $b;
    }
    echo "1 + 16 = " . add(1,16);
?>
```

Traiter les données insérées dans un formulaire

- Formulaire HTML :

```
<form action="ex.php" method="post">
    Name: <input type="text" name="fname"/>
    Age: <input type="text" name="age" />
    <input type="submit" />
</form>
```

- Fichier ex.php :

```
<html> <body>
    Welcome <?php echo $_POST["fname"]; ?>!<br />
    You are <?php echo $_POST["age"]; ?> years old.
</body></html>
```

Différence entre POST et GET

- Avec GET les données sont ajoutées à l'url et sont visibles par tout le monde
- Avec GET, la taille des données est limitée à 100 caractères
- Avec POST, les données sont cachées et il n'y a pas de limitation sur la taille des données
- Les données envoyées par POST sont stockées dans la variable $_POST et celles envoyées par GET sont stockées dans la variable $_GET
- La variable $_REQUEST permet de récupérer les données envoyées par GET et POST

Envoyer un fichier au serveur (1/2)

- Fichier HTML :

```
<html>
    <body>
        <form action="upload_file.php" method="post"
        enctype="multipart/form-data">
            <label for="file">Filename:</label>
            <input type="file" name="file" id="file" /><br />
            <input type="submit"  value="Submit" />
        </form>
    </body>
</html>
```

Envoyer un fichier au serveur (2/2)

- Fichier PHP :

```
<?php
    if ($_FILES["file"]["error"] > 0)
        echo "Error: " . $_FILES["file"]["error"] . "<br />";
    else{
        echo "Upload: " . $_FILES["file"]["name"] . "<br />";
        echo "Type: " . $_FILES["file"]["type"] . "<br />";
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
        echo "Stored in: " . $_FILES["file"]["tmp_name"];
    }
?>
```

Les filtres

- Pour tester si les données en entrées sont valides
- Les données en entrées peuvent provenir des formulaires, cookies, web services, base de données

```
<?php
    if(!filter_has_var(INPUT_GET, "email"))
        echo("Le courriel n'est pas défini");
    else
 if (!filter_input(INPUT_GET, "email", FILTER_VALIDATE_EMAIL))
            echo "E-Mail is not valid";
        else
            echo "E-Mail is valid";
?>
```

Les filtres : https://www.php.net/manual/en/filter.filters.validate.php

Cookies

- Un cookie est un fichier sauvegardé par un site sur l'ordinateur du client. Les cookies d'un site sont envoyés dans chaque requête HTTP à ce site.

- Créer un cookie pour une heure : setcookie("user", "Alex" ,time()+3600); //avant la balise html
- Récupérer le cookie "user" : echo $_COOKIE["user"];
- Tester si le cookie "user" est défini : isset($_COOKIE["user"])
- Supprimer un cookie : setcookie("user", "", time()-3600);

Sessions (1/3)

- Permet de passer les données d'un utilisateur d'une page web à l'autre.
- Elle sauvegarde temporairement les données d'un utilisateur sur le serveur. Les données sont supprimées dés que la session expire.
- Pour une sauvegarde permanente des données utilisateur, il faut utiliser une base de données.
- Les sessions créent un identifiant unique (UID) pour chaque visiteur et lient les données de l'utilisateur sauvegardées sur le serveur à l'UID.

Sessions (2/3)

- Les UID des sessions peuvent être propagées d'une page à l'autre en utilisant :
  1. Les cookies (l'UID est sauvegardé dans un cookie et chaque page récupère l'UID lors du chargement)
  2. L'URL (l'UID est passé comme paramètre dans l'URL de la page destinataire)
- Pour commencer une session : session_start()
- Pour détruire une session : session_destroy()
- Pour stocker des données : $_SESSION['nomVar']=$data
- Pour libérer une variable : unset($_SESSION['nomVar'])

Sessions (3/3)

Exemple :

```
<?php
    session_start();
    if (empty($_SESSION['count'])) {
        $_SESSION['count'] = 1;
    } else {
        $_SESSION['count']++;
    }
?>
<p> Bonjour, vous avez visité cette page <?php echo $_SESSION['count']; ?> fois. </p>

<p> Pour continuer, <a href="nextpage.php?<?php echo htmlspecialchars(SID); ?>">cliquer ici</a>.
</p>
```

PHP avec MySQL

- MySQL est un système de gestion de base de données.
- MySQL est open source et il est souvent utilisé avec PHP pour stocker les données selon le modèle relationnel.
- Les données sont stockées dans des tables.
- Les colonnes dans les tables représentent les attributs des données et les lignes (tuples) les instances des données
- SQL (Structured Query Language) sert à effectuer des opérations sur les bases de données : créer, modifier, supprimer, rechercher, etc.

Connexion

- mysqli_connect(servername,username,password) pour se connecter à MySQL

  Exemple de connexion :

```
<?php
    $con = mysqli_connect("localhost","peter","abc123");
    if (mysqli_connect_errno()) {
       echo "Failed to connect: " . mysqli_connect_error();
       exit();
    }
?>
```

Manipuler une base de données (1/3)

- Avec la fonction mysqli_query() on peut envoyer une requête SQL à MySQL
- Créer une base de données :

```
mysqli_query($con,"CREATE DATABASE my_db")
```

- Créer une table dans la base de donnée my_db

```
mysqli_select_db($con,"my_db");
$sql = "CREATE TABLE Persons(
personID int NOT NULL AUTO_INCREMENT,
PRIMARY KEY(personID),
FirstName varchar(15),
LastName varchar(15),
Age int )";
mysqli_query($con,$sql);
```

Manipuler une base de données (2/3)

- Insérer des données à la table Persons:

```
mysqli_query($con,"INSERT INTO Persons (FirstName,
LastName, Age) VALUES ('Peter', 'Griffin', '35')");
```

- Sélectionner des données de la base de données :

```
$result = mysqli_query($con,"SELECT * FROM Persons
WHERE FirstName='Peter' ORDER BY age" );
while($row = mysqli_fetch_array($result)){
    echo $row['FirstName'];
    echo " " . $row['LastName'];
    echo " " . $row['Age'];
    echo "<br />";    }
```

Manipuler une base de données (3/3)

- Modifier un tuple :

```
mysqli_select_db($con,"my_db");
mysqli_query($con,"UPDATE Persons SET Age = '36'
WHERE FirstName = 'Peter' AND LastName = 'Griffin'");
```

- Supprimer un tuple :

```
mysqli_select_db($con,"my_db");
mysqli_query($con,"DELETE FROM Persons WHERE
LastName='Griffin'");
```

Exemple d'insertion de données provenant d'un formulaire (1/2)

Formulaire :

```
<html>
    <body>
        <form action="insert.php" method="post">
            Firstname: <input type="text" name="firstname" />
            Lastname: <input type="text" name="lastname" />
            Age: <input type="text" name="age" />
            <input type="submit" />
        </form>
    </body>
</html>
```

Exemple d'insertion de données provenant d'un formulaire (2/2)

Fichier PHP :

```
<?php
    $con = mysqli_connect("localhost","peter","abc123");
    if (!$con)
        die('Could not connect: ' . mysql_error());

    mysqli_select_db($con,"my_db");

    $sql="INSERT INTO Persons (FirstName, LastName, Age)
    VALUES ('$_POST[firstname]','$_POST[lastname]','$_POST[age]')";

    if (!mysqli_query($con,$sql))
        die('Error: ' . mysql_error());
    echo "1 record added";
    mysqli_close($con) ;
?>
```

Références

- https://www.php.net/manual/fr/