GJU

German Jordanian University الجامعة الألمانية الأردنية

CS355 Web Technologies

Dr. Ismail Hababeh

German-Jordanian University

Lecture 11

PHP Float/Double Data Type

- Floating point numbers in PHP can be larger than integers and they can have a decimal point.
- Floating point syntax formats:

```
 <?php
    $w = 3.245;
    $x = 3.2e3;
    $y = 3E-10;
    $z = 1264275425335735;
    var_dump($w); var_dump($x); var_dump($y); var_dump($z);
?>
```

PHP Float/Double Data Type

- The $z variable is assigned a large number, so it is automatically converted to float type.
- Output: float(3.245) float(3200) float(3.0E-10)

float(1264275425340000)

PHP Boolean Data Type

Boolean variables has two possible states: TRUE or FALSE.

Example:

```
<?php
$BL_val = False;
$x = rand(0, 1);
$BL_val = $x ? True: False;  // ternary operator: a shorthand way of
if ($BL_val) {               //writing an if-else statement
echo “Pass\n";
} else {
echo “Fail\n";
}        ?>
```

Notes on the Boolean Data Type Example

- The PHP script uses a random integer generator rand( ) function that returns a random number between the given integer boundaries.
- The ternary operator (allows to assign one value to the variable if the condition is true, and another value if the condition is false.)
- The Boolean variable is based on the random function evaluation.
- If the random function returns 1, then $BL_val variable is set to True; otherwise, If the random function returns 0, the $BL_val variable is set to False.

PHP String Data Type

- String is a data type representing textual data.
- Single quotes and double quotes can be used to hold string values.
- The concatenation operator (.) is used to set string values together

Example:

```
<?php
$course = "CS355";
$title = 'Web Technologies';
echo $course . $title;
?>
```

Output: CS355 Web Technologies

Search for a Specific Text in a String

- The PHP strpos() function searches for a specific text within a string.
- strpos() function returns the character position of the first match if a match is found, otherwise, the function returns FALSE.
- The position of the first character in the string is 0.

Syntax:

```
strpos(string, search text);
```

Search for a Specific Text in a String

Example:

```
<?php

echo strpos("CS355 Web Technologies", "Web");

?>
```

Output: 6

Replace a Specific Text in a String

- The str_replace() function replaces some characters with some other characters in a string.
- Syntax:

```
str_replace(find,replace,string,[count]) // counts the number of replacements
```

Example:

```
<?php
echo str_replace("Java", "Python", "Java Language");
?>
```

Output: Python Language

PHP String Length

- The strlen() function is used to find the length (i.e number of characters) of a string.
- It takes a string as a parameter and returns its integer length.

Example:

```
<?php
echo strlen("Hello World!");
?>
```

Output: 12

PHP Array Data Type

- The array() function is used for creating an array.
- PHP Array types:
  - Indexed arrays: Arrays with numeric index
  - Associative arrays: Arrays with named keys
  - Multidimensional arrays: Arrays consisting of one or more arrays

PHP Indexed & Associative Arrays

- Indexed arrays: Arrays with numeric index

  Syntax: array(value 1, value 2, value 3, … value n)

- Associative arrays: Arrays with named keys

  Syntax: array(key1=>value,key2=>value,key3=>value,...)

Using Loops on PHP Index Arrays – Example

```
<?php
$cars=array("Volvo","BMW","Toyota");
$arrlength=count($cars);
for($x=0;$x<$arrlength;$x++)
  {
  echo $cars[$x] ;
  echo "<br>";
  }
?>
```

PHP Associative Array - Example

```
<?php
$num=array("Web"=>"355","Java"=>"214","C++"=>"116");
echo "Java number is " . $num['Java'] ;
?>
```

Output: Java number is 214

Using Loops on PHP Associative Arrays

Example:

```
<?php
$grade=array("Zaid"=>"85", "Khaled"=>"97", "Laith"=>"73");
foreach($grade as $x=>$x_value)
  {
  echo "Key=" . $x . ", Value=" . $x_value;
  echo "<br>";
  }
?>
```

Using PHP Multidimensional Arrays

Creating a two-dimensional array example:

```
<?php
$cars=array
  (
  array("Ford",70,88),   // row 1 (col 1, col 2, col 3)
  array("Toyota",100,95),   // row 2 (col 1, col 2, col 3)
  array("Kia",40,50)   // row 3 (col 1, col 2, col 3)
  );
?>
```

Using Loops on PHP Multidimensional Arrays

```
<?php
for ($row = 0; $row < 3; $row++) {
  echo "<p><b>Row number $row</b></p>";
  echo "<ul>";
  for ($col = 0; $col < 3; $col++) {
    echo "<li>".$cars[$row][$col]."</li>";
  }
  echo "</ul>";
}?>
```

Example
HTML Form PHP 1/8

```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>

<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";
```

Example
HTML Form PHP 2/8

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  if (empty($_POST["name"])) {
    $nameErr = "Name is required";
  } else {
    $name = test_input($_POST["name"]);
    // check if name only contains letters and
whitespace
    if (!preg_match("/^[a-zA-Z-' ]*$/",$name)) {
      $nameErr = "Only letters and white space
allowed";
    }
  }
```

Example
HTML Form PHP 3/8

```
  if (empty($_POST["email"])) {
  $emailErr = "Email is required";
} else {
  $email = test_input($_POST["email"]);
  // check if e-mail address is well-formed
  if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $emailErr = "Invalid email format";
  }
}
```

Example
HTML Form PHP 4/8

```
  if (empty($_POST["website"])) {
    $website = "";
  } else {
    $website = test_input($_POST["website"]);
    // check if URL address syntax is valid (this
regular expression also allows dashes in the URL)
    if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-
a-z0-9+&@#\/%?=~_|!:,.;]*[-a-z0-
9+&@#\/%=~_|]/i",$website)) {
      $websiteErr = "Invalid URL";
    }
  }
```

Example
HTML Form PHP 5/8

```php
if (empty($_POST["comment"])) {
    $comment = "";
  } else {
    $comment = test_input($_POST["comment"]);
  }
  if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
  } else {
    $gender = test_input($_POST["gender"]);
  }
}
function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;}
?>
```

Example
HTML Form PHP 6/8

```
<h2>PHP Form Validation Example</h2>
<p><span class="error">* required field</span></p>
<form method="post" action="<?php echo htmlspecialchars
($_SERVER["PHP_SELF"]);?>">
  Name: <input type="text" name="name" value="<?php ech
o $name;?>">
  <span class="error">* <?php echo $nameErr;?></span>
  <br><br>
  E-
mail: <input type="text" name="email" value="<?php echo
 $email;?>">
  <span class="error">* <?php echo $emailErr;?></span>
  <br><br>
  Website: <input type="text" name="website" value="<?p
hp echo $website;?>">
  <span class="error"><?php echo $websiteErr;?></span>
  <br><br>
  Comment: <textarea name="comment" rows="5" cols="40
```

Example
HTML Form PHP 7/8

```
<?php echo $comment;?></textarea>
  <br><br>
  Gender:
  <input type="radio" name="gender" <?php if (isset($ge
nder) &&
$gender=="female") echo "checked";?> value="female">Fem
ale
  <input type="radio" name="gender" <?php if (isset($ge
nder) &&
$gender=="male") echo "checked";?> value="male">Male
  <input type="radio" name="gender" <?php if (isset($ge
nder) &&
$gender=="other") echo "checked";?> value="other">Other
<span class="error">* <?php echo $genderErr;?></span>
  <br><br>
  <input type="submit" name="submit" value="Submit">
</form>
```

Example
HTML Form PHP 8/8

```
<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
echo $email;
echo "<br>";
echo $website;
echo "<br>";
echo $comment;
echo "<br>";
echo $gender;
?>

</body>
</html>
```