Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

2) PHP syntax

Emmanuel Benoist

Spring Term 2017

Computer Science Division

Table of Contents

Presentation

PHP Hypertext Preprocessor (PHP)

- **A script language**
- **Inserted inside HTML documents**

```
<html>
<body>
<?php
    print("Hello World");
?>
<?
    print("Another hello");
?>
</body>
</html>
```

Comments

Comments

- **One line comments**

  // This line is commented out

- **Multi line comments**

  /∗ The content of the lines
  hereafter is commented out
  until the end of the comments ∗/

- **C shell like comments**

  # This line is also commented out

Functions

Define a function

- **Signature = number of arguments**
  - A function does not have any return type
  - Arguments do not have types
  - Arguments can have a default value (the function may be called with different number of arguments)

```
function add($x, $y){
    return $x+$y;
}
function mult($x,$y){
    return $x*$y;
}
print("5 + 4 =");
print(add(5,4));
print("\n");
print("5 * 4 =");
print(mult(5,4));
print("\n");
```

Variables and types

Variables I

- **Variables start with** `$`
  - Name should not start with a number
  - It can start with an `_`
- **You can use a variable without declaration**
  - A declaration is possible using the word `var`
- **You can use a variable without initialisation** (depends on the server's configuration)
  - The variable contains the default value which is evaluated as:
  - `0`, `""` (i.e. empty string), or `false`

Variables II

```
$x=1;
echo $x +$y ; // writes 1

if($z){
    echo "very unlikely";
}
else{
    echo "more probable";
}
```

PHP is loosly typed I

- **Variables do not have types**
  - The same variable can receive values from different types
  - $a can successively be an integer a string and a double:

```
$a="Hello";
echo $a;
$a = 1;
echo 2*$a;
$a = 1.1;
echo $a;
```

PHP is loosly typed II

- **Values still have a type**
  - Function `is_numeric()` tests if a variable contains a number

    ```
    $a=1;
    echo is_numeric($a); // prints true
    $a="Hello world";
    echo is_numeric($a); // prints false
    ```

  - Other functions
    `is_bool()`,`is_float()`,`is_string()`,`is_int()`, ...

Example

```php
<?php
  $a = "Hello ";
  echo $a; // writes: Hello

  $a = 1;
  $a++;
  echo $a; // writes: 2

?>
```

Strings

Strings

- **Initializied with ' '**
  - String is initialized with all the chars "as is"
- **Initializied with " "**
  - The content of the string is "interpoled"
  - Variables are replaced by their value
  - Escaped characters are also replaced: `\n`, `\r`, `\x87` (hexadecimal char), ...
  - One must escape special chars: `\$`, `\\`, `\"`
- **Here Doc**
  - Start with `<<<EOF` (where `EOF` is any string)
  - Ends with `EOF` (exactly the same sentence)
  - Content is interpoled

Example: Strings

```php
<?php
  $a = 1;
  $b = 2;
  $str1 = "Hello World";
  $str2 = "Hello $a";
  $str3 = 'Hello $b';

  $str4 = 'Hello \n';
  $str5 = "Hello \n";

  echo($str1); // writes: Hello World
  echo($str2); // writes: Hello 1
  echo($str3); // writes: Hello $b
  echo($str4); // writes: Hello \n
  echo($str5); // writes: Hello (and adds a linefeed)
?>
```

String : Here Doc example I

- **String is interpolated**

```
<?php
$a =1;
$str1 = <<<EOF
This string contains the value $a
EOF;

print $str1; // writes: This string contains the value 1
?>
```

String conversion

- **When a string is evaluated as a numeric value, the resulting value and type are determined as follows.**
  - The string will evaluate as a double if it contains any of the characters '.', 'e', or 'E'. Otherwise, it will evaluate as an integer.
  - The value is given by the initial portion of the string. If the string starts with valid numeric data, this will be the value used. Otherwise, the value will be 0 (zero).
    Valid numeric data is an optional sign, followed by one or more digits (optionally containing a decimal point), followed by an optional exponent.
    The exponent is an 'e' or 'E' followed by one or more digits.
  - When the first expression is a string, the type of the variable will depend on the second expression.

String conversion(Cont.)

```
$foo = 1 + "10.5"; // $foo is double (11.5)
$foo = 1 + "-1.3e3"; // $foo is double (-1299)
$foo = 1 + "bob-1.3e3"; // $foo is integer (1)
$foo = 1 + "bob3"; // $foo is integer (1)
$foo = 1 + "10 Small Pigs"; // $foo is integer (11)
$foo = 1 + "10 Little Piggies"; // $foo is integer (11)
$foo = "10.0 pigs " + 1; // $foo is integer (11)
$foo = "10.0 pigs " + 1.0; // $foo is double (11)
```

Arrays

Arrays

- **Arrays in PHP can be**
  - Numerical arrays : keys are numbers
  - Associative arrays
- **Numerical Arrays:**

```
$arrayA = array("zero","one","two");
echo $arrayA[0]; // output : zero
$arrayA[1]="eins";
$arrayA[3]="drei";
// an array can be used like a stack
$arrayB = array();
$arrayB[]="Hello";
$arrayB[]="World";
echo $arrayB[0]." ".$arrB[1]; // output: Hello World
```

Associative arrays I

- **The keys can also be any string**
  - We define a mapping key => value

```
$arrayC = array("one"=>"eins", "two"=>"zwei", "three"=>"drei");
echo $arrayC["one"]; // output: eins
$arrayC["one"]="un";
echo $arrayC["one"]; // output: un
```

The beauty of PHP arrays

- **In PHP one can mix all functionalities**
  - The same array may contain numerical and associative elements

```
$arrayD= array();
$arrayD[]='Hello'; // index implicit: 0
$arrayD[]="World"; // index implicit: 1
$arrayD[5]='Five';
$arrayD["X2a"]="Medizininformatik";
$arrayD["Hello"]="World";
$array["roesti"]=array("ingredient"=>"patatoes", price=>9.5);
echo $array["roesti"][price]; // Output : 9.5 (float)
```

Control structures

Control structures

- **Conditional branching**

```
if($x == 1){
    echo "x is one";
}
else{
    echo "x is not one";
}
```

- **Loops**

```
while($i<100){
  print $i.",";
  $i*=2;
}
for($j=0;$j<10;$j++){
  echo "$j, ";
}
```

Foreach

Foreach I

- **Visits all the items in an array**

  foreach(ARRAY as VALUE){
  }

```
$arrayE = array(10,20,30,40);
$sum=0;
foreach($arrayE as $val){
   $sum += $val;
}
echo $sum; // Output: 100
```

Foreach II

- **Can also access to the key of the elements**

```
foreach( ARRAY as KEY => VALUE){
}
```

```
$arrayF=array( "orange"=>10.5,
               "apple" => 5.9,
               "grappefruit"=>4.9);
foreach($arrayF as $fruit => $price){
  echo "Price of $fruit is $price\n";
}
```

isset()

The boolean function `isset()`

- `isset()` **tests if a value has been initialized before**

```
$x = 1;
if(isset($x)){
  echo "x has been initialized";
}
// value $y has not been set
if(isset($y)){
  echo "y has been initialized";
}
// output: x has been initialized
```

Conclusion

Conclusion

- **PHP is similar to Java**
  - Basic syntax (if, while, for, . . .)
- **It is also very different from Java**
  - Variables are loosly typed
  - Arrays are structures that are polymorph
  - Transform an element from one type into another one is easy