Dept of Computer Science
Ahmadu Bello University

WEB APPLICATION ENGINEERING II

Lecture #3

Umar Ibrahim Enesi

Dept of Computer Science
Ahmadu Bello University

Objectives

- Gain understanding on the various control structures available in PHP.
- Learn how to create user defined functions.
- Learn how to locate documentations for and use various built-in functions.

Dept of Computer Science
Ahmadu Bello University

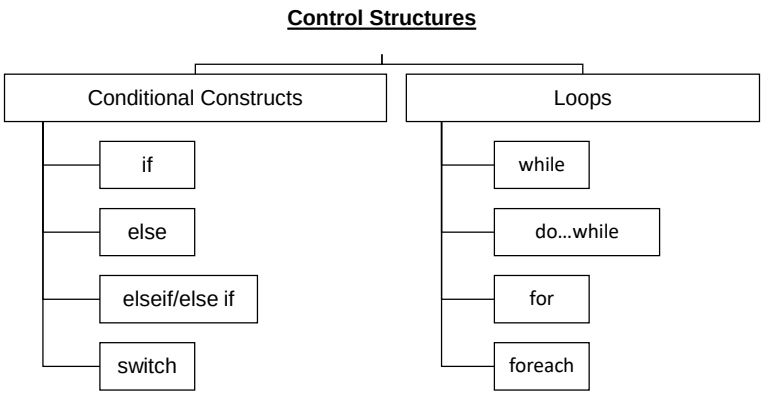
Control Structures: Introduction

- Normally, statements in a script file are executed from top to bottom.
- Sometimes it makes sense logically to alter the normal flow of execution in other to achieve a desired result.
- Control structures enable us to do that.
- Control structures can be loop or conditional construct.

Dept of Computer Science
Ahmadu Bello University

Control Structures: Introduction

- Many of PHP's control structures are controlled by conditional expressions.



Control Structures: keywords

- Some PHP keywords are mostly used in conjunction with control structures

break	continue
as	default
goto	case
return	

Control Structures: if, if…else

Syntax:

```
if(cond_expr){
     //statement(s) if cond_expr evaluates to true
}

if(cond_expr){
     //statement(s) if cond_expr evaluates to true
}else{
     //statement(s) if cond_expr evaluates to false
}
```

Dept of Computer Science
Ahmadu Bello University

Control Structures: elseif, else if

Syntax:

```
if(cond_expr_1){
      //statement(s) if cond_expr_1 evaluates to true
}elseif (cond_expr_2) {
      //statement(s) if cond_expr_2 evaluates to true
}

if(cond_expr_1){
      //statement(s) if cond_expr_1 evaluates to true
}else if(cond_expr_2){
      //statement(s) if cond_expr_2 evaluates to true
}
```


Dept of Computer Science
Ahmadu Bello University

Control Structures: switch

Syntax:

```
switch(expr){
      case v1:
            statement(s);    break;
      case v2:
            statement(s);    break;
      ...
      case vn:
            statement(s);    break;
      default:
            statement(s);
}
```

Dept of Computer Science
Ahmadu Bello University

Control Structures: while

Syntax:

```
while(cond_expr){
    //statement(s) if cond_expr evaluates to true
}
```


Dept of Computer Science
Ahmadu Bello University

Control Structures: do…while

Syntax:

```
do{
    statement(s);
}while(cont_expr);
```

Dept of Computer Science
Ahmadu Bello University

Control Structures: for loop

Syntax:

```
for(initialisation; cond_expr; increment){
    statement(s);
}
```

- *initialisation*, *cond_expr* and *increment* are optional

What does the above implies?

Dept of Computer Science
Ahmadu Bello University

Control Structures: foreach

Syntaxes:

```
foreach (array_expr as $value) {
    statement(s);
}

 foreach (array_expr as $key => $value) {
    statement(s);
}
```

- This loop construct is specially suited for arrays and traversable objects (see PHP's SPL)

Control Structures: break, continue, return

- `break` ends execution of the current *loop or switch structure.*
- `continue` ends execution of the current iteration of a loop.
- `return` statement returns program control to the calling module.
- `return` automatically ends execution of a function, loop, included file or script.

File Inclusion: include

- `include` statement includes the contents of a specified file into the calling script.
- code in the included file behave as though it was originally defined in the calling script.
- Thus inherits the variable scope at the point of inclusion.
- Path given to include can be absolute or relative path.

```
<?php
     include "myfile.php";
?>
```

- PHP issues warning if the file to be included cannot be found.

Dept of Computer Science
Ahmadu Bello University

File Inclusion: include_once

- `include` statement allows files to be included more than once.
- `include_once` ensures that a file is included only once.

```
<?php
     include_once "myfile.php";
?>
```

- PHP issues warning if the file to be included cannot be found.

Dept of Computer Science
Ahmadu Bello University

File Inclusion: require(_once)

- `require` works similar to `include` statement except that an error is issued when a file cannot be found.
- `require_once` works similar to `require` but ensures that files are included only once.

Dept of Computer Science
Ahmadu Bello University

Functions: Introduction

- To facilitate code reuse and organisation, statements are grouped (code block) under a named construct called *function*.
- The code block is executed whenever the function is called.
- Functions always return values and may receive input (arguments).

Dept of Computer Science
Ahmadu Bello University

Functions: Syntax

Simplistically,

```
function func_name(optional_args){
      //statement(s);
      //optional return statement
}
```

- `func_name` follows the same naming rules for variables.
- Function names are case insensitive.

Functions: argument

- A function may be passed some information (inputs).
- Such inputs are called arguments.
- Multiple arguments in a function declaration are separated by comma (,)

```
<?php
    /*function declaration */
    function product($num1, $num2){
        $prod = $num1 * $num2;
        return $prod;
    }

    $x = product(100, 4); //function call
    echo $x; //outputs 400
?>
```


Functions: argument value vs reference

- By default function arguments are passed by value.
- So a copy of the values of function arguments are used within the function.
- To pass the reference of variables to function arguments, prepend ampersand (&) to the argument name.

```
<?php
    function allCaps(&$word){
        $word = strtoupper($word);
    }
    $course = "Web Application Engineering II";
    allCaps($course);//function call
    echo $course; //WEB APPLICATION ENGINEERING II
?>
```

Dept of Computer Science
Ahmadu Bello University

Functions: default argument value

- Default values of function arguments are specified by assigning a value to the argument.
- Default values must be literal values – variables or return value of function are not allowed
- Default arguments should always be place at the right side of the function argument list.
-
```
<?php
    function greet($name= "Anonymous"){
        echo "Hello, $name!";
    }
    greet("Ibrahim");//Outputs Hello, Ibrahim!
    greet();//Outputs Hello, Anonymous!
?>
```


Dept of Computer Science
Ahmadu Bello University

Functions: variable argument list

- Sometimes a function might be declared to accept variable number of arguments.
- Use "…" token to denote variable-length argument in a function declaration.
- Arguments will be received as an array
-
```
<?php
    function sum(...$numbers){
        $sum= 0;
        foreach($numbers as $number){ $sum+=$number; }
        return $sum;
    }
    $total1 = sum(1, 5, 2, 5, 23, 62);
    $total2 = sum(54,13, 4);
?>
```

Dept of Computer Science
Ahmadu Bello University

Functions: Local scope

- Functions introduce local scope within its declaration.
- Within a function declaration, the global scope is not accessible.
- To access the global scope and reach out for global variables and functions, use:
  - `global` keyword
  - `$GLOBALS[]` superglobal array

Dept of Computer Science
Ahmadu Bello University

Functions: Return values

- Functions may return values before passing control back to its calling module.
- Return values can be of any type.
- If `return` statement is omitted, NULL is returned.

```
<?php
     function product($num1, $num2){
          return ($num1 * $num2);
     }
?>
```

Dept of Computer Science
Ahmadu Bello University

Functions: Built-in Functions

- PHP comes with many built-in functions.
- Some functions require certain PHP extensions.
- The PHP documentation manual describes all of these functions.

Dept of Computer Science
Ahmadu Bello University

Key Points

- Control statements may be loop or conditional structures.
- Most control statements are controlled by conditional expressions.
- `foreach` loop is specially suited for arrays and traversable objects.
- Some PHP keywords are mostly used in conjunction with control structures.
- `while`, `do…while` and `for` loops performs similar function but in different fashion.
- Functions facilitate code reuse and organisation and may be called multiple times.
- Functions may receive variable argument list.
- Functions always return values.
- Functions introduces a local scope which overshadows the global scope within its declaration.
- Within a function global scope can be accessed with `global` or `$GLOBALS[]` super global array
- PHP comes with a lot of built-in functions, some of which are provided by PHP extensions.

References

- PHP Documentation Manual
- Murach PHP and MySQL
- http://www.w3schools.com
- http://www.php5-tutorial.com
- http://www.tutorialspoint.com