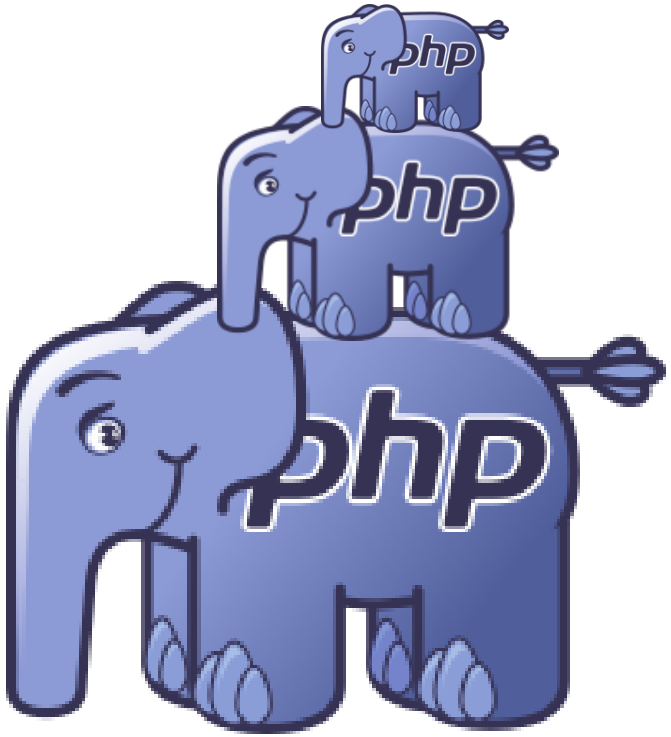
PHP Basics



INCLUDING CODE

FUNCTIONS

EASTERN WASHINGTON UNIVERSITY

Including code

Load code and HTML from another module/file

- First look in `include_path` in `php.ini` file
- Finally looks in same directory as script

`include` – issues warning if file not found

`require` – fatal error if file not found

include

Common use is to bring in page-specific content from site design files, i.e., headers and footers

```
<?php include “header.html” ?>

  page contents

<?php include “footer.html” ?>
```

require

Best used for code inclusion

```
<?php
    require “my_code_library.php”;
   . . .
    my_sub(); //defined in my_code_library.php
    . . .
?>
```

Load once

Variations to quietly ignore reloads of same files

```
include_once
require_once
```

Embedding Code in HTML

XML-style

- Use <?php and ?> tags to embed directly in HTML document

```
<body>
  <p><?php echo “Hello, World!”; ?> <br />
   </p>
</body>
```

Embedding Code in HTML

Server will run PHP code when serving the page

- Substitute the results directly in HTML
- No trace of PHP in HTML response

Functions

(Obvious) Details

Block of code performing a specific task of computation

Send values from outside function through parameters

May return single value or multiple values through array

Call function by giving name and required parameters

Calling Examples

```
$length = strlen(“My String”);

$result = sin(asin(1));

$AtoZ = range(“A”, “Z”);
```

PHP has a huge set of functions already defined

https://www.php.net/manual/en/indexes.functions.php

Defining a Function

```
function functionname( <parameter list> )
{
    statements
}
```

Function names start with letter or _ (underscore), then zero or more { _, letters, digits }

Defining a Function

You can declare a function that doesn't include any PHP code

```
<?php function column( )
{ ?>
    </td><td>
<?php } . . .
```

Defines a short name to the contained HTML code (within PHP) used many times throughout the page

Defining a Function

Use `return` statement to return value from function

```
function strcat($s1, $s2)
{
    $combined_string = $s1 . $s2;
    return $combined_string;
}
```

Defining a Function

Once a function is defined, it can be used from anywhere in the page

```
<?php
  function strcat($s1, $s2)
  {
      return $s1 . $s2;
  }

  $first = “This is a ”;
  $second = “complete sentence.”;
  echo strcat($first, $second);
```

Return Value

Single return value allowed

Use array to return multiple values

```
function returnArray( )
{
    return array(“Fred”, 35);
}
```

Variable Scope

Local – within function where defined/used

```
$a = 3;
function iPityTheFoo( )
{
    $a += 2;
}

iPityTheFoo();
echo $a;
```

Variable Scope

Global – use `global` keyword to access external variable

```
$a = 3;
function foo( )
{
    global $a;
    $a += 2;
}

foo();
echo $a;
```

Variable Scope

Static – retains value in function between calls

```
$a = 3;
function foo( )
{
    static $a = 0;
    return $a += 2;
}

for ($i = 0; $i < 10; ++$i)
    echo “foo()\n”;
```

Parameters

Default is *Pass-By-Value*

- Parameter evaluated, then copy of value sent to appropriate variable in parameter list of function

Pass-By-Reference

- Add & to formal parameter in list
  - Sends reference to function
  - Can change/update actual parameter

CANNOT send literals to reference parameters

Pass-By-Reference Example

```
function doubler( &$value )
{
    $value = $value << 1;
}

$a = 3;
doubler($a);
echo $a;
```

Can save time if sending large strings, arrays, objects

- No copy needed (even if not changing value)

Default Parameters

Assign value in formal parameter list

MUST follow all non-defaulted parameters

```
function bar($n, $m = “none”)
{
    // function code
}

bar(12, “two”);
bar(7);  // uses default for $m
```

Variable Parameters

Leave formal list of parameters blank

Use PHP functions to examine/retrieve arguments sent

`func_get_args()` – array of all parameters provided

`func_num_args()` – the number of parameters used

`func_get_arg(<number>)` – access specific argument

Variable Parameters Example

```
function sumList( )
{
    if (func_num_args() == 0)
        return false;
    else {
        $sum = 0;
        for ($i = 0; $i < func_num_args(); ++$i) {
            $sum += func_get_arg($i);
        }
        return $sum;
    }
}

echo sumList(1, 5, 9);
```

Miscellaneous

Variable Functions

- Call function whose name is stored in variable
  - Add `()` or parameter list
- Runtime error if no function exists (`function_exists()`)

Anonymous Functions

- Use normal syntax, but use as parameter or assign to variable
  - Example: `usort()`