Introduction to Server-Side Development with PHP

Chapter 11

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development

http://www.funwebdev.com

Chapter 11

1 What Is Server-Side Development?

2 Quick Tour of PHP

3 Program Control

4 Functions

5 Summary

Chapter 11

1 What Is Server-Side Development?

2 Quick Tour of PHP

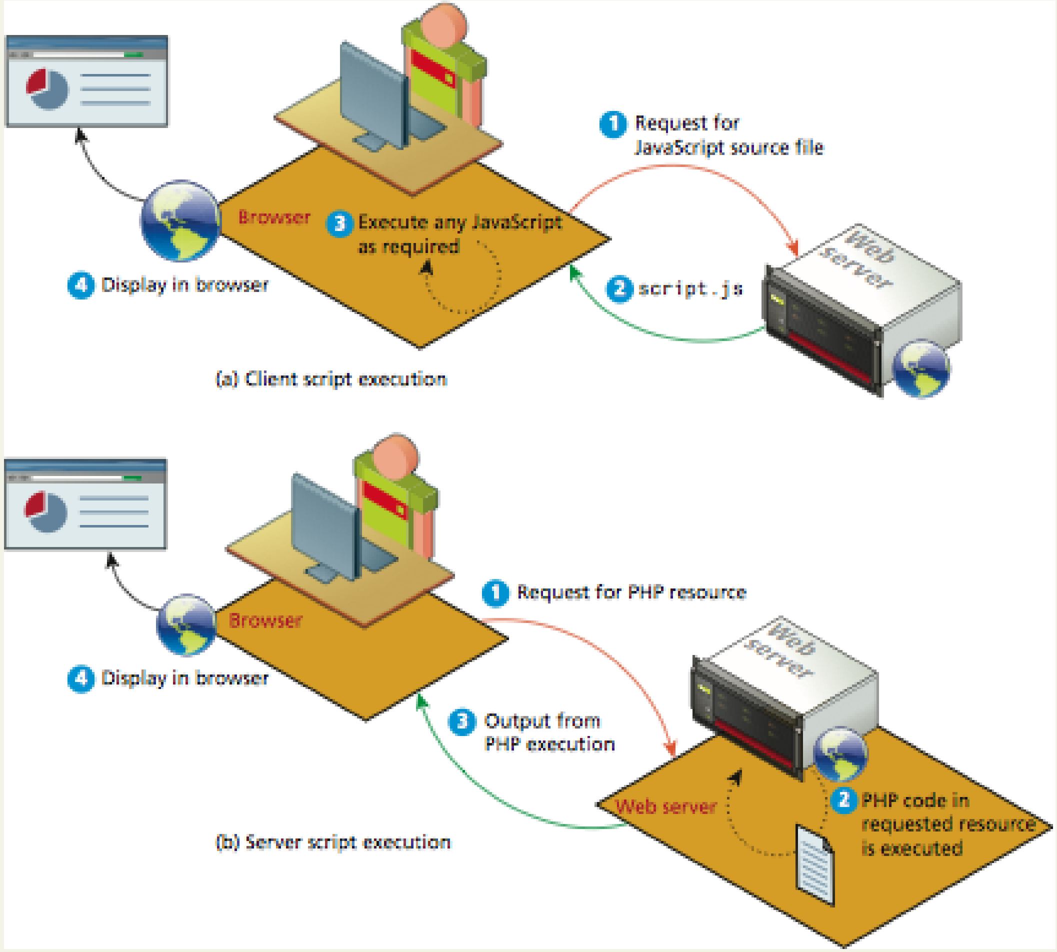
3 Program Control

4 Functions

5 Summary

What Is Server-Side Development?

Comparing Client and Server Scripts



1 Request for JavaScript source file
Browser
3 Execute any JavaScript as required
2 script.js
4 Display in browser
Web server
(a) Client script execution
1 Request for PHP resource
Browser
4 Display in browser
3 Output from PHP execution
Web server
2 PHP code in requested resource is executed
(b) Server script execution

What Is Server-Side Development?

Server-Side Script Resources

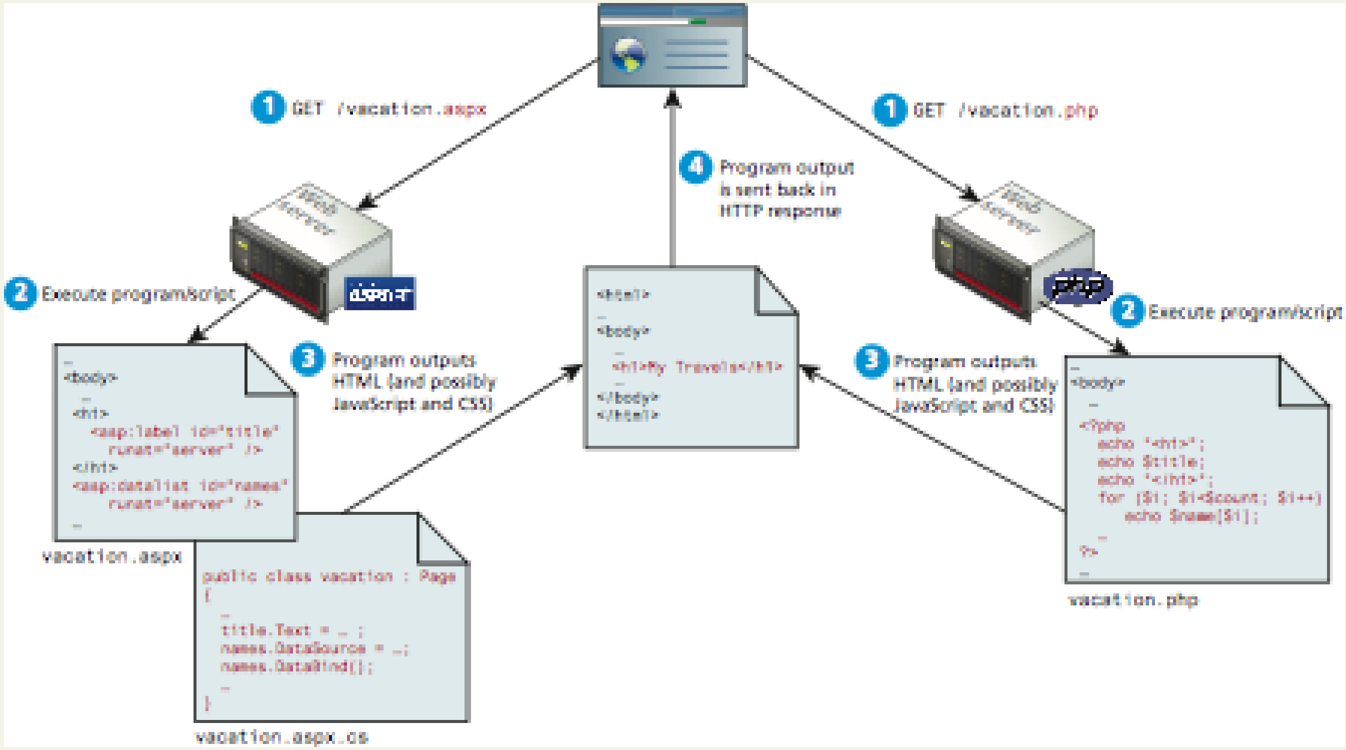
What Is Server-Side Development?

Comparing Server-Side Technologies

- ASP (Active Server Pages) / ASP.NET
- JSP (Java Server Pages)
- Node.js
- Perl
- PHP
- Python
- Ruby on Rails

What Is Server-Side Development?

Comparing Server-Side Technologies


1 GET /vacation.aspx
Web Server
2 Execute program/script
3 Program outputs HTML (and possibly JavaScript and CSS)
4 Program output is sent back in HTTP response
1 GET /vacation.php
Web Server
php
2 Execute program/script
3 Program outputs HTML (and possibly JavaScript and CSS)
vacation.aspx
vacation.aspx.cs
vacation.php

```
…
<body>
  …
 <h1>
   <asp:label id="title"
     runat="server" />
 </h1>
 <asp:datalist id="names"
     runat="server" />
 …
```

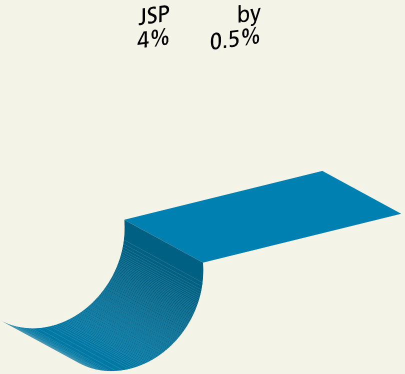
```
public class vacation : Page
{
  …
  title.Text = … ;
  names.DataSource = …;
  names.DataBind();
  …
}
```

```
<html>
…
<body>
  …
  <h1>My Travels</h1>
  …
</body>
</html>
```

```
…
<body>
  …
<?php
  echo '<h1>';
  echo $title;
  echo '</h1>';
  for ($i; $i<$count; $i++)
     echo $name[$i];
  …
?>
…
```

What Is Server-Side Development?

Comparing Server-Side Technologies



Top 10,000 Sites

Chapter 11

1 What Is Server-Side Development?

2 Quick Tour of PHP

3 Program Control

4 Functions

5 Summary

Quick Tour of PHP

PHP Tags

<?php tag and a matching closing **?>**

Inside is code to execute, outside is HTML to echo directly

Quick Tour of PHP

PHP Comments

- **Single-line comments.** Lines that begin with a #
- **Multiline (block) comments.** begin with a /* and encompass everything that is encountered until a closing */
- **End-of-line comments.** // to end of line

Quick Tour of PHP

Variables, Data Types, and Constants

- Variables in PHP are dynamically typed
- To declare a variable you must preface the variable name with the dollar ($) symbol.
- Whenever you use that variable, you must also include the $ symbol with it.
- Right to left assignment

```
$count = 42;
```

Quick Tour of PHP

Variables, **Data Types**, and Constants

- Boolean A logical true or false value
- Integer Whole numbers
- Float Decimal numbers
- String Letters
- Array A collection of data of any type (covered in the next chapter)
- Object Instances of classes

Quick Tour of PHP

Variables, **Data Types**, and Constants

Escaping Strings

- \n Line feed
- \t Horizontal tab
- \\ Backslash
- \$ Dollar sign
- \” Double quote

Quick Tour of PHP

Variables, Data Types, and **Constants**

- Use define()
- uppercase for constants is a programming convention
- Then use the word without quotes (or $)

define("DATABASE_LOCAL", "localhost");

echo DATABASE_LOCAL;

Quick Tour of PHP

Writing to Output

echo()

echo ("hello");

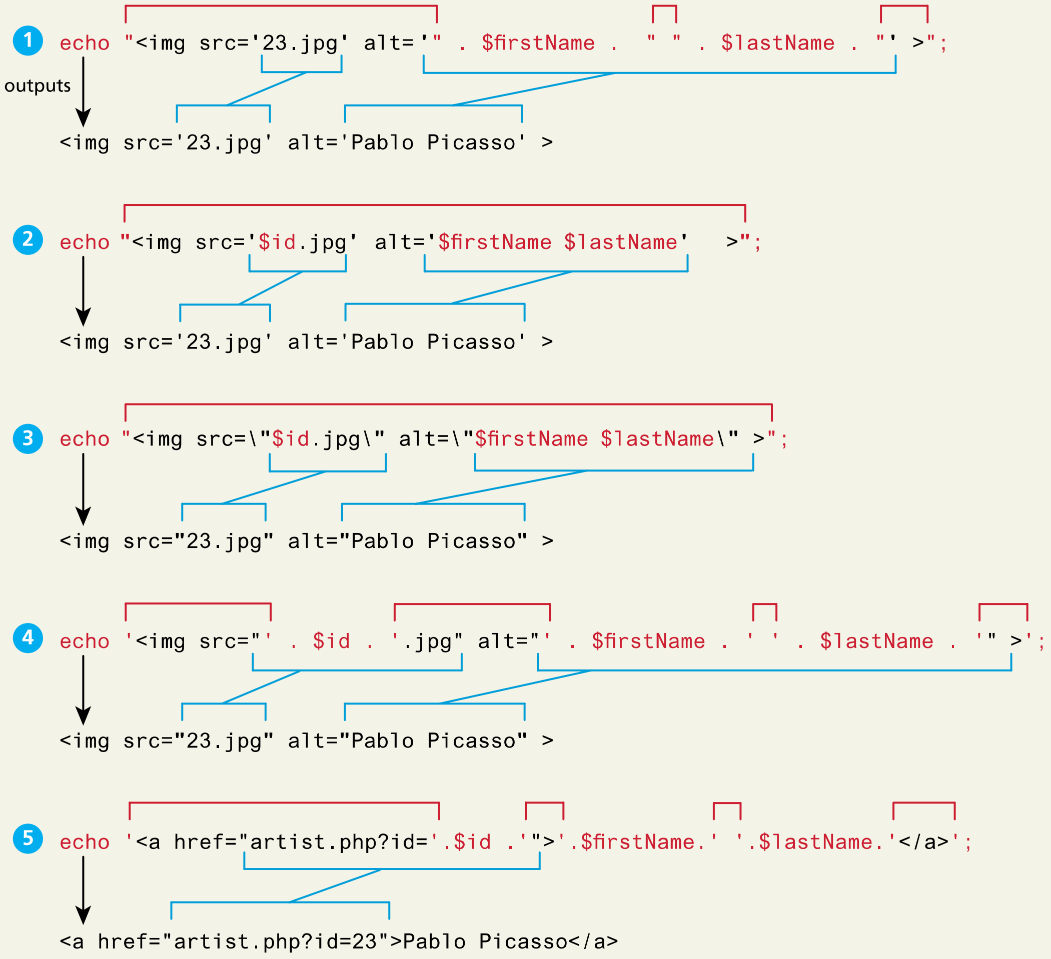
Strings can easily be appended together using the concatenate operator, which is the period (.) symbol.

$username = "Ricardo";

echo "Hello". $username; //outputs *Hello Ricardo*

Quick Tour of PHP

More concatenation examples



1

```
echo "<img src='23.jpg' alt='" . $firstName . " " . $lastName . "' >";
```

outputs

```
<img src='23.jpg' alt='Pablo Picasso' >
```

2

```
echo "<img src='$id.jpg' alt='$firstName $lastName'   >";
```

```
<img src='23.jpg' alt='Pablo Picasso' >
```

3

```
echo "<img src=\"$id.jpg\" alt=\"$firstName $lastName\" >";
```

```
<img src="23.jpg" alt="Pablo Picasso" >
```

4

```
echo '<img src="' . $id . '.jpg" alt="' . $firstName . ' ' . $lastName . '" >';
```

```
<img src="23.jpg" alt="Pablo Picasso" >
```

5

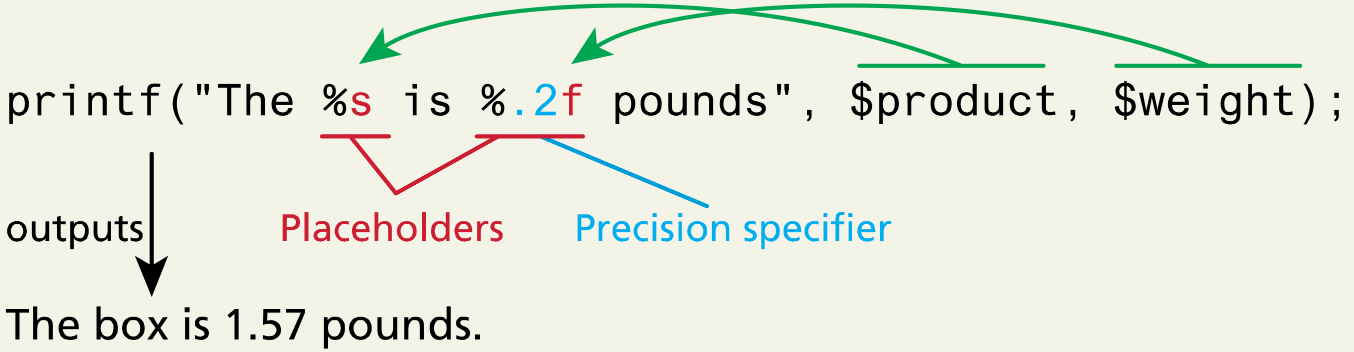
```
echo '<a href="artist.php?id=' .$id .'">'.$firstName.' '.$lastName.'</a>';
```

```
<a href="artist.php?id=23">Pablo Picasso</a>
```

Quick Tour of PHP

printf

```
$product = "box";
$weight = 1.56789;
```



```
printf("The %s is %.2f pounds", $product, $weight);
```

outputs

Placeholders

Precision specifier

The box is 1.57 pounds.

Chapter 11

1 What Is Server-Side Development?

2 Quick Tour of PHP

3 Program Control

4 Functions

5 Summary

Program Control

if …else

```
// if statement
if ( $hourOfDay > 6 && $hourOfDay < 12) {
        $greeting = "Good Morning";
}
else if ($hourOfDay == 12) { // optional else if
        $greeting = "Good Noon Time";
}
else { // optional else branch
        $greeting = "Good Afternoon or Evening";
}
```

Program Control

switch . . . case

```
switch ($artType) {
        case "PT":
                $output = "Painting";
                break;
        case "SC":
                $output = "Sculpture";
                break;
        default:
        $output = "Other";
}
```

Program Control

while and do . . . while

```
$count = 0;
while ($count < 10){
        echo $count;
        $count++;
}
```

Program Control

while and **do . . . while**

```
$count = 0;
do {
        echo $count;
        // this one increments the count by 2 each
time
        $count = $count + 2;
} while ($count < 10);
```

Program Control

for

```
for ($count=0; $count < 100; $count+=5)
{
        echo $count;
}
```

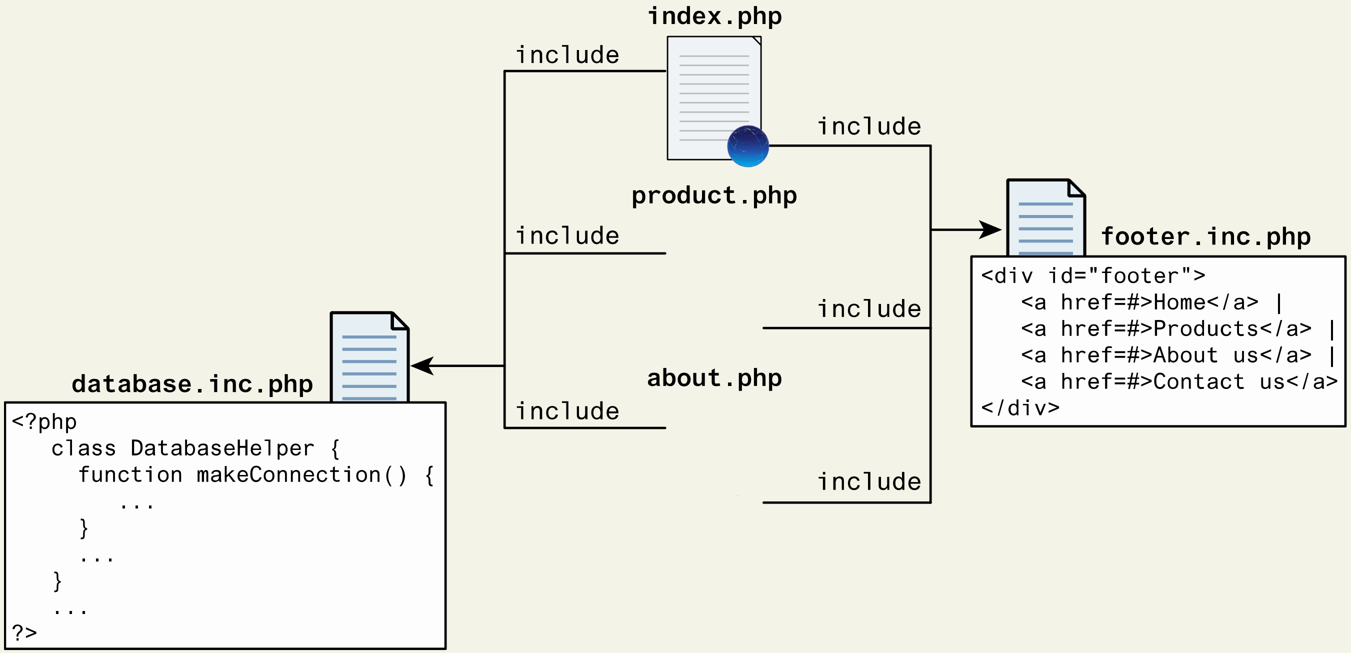
Program Control

Alternate Syntax for Control Structure

```
<?php if ($userStatus == "loggedin") : ?>
        <a href="account.php">Account</a>
        <a href="logout.php">Logout</a>
<?php else : ?>
        <a href="login.php">Login</a>
        <a href="register.php">Register</a>
<?php endif; ?>
```

Program Control

Include Files



```
<?php
   class DatabaseHelper {
     function makeConnection() {
        ...
     }
     ...
   }
   ...
?>
```

```
<div id="footer">
   <a href=#>Home</a> |
   <a href=#>Products</a> |
   <a href=#>About us</a> |
   <a href=#>Contact us</a>
</div>
```

Program Control

Include Files

```
include "somefile.php";

include_once "somefile.php";
```

Chapter 11

1 What Is Server-Side Development?

2 Quick Tour of PHP

3 Program Control

4 Functions

5 Summary

Functions

Function Syntax

```
/**
* This function returns a nicely formatted string using the
* system time.
*/
function getNiceTime(){
        return date("H:i:s");
}
```

Functions

Function Syntax – return type declaration

A Return Type Declaration explicitly defines a function’s return type by adding a colon and the return type after the parameter list when defining a function

```
function mustReturnString() : string {

		return "hello";

}
```

Functions

Calling a Function

Functions

Parameters

Parameters are the mechanism by which values are passed into functions

- To define a function with parameters, you must decide
  - how many parameters you want to pass in, and
  - in what order they will be passed.

Functions

Parameters

```
function getNiceTime($showSeconds) {
        if ($showSeconds==true)
                return date("H:i:s");
        else
                return date("H:i");
}

echo getNiceTime(true); // this will print seconds
echo getNiceTime(false); // will not print seconds.
```

Functions

Parameters – default values

In PHP you can set parameter **default values** for any parameter in a function. However, once you start having default values, all subsequent parameters must also have defaults.

```
function getNiceTime($showSeconds=true) {
        if ($showSeconds==true)
                return date("H:i:s");
        else
                return date("H:i");
}
```

Functions

Parameters – Passing Parameters by reference

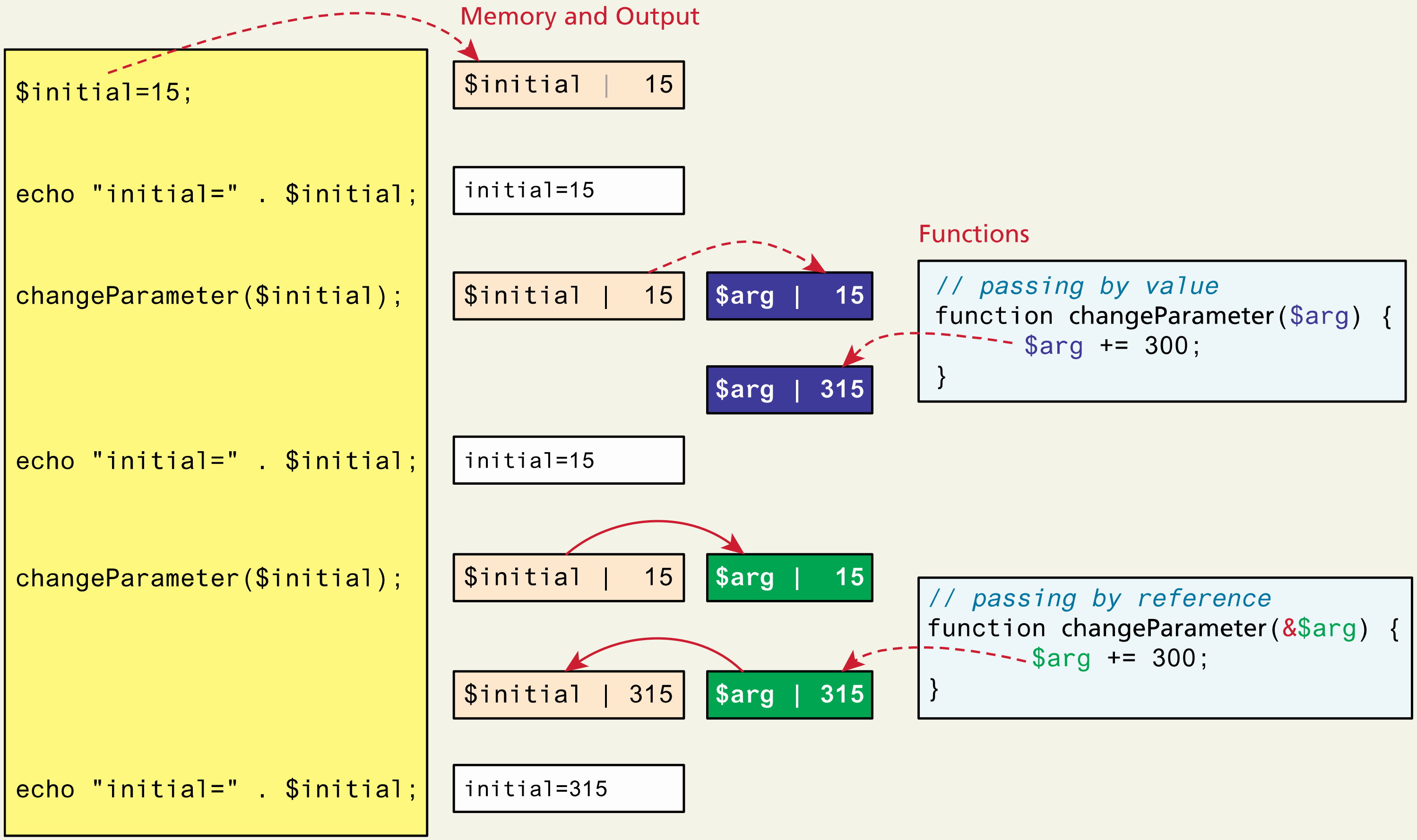
By default, arguments passed to functions are **passed by value** in PHP.

PHP also allows arguments to functions to be passed by reference , which will allow a function to change the contents of a passed variable

The mechanism in PHP to specify that a parameter is passed by reference is to add an ampersand (&) symbol next to the parameter name in the function declaration

Functions

Parameters – Passing Parameters by reference



```
$initial=15;

echo "initial=" . $initial;

changeParameter($initial);


echo "initial=" . $initial;

changeParameter($initial);


echo "initial=" . $initial;
```

Memory and Output

Functions

```
// passing by value
function changeParameter($arg) {
    $arg += 300;
}
```

```
// passing by reference
function changeParameter(&$arg) {
    $arg += 300;
}
```

Functions

Variable Scope within Functions

All variables defined within a function have function scope , meaning that they are only accessible within the function.

While variables defined in the main script are said to have global scope , these global variables are not by default, available within functions.

PHP does allow variables with global scope to be accessed within a function using the **global** keyword,

Chapter 11

1 What Is Server-Side Development?

2 Quick Tour of PHP

3 Program Control

4 Functions

5 Summary

Summary

Key Terms

ASP /ASP.NET
built-in function
Common Gateway Interface (CGI)
constant
daemon
data storage
data types
database
database management system (DBMS)
dynamically typed
extension layer
fork
function
function scope
global scope
handlers
Java Server Pages (JSP)
loosely typed
module
multi-process
multi-threaded
opcodes
overloading
parameters
parameter default values
passed by reference
passed by value
Perl
PHP
PHP core
preforked
process
Python
Return-type declarations
Ruby On Rails
SAPI
server-side includes (SSI)
thread
user-defined function
virtual machine
web services
worker
Zend Engine

Summary

Questions?