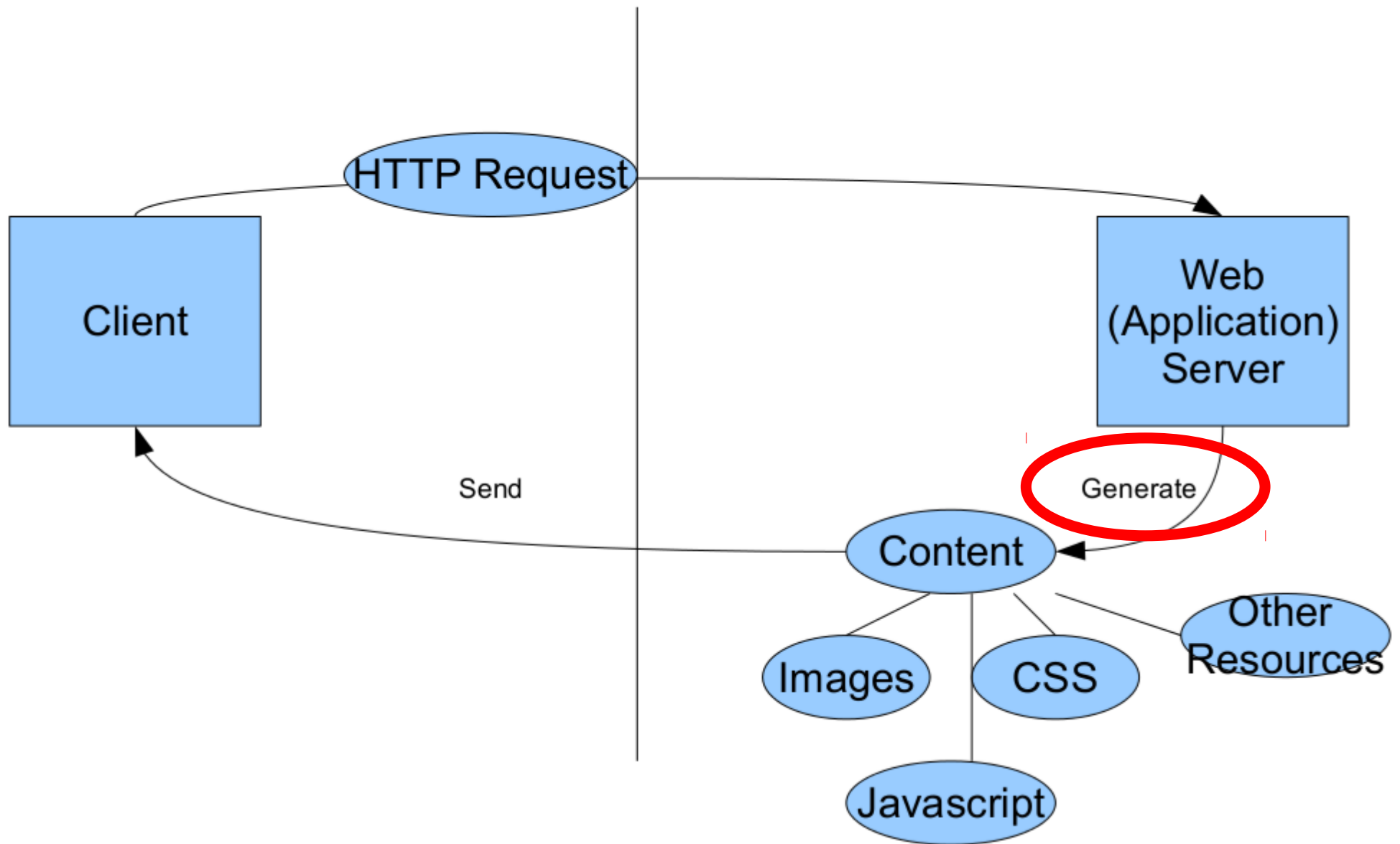


CMSC 100 Web Programming

PHP



Goals/purposes of the Server-side Script

- To generate content according to the specification indicated by a request.
- To generate content depending on the user.

PHP

- General purpose scripting language
 - Designed for web development and generally runs on a web server.
- Interpreted language
- Open source and free software
- Used in many websites and other open source projects.

PHP Installation

- **For Windows:**

- Easy to Install bundles: WAMPServer, XAMP, etc.
- Get Installers for Apache WS, PHP, MySQL individually.

- **For Linux:**

- Linux-Apache-MySQL-PHP (LAMP)
 - Standard installation (i.e. apt-get/Package Manager)

Syntax

- Scripting Blocks – PHP code embedded/mixed into the content.

<?php

echo “This is PHP Code”;

?>

Syntax: Variables

- PHP is a loosely typed language.
 - Data types are not declared with the variable
 - Variables need not to be declared before adding a value to it.
 - The type of the value can change during runtime.

Syntax: Variables

- Variables start with \$.
- Must start with a letter or underscore.
- May contain alphanumeric and underscore characters after the first.
- No spaces.

Syntax: Variables

- `$_myVar = 10;`
- `$_____ = "also a variable";`
- `$_myVar = "Now a string!";`
- `$index7 = 'Also a string.';`

Syntax: Comments

- Similar to C/Java comments

<?php

//Single line comment

/*

Multi-line comment

***/**

?>

Syntax: Functions

- Similar to Javascript

```
function functionName( ) {  
    //code to be executed  
}
```

Syntax: Functions

- Functions with parameters

```
<?php
```

```
function printConcat($x, $y ) {  
    echo $x.$y;  
}
```

```
//calling
```

```
printConcat("Hello ", "there" )
```

```
?>
```

Syntax: Functions

- Functions with parameters

```
<?php
```

```
function printConcat($x, $y ) {  
    echo $x.$y;  
}
```

```
//calling
```

```
printConcat("Hello ", "there" );
```

```
?>
```

Syntax: Global Variables

- Generally \$vars you have outside functions are “global” variables.

```
<?php
```

```
$g = 10;
```

```
function useG(){
```

```
    echo "g = ".$g;
```

```
}
```

```
?>
```

Syntax: Classes

- Object Oriented Programming via classes

<?php

class SimpleClass {

//property (attribute) declaration

public \$x = 10;

//method

public function getVar() {

return \$this->x;

}

**}
?>**

Syntax: Classes/Objects

- **\$obj = new SimpleClass();**
- **\$another = \$obj; // cloned**
- **\$ref =& \$obj; //reference**

PHP Data Types

- **Scalar** types
 - boolean
 - Integer
 - float
 - string
- **Compound** types
 - array
 - object

PHP Data Types

- `$boolean_val = true; //or TRUE or True`
- `$string_val = "This is a string.";`
- `$string_val = 'Another string.';`
- `$anumber = 12; //integer, may use octal or hex`
- `$hexnum = 0XABC;`
- `$octal = 012;`
- `$float = 12.34;`

Strings

- Maybe enclosed in “ “ or ' ' (single quotes).
- Note: Variables are evaluated inside “ “ .

```
<?php
```

```
$myStr = 'this is it.';
```

```
echo “Well, $myStr”;
```

```
?>
```

PHP Arrays

```
<?php
//Numeric index
$sections = array("UV-1L", "UV-2L", "UV-
3L");
echo $sections[0]; //0 based automatic

$subjs = array(); //manual assignment
$subjs[0] = 'CMSC 100';
?>
```

PHP Arrays: Associative Arrays

```
<?php
```

```
//Associative
```

```
$difficulty = array( "CMSC100"=>"hard",  
                    "CMSC22"=>"hard");
```

```
echo $difficulty['CMSC100'];
```

```
$difficulty['MATH17'] = "easy";
```

```
?>
```

PHP Multidimensional Arrays

```
<?
$multi = array (
    "Cartoon"=> array ("Ben", "Bubbles"),
    "Action"=> array("Arnold", "Bruce")
);
echo $multi['Cartoon'][0];
?>
```

Reading Assignment

- Predefined Variables in PHP
 - \$_GET
 - \$_POST
 - \$_REQUEST
 - ...