

Basic PHP

Lecture 19

Robb T. Koether

Hampden-Sydney College

Mon, Feb 26, 2108

- 1 PHP
- 2 The `echo` Statement
- 3 Variables
- 4 Operators
- 5 Decision Structures

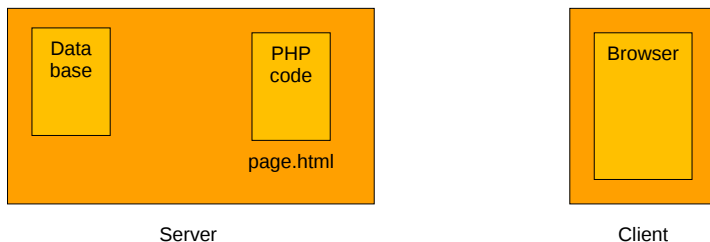
Outline

- 1 PHP
- 2 The `echo` Statement
- 3 Variables
- 4 Operators
- 5 Decision Structures

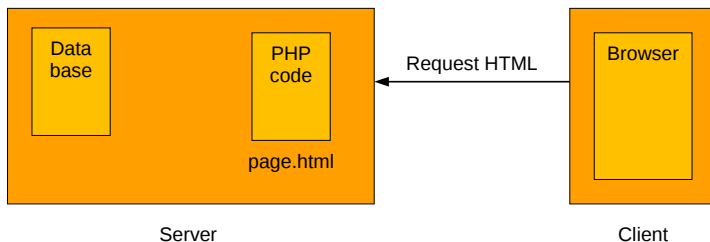
- PHP = “PHP Hypertext Preprocessor.”

- PHP = “PHP Hypertext Preprocessor.”
- So, what does the first P stand for?

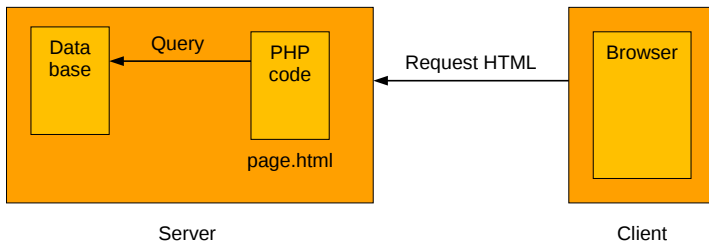
- PHP = “PHP Hypertext Preprocessor.”
- So, what does the first P stand for?
- PHP code is placed in a PHP file. . .
- . . . or it can be embedded in an HTML file.
- When the client requests a file containing PHP code, the server
 - Executes the PHP code.
 - Sends the output of the PHP code to the client.
- Typically, the output of the PHP program is HTML code and Javascript code.



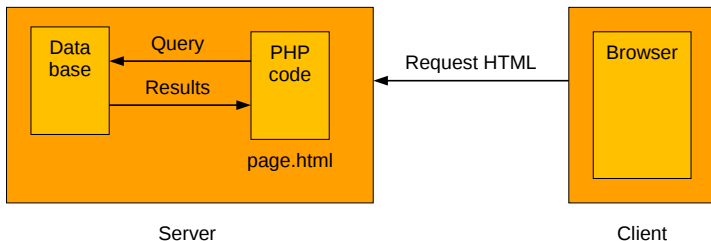
Client has browser; server has PHP code and database



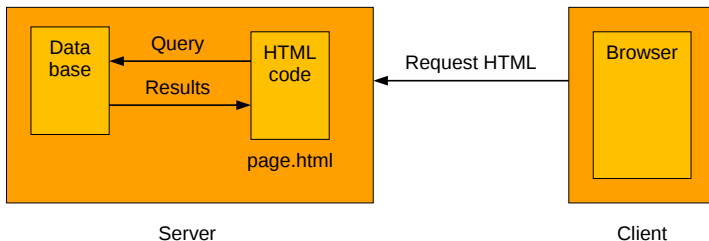
Client requests HTML file



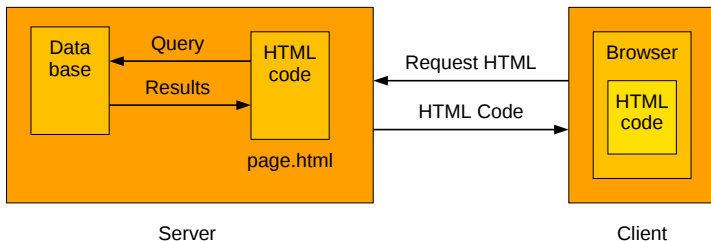
PHP code sends query to database



Database returns results



PHP code outputs HTML code



HTML code sent to client

PHP Code

```
<?php  
    PHP_code  
?>
```

- PHP code is opened by `<?php` and is closed by `?>`.

Outline

- 1 PHP
- 2 **The `echo` Statement**
- 3 Variables
- 4 Operators
- 5 Decision Structures

The `echo` Statement

The `echo` Statement

```
echo expression;
```

- The `echo` statement will output the value of an expression.

The `echo` Statement

The `echo` Statement

```
<?php  
    echo "Hello, world!";  
?>
```

- For example, the above code will output `Hello, world!`

Strings and Quotation Marks

```
<?php
    echo "Hello, 'world!' ";
    echo 'Hello, "world!"';
?>
```

- Strings may be delimited by either single quotes (') or double quotes (").
- Single quotes may be embedded within double quotes, and vice versa.
- The above code will output `Hello, 'world!'` and `Hello, "world!"`

The `echo` Statement

Concatenation

```
echo "Hello," . " world!";
```

- The dot (.) is the string concatenation operator.
- The above code will output

```
Hello, world!
```

Mixing PHP and HTML

Mixing PHP and HTML

```
<html>
<head>
<?php
    echo "<title>My Web Page</title>";
?>
</head>
<body><?php echo "<h1>My Web Page</h1></body></html>" ?>
```

- Mixing PHP and HTML can be ugly, but it works.

The Extended `echo` Statement

The Extended `echo` Statement

```
echo <<<HTML  
  long section of text  
HTML;
```

- Sometimes we want to echo very long string.
- We may use whatever delimiter we like. I suggest `HTML` because this form is used typically to output HTML code.
- Terminating delimiter *must* begin in column 1 and there must be *nothing else* on that line.
- This is better than writing *many* `echo` statements.

The Extended `echo` Statement

The Extended `echo` Statement

```
echo <<<HTML
<html>
<head>
<title>My Web page</title>
<style>
h1 {text-align: center; color: olive;}
</style>
<head>
<body>
HTML;
```

Outline

- 1 PHP
- 2 The `echo` Statement
- 3 Variables**
- 4 Operators
- 5 Decision Structures

Assignment Statements and Data Types

```
$x = "Hello";      // String type  
$x = 44 + 55;      // Integer type  
$x = $x . " bottles of beer on the wall"; // What type?  
echo $x;
```

- Variable names begin with \$.
- A variable's type can change.
- A variable's current type is determined by its current value.
- When the value changes, the type may change.

Variables

Variables Within Strings

```
$x = 44 + 55;  
echo "$x bottles of beer on the wall";  
echo '$x bottles of beer on the wall';
```

- What will the above code produce?

Outline

- 1 PHP
- 2 The `echo` Statement
- 3 Variables
- 4 Operators**
- 5 Decision Structures

Operators

- PHP operators
 - The assignment operator: =
 - Numerical operators: +, -, *, /, %, ++, --
 - String operators: .
 - Comparison operators: ==, !=, <, >, <=, >=, ===, !==
- The expression `$a === $b` is true if `$a` and `$b` have the same value **and** are of the same type.
- The expression `$a !== $b` is true if `$a` and `$b` have different values **or** are of different types.

Debugging Programs

Inspecting Variables

```
echo 'The value of $x is ' . $x . '<br/>';
```

- The most common method of debugging a PHP program is to inspect the values of variables.

Example

Example

```
$fname = 'Alice';  
$lname = 'Smith';  
$ssn = '123-45-6789';  
$bdate = '1968-05-22';  
$salary = 35000.00;
```

- Use PHP to create a table, with headings, that fills row 1 with the above data.

Example

Example

```
echo <<<HTML
<table border="1">
<tr>
    <th>Name</th><th>SSN</th><th>Birthday</th>
    <th>Salary</th>
</tr>
<tr>
    <td>$fname $lname</td><td>$ssn</td><td>$bdate</td>
    <td>$salary</td>
</tr>
</table>
HTML;
```

Outline

- 1 PHP
- 2 The `echo` Statement
- 3 Variables
- 4 Operators
- 5 Decision Structures**

if Statements

The if Statement

```
if (condition)  
    true-block;  
else  
    false-block;
```

- if statements in PHP are just like if statements in C.
- The else part is optional.
- If the block contains more than one statement, then group them using curly braces.

if Statements

Example

```
$online = value from database;
```

- In the last example, suppose that we have another variable `$online` which is 1 if the person is online and 0 if the person is not online.
- Add one more column to the table named `Online`. Fill in `Yes` or `No`, depending on whether the person is online.

Example

Example

```
echo <<<HTML
<table border="1">
<tr>
    <th>Name</th><th>SSN</th><th>Birthday</th>
    <th>Salary</th><th>Online</th>
</tr>
<tr>
    <td>$fname $lname</td><td>$ssn</td><td>$bdate</td>
    <td>$salary</td>
HTML;
if ($online == 1)
    echo "<td>Yes</td>";
else
    echo "<td>No</td>";
echo "</tr></table>";
```

The switch Statement

```
switch (expression)
{
    case value:
        case-block;
        break;
    :
    default:
        default-block;
        break;
}
```

- switch statements in PHP are just like switch statements in C.