

PHP - Part 2

More operators...

Arithmetic and Assignment Operators

- e.g., using + and =
 - `$intA=5; $intB=8; $intC=$intA+$intB; //intC is 13`
 - `// Same +, -, *, / and % as C`
 - `$intA += $intB; //as in C`
- Bitwise:
 - `&, |, ^, ~, << and >>`
 - e.g., `$intA=7; $intB=9;`
 - `$intC = $intA & $intB;`

Comparison Operators

- == //true if equal
- === //true if **identical** (i.e., same type and value)
- !=, <> //true if not equal
- !== //true if not **identical**
- >, <, <=, >=
- **Ternary operators:**
 - (expre1) ? (expre2) : (expre3); //expre2 if expre1 true
 - \$strA = (\$intValue >0) ? “positive” : “zero or negative”;

String Operators

- Concatenate with `.`
 - `$strResult = $strOne . $strTwo;`
- Convert other types to string:

```
$intNumber = 45;
```

```
$strAgeis = "My age is";
```

```
$strResult = $strAgeis . " " . $intNumber;
```

```
echo $strResult;
```

String Processing

- Strings specified using single or double quotes

```
$str="hello";
```

```
$str='hello';
```

- Single quotes are literal

```
$myStr='$str one';
```

- Double quotes substitute the content of variables

```
$myStr="$str world";
```

- Use curly braces if variable name is joined to more text

```
$myStr="${str}world";
```

Substrings

- `$subStr=substr($str, int $start [, int $length]);`
 - Extracts portion of \$str
- `$count=substr_count($str, 'text') ;`
 - Counts the number of occurrences of text in the string
- `$restStr=strstr($str, 'text') ;`
 - Extract substring of \$str from first occurrence of 'text'
- `strlen($str)`
 - Length of a string
- `$str{0}`
 - Access individual characters in a string
- `$newStr=$Str.'more text'`
 - Concatenate strings using the dot '.' operator

Logical Operators

- And
- Or
- Xor
- !
- &&
- ||

Error Control Operator (@)

- Example:

```
$intA = 58;
```

```
$int B = 0;
```

```
@$intC = $intA / $intB; //no error message...
```

```
print "<br><br>Is it possible that " . "$intA/$intB" . "=" . "$intC" . "? <br>";
```

PHP Built in variables

```
$GLOBALS  
$_SERVER  
$_GET  
$_POST  
$_COOKIE  
$_FILES  
$_ENV  
$_REQUEST  
$_SESSION
```

But be careful: many are server-dependent

Try using **print_r()** on these.

PHP Built in Variables.

```
print "<p>My host name is " . $_SERVER['HTTP_HOST'] . "<br/>\n";

print "<p>I'm viewing this page from " . $_SERVER['HTTP_USER_AGENT'] . "<br/>";

print "We can split the Browser string into a new array using split()<br/>";

$userBits = split(" ", $_SERVER['HTTP_USER_AGENT']);
print "The browser is identified as <b>" . $userBits[0] . "</b> ";
print "or you can split this up too!<br/>";
$theBrowserID = split("/", $userBits[0]);

print "The browser is advertising itself as <b>" . $theBrowserID[0] . " ";
print $theBrowserID[1] . "</b><br/>\n";

print "Of course the real browser is <b>";
$lastIndex = count($userBits) - 1;
$realBrowser = split("/", $userBits[$lastIndex]);
print $realBrowser[0] . "</b> version <b>" . $realBrowser[1] . "</b> <br/>\n";

print "My browser can accept " . $_SERVER['HTTP_ACCEPT'] . "<br/>\n";
print "My proxy server (if I have one) is " . $_SERVER['HTTP_VIA'] . "<br/>\n";
print "Document root is " . $_SERVER['DOCUMENT_ROOT'] . "<br/>\n";
print "This page is called " . $_SERVER['PHP_SELF'] . "<br/>\n";
```

Sample Output

My host name is **localhost:8080**

I'm viewing this page from **Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US)**

AppleWebKit/533.4 (KHTML, like Gecko) Chrome/5.0.375.125 Safari/533.4

We can split the Browser string into a new array using split()

The browser is identified as **Mozilla/5.0** or you can split this up too!

The browser is advertising itself as **Mozilla 5.0**

Of course the real browser is **Safari** version **533.4**

My browser can accept

application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*
;q=0.5

My proxy server (if I have one) is

Document root is

This page is called **webpages/phptest/php-built-in-variables.php**

How to use Forms with PHP

PHP Processing Form Variables

- Recall how CGI used POST and GET methods

- In PHP:

Extract submitted form variables from:

\$_GET

\$_POST

\$_REQUEST (also contains variables but may violate security by using the wrong method compared to the application design)

Submitted files can be extracted from:

\$_FILES (...more details later)

Example using POST - HTML

```
<form action="action_part2_example1.php" method="post">  
<div><label>Number 1: <input name="m"  
  size="5"></label></div>  
<div><label>Number 2: <input name="n"  
  size="5"></label></div>  
<div><input type="submit" value="Multiply"></div>  
</form>
```

Action using POST - PHP

```
<h2>Multiply Using PHP with POST</h2>
```

```
<?php print "Apache receives the following array: ";
```

```
print_r($_POST)
```

```
$intResult = $_POST['m'] * $_POST['n'];
```

```
print "The result of " . (int)$_POST['m'] . "*" . $_POST['n'] . "=" . $intResult;
```

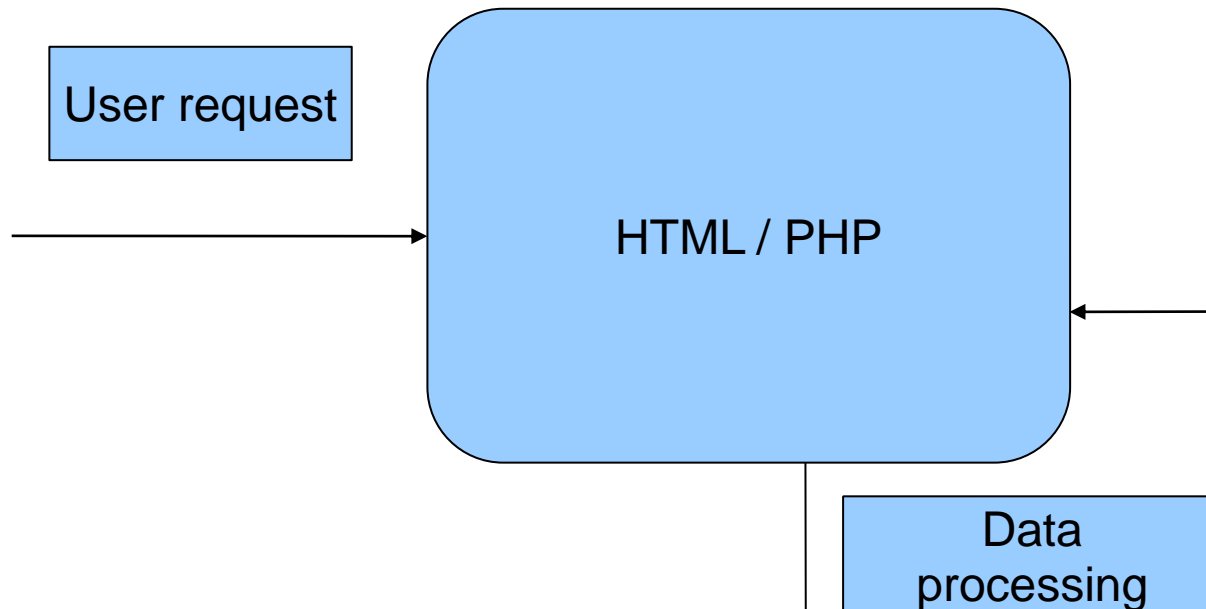
```
?>
```


Exercise6

- Copy the previous code and change the method to **GET** and **REQUEST**.
- Try to combine the array examples with forms.

Combining PHP with forms

- Recall the code for a self-generating CGI script
- Combining HTML with PHP



HTML/PHP

```
<form action='<?php echo $ SERVER["PHP_SELF"];?>' method="post">
```

```
<div><label>Number 1: <input name="m" size="5"></label></div>
```

```
<div><label>Number 2: <input name="n" size="5"></label></div>
```

```
<div><input type="submit" name="submit" value="Multiply"></div></form>
```

```
<h2>Self generating Multiply Using single PHP file with POST</h2>
```

```
<?php print "Apache receives the following array: ";print_r($_POST) ?>
```

```
<?php
```

```
if (isset($_POST['submit'])) {
```

```
    $intResult = $_POST['m'] * $_POST['n'];
```

```
    print "The result of " . (int)$_POST['m'] . " * " . $_POST['n'] . " = " . $intResult;
```

```
} else { echo "This is the first time the page is loaded<br>";}
```

```
?>
```

File processing with PHP

File Processing

- The normal technique for storing permanent information on the server side is using a database
- Sometimes storage in flat files is useful
 - When database storage is overkill
 - During development and testing of code
 - Rapid prototyping
 - When saving specific formats

Basic File Processing

- Open a file for writing
- Write to the file
- Close the file

- Open a file for reading
- Read from the file
- Close the file

Opening Files

- `$fp = fopen("file.txt", "r");`
 - Open a file for reading
- `$fp = fopen("file.txt", "w");`
 - Open a file for writing
 - Note depending on operating system (i.e., Windows) file paths might need to be escaped
 - `"\\pathtofile\\filename.txt"`

Reading a File

- `$contents = fread($fp, filesize($filename));`
 - Reads whole of file into one string
 - Poor performance for large files
- `$contents = fgets($fp, 4096);`
 - Reads one line or the number of bytes specified
 - Whichever is less
- `$contents = file_get_contents($filename)`
 - Efficient way to read whole file into string

Writing to a File

- `fwrite($fp, $outputstring);`
 - Write string out to given file pointer
- `fwrite($fp, $outputstring, 80);`
 - Write first 80 characters to output string

Closing Files

- `fclose ($fp) ;`
 - Close given file pointer
 - Normally won't be an error.

Superglobals

From **version 4.1.0 onward**, PHP provides an additional set of predefined arrays containing variables from the web server (if applicable), the environment, and user input.

- **automatically global**-i.e., automatically available in every scope.
- For this reason, they are often known as "**superglobals**".
- There is no mechanism in PHP for user-defined superglobals.
- You'll notice how the older predefined variables (***\$HTTP_*_VARS***) still exist.
- As of PHP 5.0.0, the long PHP predefined variable arrays may be disabled with the **register_long_arrays** directive.

Demo

- welcome_html.htm, welcome.php
- php_superglobals.php

Look for QUERY_STRING, \$_GET

- part2_example1.php
- part2_example2.php
- part2_example3.php

Inspect using web browser, try modifying the URL' to indicate new parameters

EasyPHP

Apache: (httpd.conf)

```
cgi.force_redirect = 0
```

```
Listen 127.0.0.1:5080
```

PHP.ini

```
variables_order = "EGPCS"
```

```
request_order = ""
```

```
register_long_arrays = Off
```

```
register_globals = Off
```