

PHP

ANGEL R. ROMAN
KEYLA PEREZ VELEZ
MICHAEL VOSS
JULIAN RICHEN

GROUP 12



QUESTIONS TO BE ADDRESSED

- + History
- + Development
- + Versioning
- + Compatibility
- + Usage



Introduction

PHP is a recursive acronym for "PHP: Hypertext Preprocessor". It's a scripting language used to develop web pages within HTML codes. PHP is a server-side scripting language, which means that the code is executed on the web server and not on the web browser, as a result the client cannot view the code. PHP is a combination of different programming languages such as Java, C and Perl. The syntax of this scripting language is close to that of C programming language.



PHP

Was created by Rasmus Lerdorf in 1994. It was initially developed for HTTP usage logging and server-side form generation in Unix



PHP 2 (1995)

Transformed the language into a Server-Side embedded scripting language. Added database support, file uploads, variables, arrays, recursive functions, conditionals, iteration, regular expressions, etc.



PHP 3 (1998)

Added support for ODBC data sources, multiple platform support, email protocols (SNMP, IMAP), and new parser written by Zeev Suraski and Andi Gutmans.



PHP 4 (2000)

Became an independent component of the web server for added efficiency. The parser was renamed the “Zend Engine”. Many security features were added.



PHP 5 (2004)

Added “Zends Engine II” with object oriented programming, robust XML support using the libxml2 library, SOAP extension for interoperability with Web Services, SQLite has been bundled with PHP. Was the first version to really advance since it added Object Oriented programming.

PHP 6 (NEVER RELEASED)

The decision was to use Full UTF-16 support which negatively impacted performance. A Large population realized this problem in the open source community and weren't interested in the project.



PHP 7 (2015)

Added “Zends Engine III” with numerous improvements and new features such as reduce memory usage, consistent 64-bit support, secure random number generator, return and scalar type declarations, anonymous classes, and zero cost asserts. (Side note: PhP7 is twice as fast as PhP5.6

COMPATIBILITY

Web Servers

Apache, NGINX,
Microsoft IIS, Caudium,
Netscape Enterprise
Server

Operating Systems

UNIX, Mac OS X,
Windows
NT/98/2000/XP/2003

Databases

Adabas D, dBase,
Empress, FilePro,
Hyperwave, IBM DB2,
Direct MS-SQL, MySQL,
ODBC, Oracle, Ovrimos,
PostgreSQL, SQLite,
Solid, Sybase, Velocis,
Unix dbm



244,000,000

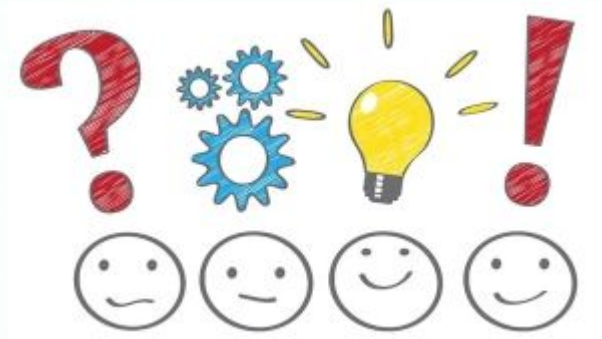
Websites have PHP installed



2,100,000

Web Servers have PHP Installed

QUESTION!



LEARN PHP*

In 2 minutes

**Basic run down, works best if you know another language already.*



PHP TAGS

```
<?php
```

```
// php code
```

```
?>
```

This won't work:

```
<?php//php code?>
```

This will work:

```
<?php //php code ?>
```


VARIABLES, TYPING, & PRINTING

- + Variables start with \$ sign
 - + Typing is done automatically
 - + Variables can switch between types
 - + Printing does not need to be formatted
 - + Concatenation is done with periods
 - + Double quotes display escaped characters
 - + Single quotes almost always display message as-is
- + Types:
 - + boolean/bool
 - + integer/int
 - + float/double
 - + string
 - + array
 - + object
 - + callable
 - + null
 - + resource



VARIABLES, PRINTING, & COMMENTS

```
<?php
```

```
$exampleString = "Hello World";  
$exampleInt = 10;  
$exampleBool = true;
```

```
echo "Hello World\n"; // Hello World  
echo "String: " . $exampleString . "\n"; // String: Hello World  
echo "Int: $exampleInt\n"; // Int: 10  
echo "Bool: {$exampleBool}ish" . "\n"; // Bool: 1ish  
echo 'Fails: $exampleString' . "\n"; // Fails: $exampleString
```

YOU CAN ALSO USE THE PRINT() FUNCTION

```
<?php
```

```
print ("My String\n"); // My String
```



ADD DIE() TO KILL THE SCRIPT AND PRINT STRING

```
<?php
```

```
die ("Critical failure, abort!\n");
```

```
// or..
```

```
exit (1); // Error code or string
```

CONSTANTS

```
<?php
```

```
define ('A_DEFINE_MSG', 'Hello World');
```

```
echo A_DEFINE_MSG . "\n";
```

```
// or...
```

```
echo constant ("A_DEFINE_MSG") . "\n";
```

TYPES ARE NOT ENFORCED

```
<?php
```

```
$example = "I'm a string!";
```

```
echo "$example\n"; // I'm a string!
```

```
$example = 52;
```

```
echo "$example\n"; // 52
```

OPERATORS & EXAMPLES OF CASTING

```
<?php
```

```
$a = (false && true);  
$b = (true || false);  
$c = (false and false);  
$d = (true or false);
```

```
echo (int)$a; // 0  
echo (int)$b; // 1  
echo (int)$c; // 0  
echo (int)$d; // 1
```

```
<?php
```

```
$add = 2 + 2;  
$sub = 8 - 5;  
$div = 8 / 2;  
$mlt = 9 * 9;
```

```
echo "2 + 2 = {$add}\n"; // 2 + 2 = 4  
echo "8 - 5 = {$sub}\n"; // 8 - 5 = 3  
echo "8 / 2 = {$div}\n"; // 8 / 2 = 4  
echo "9 * 9 = {$mlt}\n"; // 9 * 9 = 81
```

IF STATEMENT

```
<?php

$status = true;

if ($status == true) {
    echo "true\n";
} else if ($status == false) {
    echo "false\n";
} else {
    echo "Not bool\n";
}

// Echos: true
```



ARRAYS & LOOPS

```
<?php
```

```
$arr1 = array ("a", "b", "c");  
$arr2 = ["apple", "banana", "clementine"];
```

```
echo $arr1[0] . ", " . $arr1[1] . ", & " . $arr1[2] . "\n";  
// a, b, & c
```

```
for ($i = 0; $i < count ($arr2); $i++) {  
    echo "{$arr2[$i]} ";  
}  
// apple banana clementine
```

FUNCTIONS

```
<?php
```

```
function funcName ($param1, $param2 = "") {  
    return "$param1 $param2";  
}
```

```
echo funcName ("Hello", "World"); // Hello world
```

```
echo funcName ("Bonjour"); // Bonjour
```

CLASSES & INITIALIZING THEM

```
<?php
```

```
class Calculator {
```

```
    private $total = 0;
```

```
    public function add ($a) {  
        $this->total += $a;  
    }
```

```
    public function sub ($s) {  
        $this->total -= $s;  
    }
```

```
    public function result () {  
        return $this->total;  
    }
```

```
$calc = new Calculator ();
```

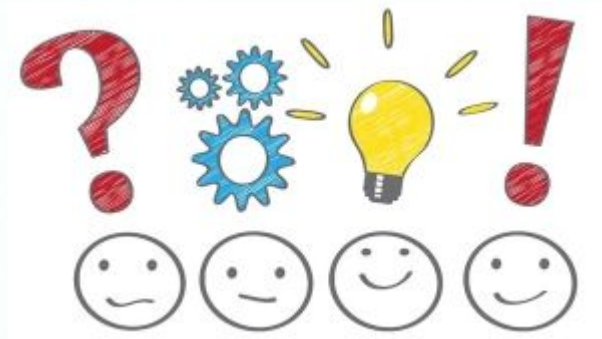
```
$calc->add (10);
```

```
$calc->sub (5);
```

```
echo $calc->result (); // 5
```

```
}
```

QUESTION!



CONCLUSION

PHP is a powerful and useful scripting language and interpreter used in many modern websites. It has many uses, but is commonly implemented to create dynamic web content. PHP is included in over 240 million websites and rising!



THANKS!

Any questions?



REFERENCES

- + "PHP: History of PHP - Manual." *PHP: History of PHP - Manual*. N.p., n.d. Web. 14 Nov. 2016.
- + Pohjolainen, Jessi. "Introduction to PHP." TAMK University of Applied Sciences, n.d. Web. 23 Sept. 2008.

