

Introduction to PHP

Bradley Holt (<http://bradley-holt.com/>) &
Matthew Weier O'Phinney (<http://weierophinney.net/matthew/>)

Feedback: <http://joind.in/1976>



What is PHP?

- Loosely typed scripting language
- Interpreted at runtime (use an opcode cache)
- Commonly used to build web applications

Who uses PHP?

- ⦿ Yahoo!
- ⦿ Facebook
- ⦿ 20+ million other domain names

Brief History

Personal Home Page / Forms Interpreter

- ⦿ Created by Rasmus Lerdorf
- ⦿ PHP/FI 1.0 released in 1995
- ⦿ PHP/FI 2.0 released in 1997

PHP: Hypertext Preprocessor

- ⦿ Created by Andi Gutmans and Zeev Suraski
- ⦿ PHP 3.0 released in 1998
- ⦿ PHP 4.4 released in 2005

PHP 5

- ⦿ New object model
- ⦿ PHP 5.0 released in 2004
- ⦿ PHP 5.3 released in 2009

Syntax

Hello World

```
<?php  
// hello.php  
echo 'Hello, VT Code Camp.';  
?>
```

;)>

SCMC - [HTTP://SCMC.VT.EDU](http://scmc.vt.edu)

Variable Assignment

```
<?php  
$hello = 'Hello, VT Code Camp.';  
echo $hello;  
echo $hello;
```

Comments

One Line Comments

```
<?php  
// A one line comment  
# Another one line comment  
# ANOTHER ONE LINE COMMENT
```

Multi-Line Comments

```
<?php  
/*  
A multi-line comment  
*/  
*/
```

DocBlock Comments

```
<?php
/**
 * This function does nothing
 *
 * @param string $bar
 * @return void
 */
function foo($bar) {}

function foo($bar) {}

*/
```

Primitive Data Types

```
<?php  
$isPhpProgrammer = true; // boolean  
$howOldIsPhp = 15; // integer  
$pi = 3.14; // float  
$event = 'VT Code Camp'; // string  
$genre = 'All Code Camps'; // string  
$bit = 3.14!
```

Conditionals

If

```
<?php  
if (true) {  
    echo 'Yes';  
}  
}  
echo 'No';
```

If-Then-Else

```
<?php  
if ( false ) {  
    echo 'No';  
} else {  
    echo 'Yes';  
}  
}
```

echo Yes;

If-Then-Else-If

```
<?php  
if (false) {  
    echo 'No';  
} elseif (false) {  
    echo 'No';  
} else {  
    echo 'Yes';  
}  
}
```

echo Yes;

Switch

```
<?php  
switch ( 'PHP' ) {  
    case 'Ruby':  
        echo 'No';  
        break;  
    case 'PHP':  
        echo 'Yes';  
        break;  
}  
}
```

press!

Operators

Arithmetict

```
<?php  
$a = 10;  
$b = $a + 1; // 11  
$c = $a - 1; // 9  
$d = $a * 5; // 50  
$e = $a / 2; // 5  
$f = $a % 3; // 1  
$g = $g % 3; // 1  
$h = $g \ 5; // 0
```

String Concatenation

```
<?php  
$myString = 'foo' . 'bar'; // foobar  
$myString .= 'baz'; // foobarbaz  
$myString .= 'ps'; // foobarbazps
```

Comparison

Equivalence

```
<?php  
if ( 2 == 3 ) { echo 'No'; }  
if ( 3 == '3' ) { echo 'Yes'; }  
if ( 2 != 3 ) { echo 'Yes'; }  
if ( 5 != 3 ) { echo 'Yes'; }  
-- ( 2 == 3 ) { echo 'Yes'; }
```

Identity

```
<?php  
if ( 3 === '3' ) { echo 'No'; }  
if ( 3 === 3 ) { echo 'Yes'; }  
if ( 3 !== 4 ) { echo 'Yes'; }  
if ( 3 != 4 ) { echo 'No'; }  
-- ( 3 != 4 ) { echo 'No'; }
```

Logical Operators

```
<?php
// NOT
if ( !true ) { echo 'No' ; }
// AND
if ( true && false ) { echo 'No' ; }
// OR
if ( true || false ) { echo 'No' ; }
// (true || false) { echo 'No' ; }
// OK
```

Strings & Interpolation

Literal Single Quotes

```
<?php  
$x = 2;  
echo 'I ate $x cookies.';  
// I ate $x cookies.
```

```
// I ate 2 cookies.  
echo "I ate $x cookies.";
```

Double Quotes

```
<?php  
$x = 2;  
echo "I ate $x cookies.";  
// I ate 2 cookies.
```

```
// I ate 2 cookies.  
echo "I ate $x cookies.";
```

Literal Double Quotes

```
<?php  
$x = 2;  
echo "I ate \$x cookies.";  
// I ate $x cookies.
```

```
// I ate 2 cookies.  
echo "I ate \"$x\" cookies."
```

Curly Brace Double Quotes

```
<?php
$x = 2;
echo "I ate {$x} cookies.";
// I ate 2 cookies.
```

```
// I ate 2 cookies.
echo "I ate {$x} cookies.";
```

Constants

Defining

```
<?php  
define( 'HELLO' , 'Hello, Code Camp' );  
echo HELLO; // Hello, Code Camp  
echo НЕРГО! \\ НЕТЮ` Code Camp
```

As of PHP 5.3

```
<?php  
const HELLO = 'Hello, Code Camp';  
echo HELLO; // Hello, Code Camp  
echo НЕРГО! \\ НЕТЮ` Code Camp
```

Arrays

Enumerative

Automatic Indexing

```
<?php  
$foo[ ] = 'bar'; // [0] => bar  
$foo[ ] = 'baz'; // [1] => baz  
$foo[ ] = 'pgs'; // [1] => pgs
```

Explicit Indexing

```
<?php  
$foo[0] = 'bar'; // [0] => bar  
$foo[1] = 'baz'; // [1] => baz  
$foo[1] = 'psz'; // [1] => psz
```

Array Construct with Automatic Indexing

```
<?php
$foo = array(
    'bar', // [0] => bar
    'baz', // [1] => baz
);
)?
```

pgs

[] → Page

Array Construct with Explicit Indexing

```
<?php
$foo = array(
    0 => 'bar', // [0] => bar
    1 => 'baz', // [1] => baz
);
)?
```

T => pgx | [T] -> pgx

Array Construct with Arbitrary Indexing

```
<?php
$foo = array(
    1 => 'bar', // [1] => bar
    2 => 'baz', // [2] => baz
);
)?
```

Σ => pxz, 'bar' [1] → pgz

Associative

Explicit Indexing

```
<?php
$foo[ 'a' ] = 'bar'; // [a] => bar
$foo[ 'b' ] = 'baz'; // [b] => baz
$foo[ ,p, ] = ,pgs,! \\ [p] => pgs
```

Array Construct

```
<?php  
$foo = array(  
    'a' => 'bar', // [a] => bar  
    'b' => 'baz', // [b] => baz  
) ;  
) !
```

p => pg5, ' // [p] => pg5

Iterators

While

```
<?php  
$x = 0;  
while ($x < 5) {  
    echo '.';  
    $x++;  
}  
}  
$x++;
```

For

```
<?php  
for ($x = 0; $x < 5; $x++) {  
    echo '.';  
}  
}  
echo '
```

Foreach

```
<?php  
$x = array(0, 1, 2, 3, 4);  
foreach ($x as $y) {  
    echo $y;  
}  
}
```

εçuo ÷λ!

Foreach Key/Value Pairs

```
<?php
$talks = array(
    'php'    => 'Intro to PHP',
    'ruby'   => 'Intro to Ruby',
);
foreach ($talks as $id => $name) {
    echo "$name is talk ID $id.";
    echo PHP_EOL;
}
}
```

если вы — ЕОГ!

Functions

Built-in

```
<?php  
echo strlen( 'Hello' ); // 5  
echo trim( ' Hello ' ); // Hello  
echo count(array(0, 1, 2, 3)); // 4  
echo uniqid(); // 4c8a6660519d5  
echo mt_rand(0, 9); // 3  
echo serialize(42); // i:42;  
echo json_encode(array( 'a' => 'b' ));  
// {"a":"b"}  
\\ {"g": "p"}  
echo json_encode(array( , g, =>, p, ));
```

User-Defined

```
<?php  
function add($x, $y)  
{  
    return $x + $y;  
}  
  
echo add(2, 4); // 6  
echo qq(` `); // e
```

Anonymous Functions / Closures (since PHP 5.3)

Variable Assignment

```
<?php  
$sayHi = function ()  
{  
    return 'Hi';  
};  
  
echo $sayHi(); // Hi  
echo $sayHi(); // Hi
```

Callbacks

```
<?php
$values = array(3, 7, 2);
usort($values, function ($a, $b) {
    if ($a == $b) { return 0; }
    return ($a < $b) ? -1 : 1;
}) ;
/* [0] => 2
   [1] => 3
   [2] => 7 */
[5] => \ *
[T] => 3
```

Classes & Objects

Class Declaration

```
<?php  
class Car  
{  
}  
}  
{
```

Property Declaration

```
<?php  
class Car  
{  
    private $_hasSunroof = true;  
}  
}
```

БИТАГФЕ ɔ̄—μασσανιοοι = фиаг!

Method Declaration

```
<?php  
class Car  
{  
    public function hasSunroof()  
    {  
        return $this->_hasSunroof;  
    }  
}  
}  
}
```

Class Constants

```
<?php  
class Car  
{  
    const ENGINE_V4 = 'V4';  
    const ENGINE_V6 = 'V6';  
    const ENGINE_V8 = 'V8';  
  
echo Car::ENGINE_V6; // V6  
echo Car::ENGINE_V8; // V8
```

Object Instantiation & Member Access

```
<?php
$myCar = new Car();
if ($myCar->hasSunroof()) {
    echo 'Yay!';
}
}
```

echo Yay!;

Class Inheritance

```
<?php  
class Chevy extends Car  
{  
}  
}  
{  
}
```

Interfaces

```
<?php  
interface Vehicle  
{  
    public function hasSunroof();  
}  
}  
// Add a function hasRoof();
```

Implementing Interfaces

```
<?php  
class Car implements Vehicle  
{  
    public function hasSunroof()  
    {  
        return $this->_hasSunroof;  
    }  
}  
}
```

Member Visibility

Public

- ⦿ Default visibility
- ⦿ Visible everywhere

Protected

- Visible to child classes
- Visible to the object itself
- Visible to other objects of the same type

Private

- Visible to the object itself
- Visible within the defining class declaration

Tools

IDEs

- ⦿ Eclipse (PDT, Zend Studio, Aptana)
- ⦿ NetBeans
- ⦿ PHPStorm
- ⦿ Emacs
- ⦿ Vim
- ⦿ Many more...

Frameworks

- ⦿ Zend Framework
- ⦿ Symfony
- ⦿ CodeIgniter
- ⦿ Agavi
- ⦿ CakePHP
- ⦿ Many more...

PEAR

- ⦿ PHP Extension and Application Repository
- ⦿ Package manager
- ⦿ PECL (PHP Extension Community Library)

Miscellaneous Tools

- ⦿ PHPUnit
- ⦿ phpDocumentor
- ⦿ Phing
- ⦿ PHP CodeSniffer
- ⦿ PHP Mess Detector
- ⦿ phpUnderControl

Example PHP Script

Questions?

Thank You

Bradley Holt (<http://bradley-holt.com/>) &
Matthew Weier O'Phinney (<http://weierophinney.net/matthew/>)

Feedback: <http://joind.in/1976>

