

Programming Concepts Syntax Guide:

Author: Ernest Edinboro

Coauthor: "Aneesh Sreedhara"

Version 1.0

Date: Summer 2018

Helpful links

PHP.net language reference guide	PHP	
W3Schools PHP	PHP	See links in the left column
JavaScript Guide	JS	
W3Schools JavaScript	JS	See links in the left column

JavaScript Review and PHP Primer

Step 1:

Research the Programming Concept for JavaScript, PHP, and programming language of choice.

Step 2:

Comment what the code does into the research guide. Write the code into the research guide.

Note:

I recommend that you use the example I provided otherwise you will need to research and create your own examples where needed. Answer any supplementary questions in the comments

Step 3:

Test your JavaScript code in the web browser. Does it work?

Step 4:

Test your PHP code in the web browser. Does it work?

Note:

Shodor may not have the necessary resources to accommodate your programming language of choice. Therefore, the third programming language is a research project ONLY.

Step 5:

Send the URL to your newdev/JS_testing directory to Ernie and Aaron.

Send the URL to your newdev/PHP_testing directory to Ernie and Aaron.

Step 6: ???

Note:

You will not be sharing the testing files with the public. Only the research guide.

Step 7: Profit

Basic Programming Concepts Review	How do you write that in JavaScript?	How do you write that in PHP?	
Other			
How do you comment code ?	<pre>//This is a single line comment.</pre> <pre>/*This is a multiline comment*/</pre>	<pre>//This is a single line comment.</pre> <pre>/*This is a multiline comment*/</pre>	
Variables			
1. How do you create a string variable - with double quotes - with single quote 2. What is the difference between single quotes vs double quotes in php?	<pre>/*Creates a string variable called myName and stores a value of Ernie using double quotes*/ var myName = "Ernie";</pre> <pre>/*Creates a string variable called myNameAgain and stores a value of Ernie using single quotes*/ var myNameAgain = 'Ernie';</pre>	<pre>/*Creates a string variable called myName and stores a value of Ernie using double quotes*/ \$myName="Ernie";</pre> <pre>/*Creates a string variable called myNameAgain and stores a value of Ernie using single quotes*/ \$myNameAgain='Ernie';</pre>	
How do you make an integer variable?	<pre>//How do you make an integer variable? var testIntVar = 5;</pre>	<pre>//How do you make an integer variable? \$testIntVar=5;</pre>	

How do I output a string, integer or variable to the screen	<pre>//How do I output a string, integer or variable to the screen document.write("Aneesh is awesome
"); console.log("Aneesh is awesome");</pre>	<pre>//How do I output a string, integer or variable to the screen echo "
" . \$myName . ' likes the number ' . \$testIntVar . "
";</pre>	
How do you concatenate a string variable? • Concatenate the myName variable with the string "is great" • Concatenate the myName variable with the variable myNameAgain	<pre>//How do you concatenate a string variable? //Concatenate the myName variable with the string "is great" testStrVar = myName + "is great."; //Concatenate the myName variable with the variable myNameAgain testStrVar = myName + myNameAgain;</pre>	<pre>//How do you concatenate a string variable? \$testStatement = '
' . \$myName . ' says hi!
'; //Concatenate the myName variable with the string "is great" \$testStatement = \$myName . 'is great'; //Concatenate the myName variable with the variable myNameAgain \$testStatement = \$myName . \$myNameAgain;</pre>	
String escape character			
Use the examples provided String escape characters	<pre>// Use the examples provided // String escape characters // \' to escape single quote example string: 'Ernie's' testStrVar = 'Ernie\'s';</pre>	<pre>// Use the examples provided // String escape characters // \' to escape single quote example string: 'Ernie's' \$testVar = 'Ernie\'s';</pre>	

• \' to escape single quote example string: 'Ernie's' • \" to escape double quote example output string: Ernie says, "Go Away!" • \\ escape backslash example output string: "8\2" • \\$ escape \$ dollar symbol PHP only example output string: "Display \$amount"	<pre>//\" to escape double quote example output string: // Ernie says, "Go // Away!" testStrVar = "Ernie says \"Go Away!\""; // \\ escape backslash // example output //string: "8\2" testStrVar = "8\\2"; //\\$ escape \$ dollar symbol PHP only example output string: //"Display \$amount" testStrVar = "Display Amount";</pre>	<pre>//\" to escape double quote example output string: // Ernie says, "Go // Away!" \$testVar = "Ernie says, \"Go Away!\""; // \\ escape backslash // example output //string: "8\2" \$testVar = "8\\2"; //\\$ escape \$ dollar symbol PHP only example output string: //"Display \$amount" \$testVar = 'Display \\$amount';</pre>	
Arrays			
Create an index array of names and store the following values Levi, Aaron, Ernie	<pre>//Create an index array of names and store the following values //Levi, Aaron, Ernie var mentors = ["Levi", "Aaron", "Ernie"];</pre>	<pre>//Create an index array of names and store the following values //Levi, Aaron, Ernie \$mentors = array("Levi", 'Aaron', 'Ernie');</pre>	
Using the index array output index number 1	<pre>//Using the index array output index number 1 testStrVar = mentors[1];</pre>	<pre>//Using the index array output index number 1 \$testVar = \$mentors[1];</pre>	

Create an associative array with named keys. Your Key names should be apple, banana, orange. You want to store following values Red, yellow, orange	<pre>//Create an associative array with named keys. Your Key names should be apple, banana, orange. //You want to store following values //Red, yellow, orange console.log("According to w3schools, associative arrays are not supported by JS.");</pre>	<pre>//Create an associative array with named keys. Your Key names should be apple, banana, orange. //You want to store following values //Red, yellow, orange \$fruitColors = array('apple'=>'Red', 'banana'=>'yellow', 'orange'=>'orange');</pre>	
Math Operators			
How do you use Arithmetic Operator Addition	<pre>//How do you use Arithmetic Operator Addition testIntVar = 2 + 2;</pre>	<pre>//How do you use Arithmetic Operator Addition \$testVar = 5 + 5;</pre>	
How do you use Arithmetic Operator Subtraction	<pre>//How do you use Arithmetic Operator Subtraction testIntVar = 2 - 2;</pre>	<pre>//How do you use Arithmetic Operator Subtraction \$testVar = 5 - 5;</pre>	
How do you use Arithmetic Operator Multiplication	<pre>//How do you use Arithmetic Operator Multiplication testIntVar = 2 * 2;</pre>	<pre>//How do you use Arithmetic Operator Multiplication \$testVar = 5 * 5;</pre>	
How do you use Arithmetic Operator Division	<pre>//How do you use Arithmetic Operator Division testIntVar = 2 / 2;</pre>	<pre>//How do you use Arithmetic Operator Division \$testVar = 5 / 5;</pre>	

How do you use Arithmetic Operator Modulus	//How do you use Arithmetic Operator Modulo testIntVar = 2 % 5;	//How do you use Arithmetic Operator Modulo \$testVar = 5 % 5;	
How do you use Arithmetic Operator Exponentiation	//How do you use Arithmetic Operator Exponents testIntVar = 2 ** 2;	//How do you use Arithmetic Operator Exponents \$testVar = (5*5); //Makeshift way for doing 5 squared	
Assignment Operators			
How do you use Basic Assignment Operator	//How do you use Basic Assignment Operator testIntVar = 5;	//How do you use Basic Assignment Operator \$testVar = 5;	
How do you use Assignment Operator Addition	// How do you use Assignment Operator // Addition testIntVar += 5;	// How do you use Assignment Operator // Addition \$testVar += 5;	
How do you use Assignment Operator Subtraction	// How do you use Assignment Operator // Subtraction testIntVar -= 5;	// How do you use Assignment Operator // Subtraction \$testVar -= 5;	
How do you use Assignment Operator	// How do you use Assignment Operator // Multiplication	// How do you use Assignment Operator // Multiplication	

Multiplication	testIntVar *= 5;	\$testVar *= 5;	
How do you use Assignment Operator Division	// How do you use Assignment Operator Division testIntVar /= 5;	// How do you use Assignment Operator Division \$testVar /= 5;	
How do you use Assignment Operator Modulus	// How do you use Assignment Operator // Modulus testIntVar %= 5	// How do you use Assignment Operator // Modulus \$testVar %= 5;	
How do you increment a value	// How do you // increment a value testIntVar++;	// How do you // increment a value \$testVar = ++\$testVar;	
How do you decrement a value	// How do you // decrement a value testIntVar--;	// How do you // decrement a value \$testVar = --\$testVar;	
Comparison Operators			
How do you use equal	// How do you use equal var x = testIntVar; if (x == 5) { } }	// How do you use equal if(\$testVar == 5) {}	

How do you use identical	// How do you use identical if (x === 5) { }	// How do you use identical if(\$testVar === 5) {}	
How do you use not equal	// How do you use // not equal if (x !== 5) { }	// How do you use // not equal if(\$testVar !== 5) {}	
How do you use not identical	// How do you use // not identical if (x !== 5) { }	// How do you use // not identical if(\$testVar !== 5) {}	
How do you use less than	// How do you use less than if (x < 5) { }	// How do you use less than if(\$testVar < 5) {}	
How do you use greater than	// How do you use // greater than if (x > 5) { }	// How do you use // greater than if(\$testVar > 5) {}	
How do you use less than or equal to	// How do you use // less than or equal to if (x <= 5) { }	// How do you use // less than or equal to if(\$testVar <= 5) {}	
How do you use	// How do you use // greater than or equal to	// How do you use // greater than or equal to	

greater than or equal to	if (x >= 5) { }	if(\$testVar >= 5) {}	
Control Structures			
How do you use if	//How do you use if if (x >= 5) { }	//How do you use if if(\$testVar <= 5) {}	
How do you use else	//How do you use else if (x >= 5) { } else { }	//How do you use else if(\$testVar <= 5) {} else {}	
How do you use else if	//How do you use else if if (x >= 5) { } else if (x >= 4) { } else { }	//How do you use else if if(\$testVar < 5) {} else if(\$testVar = 5) {} else {}	
How do you use switch	//How do you use switch switch (x) { case 4: break; case 5: break; default: }	//How do you use switch switch (\$mentors[1]) { case "Aaron": echo "A"; break; case "Levi": echo "L"; break; case "Ernie": echo "E";}	

Logical Operators			
How do you use and	//How do you use and y = 1; if (x == 5 && y == 5) { }	\$testVar2 = 5; //How do you use and if(\$testVar <= 5 && \$testVar2 <= 6) { else {}	
How do you use or	//How do you use or if (x == 5 x == 6) { }	//How do you use or if(\$testVar <= 5 \$testVar2 <= 6) {} else {}	
How do you use not	//How do you use not if (x != 5) { }	//How do you use not if(!(\$testVar2 = 5)) {} else {}	
Loops			
How do you use for	//How do you use //for for (i = 0; i < 5; i++) { }	//How do you use //for for (\$i = 0; \$i < 10; ++\$i) {}	
How do you use while	//How do you use //while while (y < 5) { y++; }	//How do you use //while while (\$testVar <20) {\$testVar += 1;}	

How do you use do while	//How do you use //do while do { y++; } while (y < 5);	//How do you use //do while do {\$testVar += 1;} while (\$testVar <20);	
Functions (user defined)			
How do you create a function	//How do you create a //function function testFunction() { }	//How do you create a //function function testFunction() {}	
How do you call that function	//How do you call that function testFunction();	//How do you call that function testFunction();	
How do you create a function that has one or more arguments	//How do you create a //function that has one or more arguments function testFunction2(testInput1, testInput2) { }	//How do you create a //function that has one or more arguments function testFunction2(\$input1, \$input2) {};	
How do you call that function?	//How do you call that function? testFunction2(4, 5);	//How do you call that function? testFunction(1, 2);	

How do you return a value from a function	//How do you return a value from a function function testFunction3(testInput1, testInput2) { return testInput1 + testInput2; }	//How do you return a value from a function function testFunction3(\$input1, \$input2) {return \$input1+\$input2;}	
---	---	---	--

Useful Functions	JavaScript	PHP	
String Functions			
Variable String Length	//Variable String Length var testString = "test"; testIntVar = (testString.length);	//Variable Is a String if(is_string(\$testVar2)) {} else {}	
Variable Is a String	//Variable Is a String if(typeof(testString) == 'string') { }	//Variable Is a String if(is_string(\$testVar2)) {} else {}	
Variable Find Values in String (* strpos)	//Variable Find Values in String //(* strpos) testIntVar = (" " + testString.indexOf("e"));	//Variable Find Values in String //(* strpos) \$testVar = strpos(\$myName, "e");	

Variable Returns Part of String (* substr)	//Variable Returns Part of String //(* substr) testStrVar = (" " + testString.substr(1, 2));	//Variable Returns Part of String //(* substr) \$testVar = substr(\$myName, 2);	
Variable String make lowercase	//Variable String make lowercase testString = testString.toLowerCase();	//Variable String make lowercase \$testVar = strtolower(\$myName);	
Variable String Uppercase first character in each word. (* ucwords)	//Variable String Uppercase first character in each word. //(* ucwords) document.write(" " + "After searching the internet it seems that there is no actual written in function for this in javascript, and the user must create their own function. All the internet answers seem to do that, at least.");	//Variable String Uppercase first character in each word. //(* ucwords) \$testVar = ucwords("hello my name is aneesh ");	
MORE String Function			
You research a String Function 1	//You research a //String Function 1 //lastIndexOf() returns the last location of what you are looking for. testIntVar = (" " + testString.lastIndexOf("t"));	//You research a //String Function 1 //addslashes() puts backslashes in front of specified characters. \$testVar = addslashes(\$testVar, "e");	

You research a String Function 2	<pre>//You research a //String Function 2 //search() returns the location of a value. testStrVar =("
" + testString.search("e"));</pre>	<pre>//You research a //String Function 2 //addslashes() adds backslashes in front of predefined characters. \$testVar = addslashes(\$testVar);</pre>	
You research a String Function 3	<pre>//You research a //String Function 3 //slice() is basically substr but the last number input specifies your ending index, not length. testIntVar = (testString.slice(1, 2));</pre>	<pre>//You research a //String Function 3 //chop() removes stuff like whitespace from the right end of a string. \$testVar = chop(\$testVar);</pre>	
You research a String Function 4	<pre>//You research a //String Function 4 //replace() does what you think, it replaces a specified value. testIntVar = testString.replace("st", "ss");</pre>	<pre>//You research a //String Function 4 //ord() returnsthe ASCII value of a string's first character \$testAscii = ord("hello");</pre>	
You research a String Function 5	<pre>//You research a //String Function 5 //trim() removes the whitespace from both sides of a string. testString = " test "; testString = testString.trim();</pre>	<pre>//You research a //String Function 5 //strrev() Reverses a string \$testVar = strrev(\$testVar);</pre>	

Math Functions			
Math Functions Random Number	//Math Functions Random Number var testNumber = Math.random(); // return random number between 0 and one, inclusive of 0.	//Math Functions Random Number \$testVar = lcg_value() * 100;	
Math Functions Floor	//Math Functions Floor testNumber = testNumber * 100 //Now we get some bigger numbers testNumberFloored = Math.floor(testNumber); //This returns the largest INTEGER under the input.	//Math Functions Floor \$testVarFloor = floor(\$testVar);	
Math Functions Ceiling	//Math Functions Ceiling testNumberCeiled = Math.ceil(testNumber); //This returns the smallest INTEGER over the input.	//Math Functions Ceiling \$testVarCeil = ceil(\$testVar);	
Math Functions Not a Number	//Math Functions Not a Number if (isNaN(testString)) {}	//Math Functions Not a Number if (is_NaN(\$testVar)) {}	
You research a Math Function 1	//You research a //Math Function 1	//You research a //Math Function 1	

	//round() is self explanatory, it rounds the input. testIntVar = (" " + Math.round(3.9));	//log() returns the natural log of a number \$testVar = log(\$testVar);	
You research a Math Function 2	//You research a //Math Function 2 //pow()returns the value of the first input to the power of the second input. testIntVar = (" " + Math.pow(3, 2));	//You research a //log() returns the natural log of a number \$testVar = log10(\$testVar);	
You research a Math Function 3	//You research a //Math Function 3 //sqrt() returns the square root of the input. testIntVar = (" " + Math.sqrt(9));	//You research a //Math Function 3 //pi() gives you the value of pi. \$testVar = pi();	
Array Functions			
Array Push	//Array Push var mentors = ["Levi", "Aaron", "Ernie"]; mentors.push("Joel");	//Array Push \$testVar = array_push(\$mentors, "Joel");	
Array Count Values	//Array Count Values var mentorsArrayLength = mentors.length;	//Array Count Values \$testVar = array_count_values(\$mentors);	

Array Sorting	//Array Sorting mentors = mentors.sort(); document.write(" " + mentors);	//Array Sorting \$testVar = sort(\$mentors);	

More about PHP

more String literals	PHP
HEREDOC, and NOWDOC	
HEREDOC	//HereDoc/NowDoc \$mystring = <<<EOT Look it's some text! EOT; \$testVar = <<<EOD Another cool string EOD;
NOWDOC	//HereDoc/NowDoc

	<pre> \$mystring = <<<EOT Look it's some text! EOT; \$testVar = <<<EOD Another cool string EOD; </pre>
Useful PHP Functions	
isset()	<pre> //isset() checks if variable is set or not if (isset(\$testVar)) {} </pre>
include_once()	<pre> //include_once is used to include a php file within another. //If it is found that the file has already been included, calling script is going //to ignore further inclusions. - w3resource //include_once(/*file goes in here*/); </pre>
var_dump()	<pre> //var_dump() gives information about the input. var_dump(\$testVar); </pre>
print_r()	<pre> //print_r prints stuff about a variable that can be read by humans. print_r(\$testVar); </pre>