

My Programming Concepts Syntax Guide:

Part 1

Base Programming Concepts	JavaScript	PHP	Python
Other			
Code Commenting	<pre>// Comments are good // Single line comment /* It is a good practice to leave comments so other will understand what you were trying to accomplish. Multiple line comments */</pre>	<pre>// Comments are good // Single line comment /* It is a good practice to leave comments so other will understand what you were trying to accomplish. Multiple line comments */</pre>	<pre># Comments are good # Single line comment """ It is a good practice to leave comments so other will understand what you were trying to accomplish. Multiple line comments """</pre>
Variables			
String Variable Double and Single quote	<pre>var myName = "Ernie"; var myNameAgain = 'Ernie';</pre>	<pre>\$myName = "Ernie"; \$myNameAgain = 'Ernie';</pre>	<pre>myName = "Leanne" myNameAgain = 'Leanne'</pre>
String Variable Concatenation	<pre>var firstName = "Leanne "; var lastName = "Chan" ; var fullName = firstName.concat(lastName); Returns: Leanne Chan</pre>	<pre>\$firstName = "Leanne "; \$lastName = "Chan" \$fullName = \$firstName . "Chan" Returns: Leanne Chan</pre>	<pre>firstName = Leanne lastName = Chan fullName = firstName + lastName</pre>
Integer Variable	<pre>var x = 7;</pre>	<pre>\$x = 7;</pre>	<pre>X = 7</pre>
Boolean Variable (true or false)	<pre>var x = Boolean(5 > 4); Returns: true</pre>	<pre>\$x = true; \$y = false;</pre>	<pre>X = bool(2==3) print(x) Returns: false</pre>
Arrays			
Get value in array via index	<pre>Var colour = [red, blue green];</pre>	<pre>\$colour = array(red, blue, green); Echo (\$colour[0]);</pre>	<pre>Colour = [red, blue, green] Print colour[0];</pre>

		Returns: red	Returns: red
Get value in associative array via key	<pre>Var colour = []; Colour[0] = "red"; Colour[1] = "blue"; Colour[2] = "green";</pre>	<pre>\$age = array("Bob" => "7" "Phill" => "10" "Joe" => "12"); Echo (\$age['Bob']);</pre> <p>Returns: 7</p>	<pre>Age = {"Bob" : "7", "Phill" : "10", "Joe" : "12" }</pre>
Math Operators			
Math Operator Addition	Var x = a + b;	\$x + \$y	X + y
Math Operator Subtraction	Var x = a - b;	\$x - \$y	X - y
Math Operator Multiplication	Var x = a * b;	\$x * \$y	X * y
Math Operator Division	Var x = a / b;	\$x / \$y	X / y
Math Operator Modulus	Var x = a % b;	\$x % \$y	X % y
Math Operator Exponentiation	<pre>Math.pow(a,b) //a is raised to the power of b</pre>	\$x ** \$y	X ** y
Assignment Operators			
Basic Assignment Operator	Var x = 3;	\$x = 3	Z = x + y
Assignment Operator Addition	<pre>Var x = 10; X += 3; // x now equals 13</pre>	\$x += \$y	Z += x
Assignment Operator Subtraction	<pre>Var x = 10; X -= 3; //x now equals 7</pre>	\$x -= \$y	Z +- x
Assignment Operator Multiplication	<pre>Var x = 10; X *= 3;</pre>	\$x *= \$y	Z *= x

	<pre>//x now equals 30</pre>		
Assignment Operator Division	<pre>Var x = 10; X /= 2; //x now equals 5</pre>	<pre>\$x /= \$y</pre>	<pre>Z /- x</pre>
Assignment Operator Modulus	<pre>Var x = 10; X %= 5 //x now equals 0</pre>	<pre>\$x %= \$y</pre>	<pre>Z %= x</pre>
Increment	<pre>Var x = 3; Y = x++; //y = 3 //X = 4 ----- Var x = 3; Y = ++x //y=4 //X = 4</pre>	<pre>\$x = 5; //Increments \$x by 1, then returns echo (++\$x); echo (\$x); Returns: 66 ----- \$x = 5; //Returns \$x, then increments by 1 echo (\$x++); echo (\$x); Returns: 56</pre>	<pre>N/A</pre>
Decrement	<pre>Var x = 3; Y = x--; //y = 3 //X = 2 ----- Var x = 3; Y = --x //y = 2 //X = 2</pre>	<pre>\$x = 5; //Decrements \$x by 1, then returns echo (--\$x); echo (\$x); Returns: 44 ----- \$x = 5; //Returns \$x, then increments by 1 echo (\$x--); echo (\$x); Returns: 54</pre>	<pre>N/A</pre>
Comparison Operators			
Equal	<pre>Var x = 5;</pre>	<pre>\$x == \$y</pre>	<pre>X == y</pre>

	<pre>X == 5; //Returns true X == 6; //Returns false</pre>		
Identical	<pre>Var x = 5; X === 5; //Returns true X === "5"; //Returns false</pre>	<pre>\$x === \$y</pre>	N/A
Not equal	<pre>Var x = 5; X != 6; //Returns true</pre>	<pre>\$x != \$y OR \$x <> \$y</pre>	$X \neq y$
Not identical	<pre>Var x = 5; X != 5; //Returns false X != "5"; //Returns true</pre>	<pre>\$x !== \$y</pre>	N/A
Less than	<pre>Var x = 5; X < 6; //Returns true</pre>	<pre>\$x < \$y</pre>	$X < y$
Greater than	<pre>Var x = 5; X > 6; //Returns false</pre>	<pre>\$x > \$y</pre>	$X > y$
Less than or equal to	<pre>Var x = 5; X <= 5; //Returns true</pre>	<pre>\$x <= \$y</pre>	$X \leq y$

Greater than or equal to	<pre>Var x = 5; X >= 5; //Returns true</pre>	\$x >= \$y	X >= y
Logical Operators			
And	<pre>x=5; y=3; (x < 6 && y >2) //Returns true</pre>	\$x and \$y	(X and y)
Or	<pre>x=5; y=3; (x == 5 y == 5) //Returns true</pre>	\$x or \$y	(X or y)
Not	<pre>x=5; y=3; !(x == y) //Returns true</pre>	! \$x	not(x and y)
Control Structures			
If	<pre>If (x == 5) { Text = "x is equal to 5" }</pre>	<pre>\$x = 5; if(\$x == 5) { echo ("x is equal to 5"); }</pre> <p>Returns: x is equal to 5</p>	<pre>If x = 5: Print "x is equal to 5"</pre>
Else	<pre>If (x == 5) { Text = "x is equal to 5" } Else { Text = "x is not equal to 5" }</pre>	<pre>\$x = 4; if(\$x == 5) { echo ("x is equal to 5"); } else { echo ("x is not equal to 5"); }</pre> <p>Returns: x is not equal to 5</p>	<pre>If x = 5: Print "x is equal to 5" Else: Print "x is not equal to 5"</pre>

Else if	<pre>If (x == 5) { Text = "x is equal to 5" } Else if (x == 4) { Text = "x is equal to 4" } Else { Text = "x is not equal to 5 or 4" }</pre>	<pre>\$x = 4; if(\$x == 5) { echo ("x is equal to 5"); } else if (\$x == 4) { Echo ("x is equal to 4"); } else { echo ("x is not equal to 5"); } Returns: x is equal to 4</pre>	<pre>If x = 5: Print "x is equal to 5" Elif x = 4: Print "x is equal to 4" Else: Print "x is not equal to 4 or 5"</pre>
Switch	<pre>Var day; Switch (new Date().getDay()) { Case 0: day = Monday; break; Case 1: day = "Tuesday" }</pre>	<pre>switch(\$x){ case 5; Echo "case is equal to 5"; Case 6; Echo "case is equal to 6"; }</pre>	N/A
Loops			
For	<pre>for(i = 1; i <=10; i++) { Text += "The number is " + i; }</pre>	<pre>for(\$i = 1; \$i <=10; \$i++) { Echo "i is equal to \$i"; }</pre>	<pre>For i in range(0,3): Print "i is equal to", i</pre>
While	<pre>While (i <10) { Text += "The number is " + i; i++; }</pre>	<pre>\$i = 1 While(\$i <=10) { Echo "i is equal to \$i"; \$i++; }</pre>	<pre>While i < 10: Print "i is equal to", i</pre>
Do while	<pre>Do { Text += "The number is " +i; i++; }</pre>	<pre>\$i = 1 Do { Echo "i is equal to 1 } While (\$i = 1);</pre>	N/A

	<pre>While (i < 10);</pre>		
Functions			
User-defined function definition (creation)	A block of code designed to perform a particular task. It is executed when it is called on.	A block of statements that can be used repeatedly in a program. It is only executed by a call to the function.	A reusable code block that only needs to be written once.
User-defined function call without arguments	<pre>Function sayHello() { Alert ("Hello!"); }</pre>	<pre>Function sayHello() { Echo ("Hello!"); } //call the function sayHello(); Returns: Hello!</pre>	<pre>def sayHello(): Print "Hello!"</pre>
User-defined function call with arguments	<pre>Var z; Function add (x,y) { Z = x + y }</pre>	<pre>Function lastName(\$fname) { Echo (" \$fname Chan"); } lastName("Leanne"); Returns: Leanne Chan</pre>	<pre>Def lastName(fname) Print fname + "Chan"</pre>
User-defined function return	<pre>Function add (x,y) { Return x + y }</pre>	<pre>Function add(\$x,\$y) { \$x + \$y = \$z; Return \$z; } Echo ("4 + 7 = " . add(4,7)); Returns: 4 + 7 = 11</pre>	<pre>Def lastName(fname) Print fname + "Chan" return;</pre>

```
<?php
if($day <= 1){
    print("STOP and TALK to Ernie :D");
```

```

}else{
    print("GET to WORK on Part II and Part III");
}
?>

```

My Programming Concepts Syntax Guide: Part 2

* PHP Syntax

Useful Internal Functions	JavaScript	PHP	Python
Variable Functions			
Variable Output to screen (* echo)	<pre> <p>Hello </p> <p id="hello"></p> <script> document.getElementById("hel lo").innerHTML; </script> <script> </pre>	<pre> echo("hello"); </pre>	N/A
Variable Output to screen (* print)	<pre> <script> document.write(1+1); </script> </pre>	<pre> print("hello"); </pre>	Print "Hello"
Variable String Length	str.length	strlen()	len()
Variable Is a String	(typeof (variable))	Bool is_string()	isinstance('hello', str)
Variable Find Values in String (* strpos)	str.indexOf()	strpos("hello world", "world")	Pos = helloworld.find(hello)

Variable Returns Part of String (* substr)	Var str = "hello world" Var t = str.substr(6,10)	substr("hello world", 6)	X = "hello world" X[6:]
Variable String make lowercase	str.toLowerCase()	strtolower()	.lower()
Variable String Uppercase first character in each word. (* ucwords)	.toUpperCase()	ucwords()	.title()
MORE String Function			
You Choose String Function 1	.trim()	trim()	.strip()
You Choose String Function 2	Function reverse(s) { Return s.split("").reverse().join(""); }	strrev()	hello[::-1]
You Choose String Function 3	.repeat()	str_repeat("hello",13)	Str = "hello" Print str[:5]*2
You Choose String Function 4	.split()	str_split("hello",1)	.split()
You Choose String Function 5	.substr(str.indexOf(':')+1)	explode()	.split(' ')
Math Functions			
Math Functions Random Number	Math.random()	rand(0,100)	random.randint()
Math Functions Floor	Math.floor()	floor()	math.floor()
Math Functions Ceiling	Math.ceil()	ceil()	math.ceil()
Math Functions Not a Number	isNaN()	is_nan()	math.isnan()
You Choose Math Function 1	math.abs()	abs()	math.fabs()
You Choose Math Function 2	math.log()	log()	math.log()
You Choose Math Function 3	pow(x,y)	pow()	math.pow

Array Functions			
Array Push	<code>.push()</code>	<code>array_push()</code>	<code>list.append()</code>
Array Count Values	<code>.length()</code>	<code>array_count_values()</code>	<code>.count()</code>
Array Sorting	<code>.sort()</code>	<code>sort()</code>	<code>.sort()</code>

My Programming Concepts Syntax Guide: Part 3

And Even More	PHP	NOTES:
Escape Variables		
<code>\'</code> to escape single quote	<code>echo("I\'m a person");</code>	Allows single quote within string
<code>\"</code> to escape double quote	<code>echo("\"Hello \" ");</code>	Allows double quote within string
<code>\\</code> escape backslash	<code>echo(' \\ ');</code>	Allows backslash within string
<code>\\$</code> escape \$	<code>Echo ("\\$500");</code>	Allows dollar sign within string
<code>\n</code> line breaks	<code>Echo ("Line 1 \n Line 2");</code>	Creates a line break in a string
<code>\t</code> tab space	<code>Echo ("\t Line 1")</code>	Creates a tab space in a string
<code>\r</code> carriage return	<code>Echo ("\r ")</code>	Creates a carriage return
Single Quotes, Double Quotes, HEREDOC, and NOWDOC		

'...'	Echo ('This is a string');	Simplest way to specify a string
"..."	Echo ("This is a string");	Allows user to specify string, escape variables are recognized
HEREDOC	<<<EOT EOT;	Delimits strings
NOWDOC	<<<'NOW' NOW;	HEREDOC with no parsing
PHP Functions		
isset()	if (isset(\$_POST["month"])) { \$month = \$_POST["month"]; }	Determines if a variable is set
include_once()	include_once("coinFunction.php5");	Allows separate document to be included once