**Compare and Contrast: PHP and Java**

In a world where technology continues to advance exponentially, knowing how to program is becoming an imperative, and a single language, nor even a single software framework, will not necessarily suffice. Although programming languages have not changed dramatically in the last 15 years (Belton, 2015), the uses for programming are growing innumerable, technology is advancing heavily for “security and reliability verification, […] modularity, and database integration” (“Programming Languages: Consolidation and Growth”, n.d.). Along with this vast growth in technology usage comes a need for programmers to be well-versed in multiple languages. Through exploration of code organization will allow for an in-depth look at the style of programming used in each, discussing object-oriented, scripting, and dynamic versus static typing. The built-in data types to examine include numeric, sequences, text variables, boolean values and map types. Lastly, analyzation of the basic syntax used in PHP and Java will aid in a final look at the main distinctions for each. These distinctions and similarities will arise naturally by comparing and contrasting the ways in which variables are declared, initialized and accessed, researching the lifetimes for variables in each language, and investigating how to utilize basic functionalities such as functions, loops, and conditional statements.

Java and PHP are both object-oriented programming (OOP) languages, meaning that both languages can function by utilizing the concept of objects to progress. Many programming languages utilize this style of programming, which makes it difficult to explicitly explain the specific attributes of OOP because there are few popular languages that are not object-oriented. The main distinction between OOP and other styles is that OOP is “organized around objects rather than ‘actions’ and data rather than logic,” (Rouse, n.d.). Java and PHP are also both imperative, or procedural, languages, meaning that the program written in this style has a specific state which changes throughout the execution and whose “basic unit of abstraction is the PROCEDURE” (Bellaachia, n.d.).

PHP and Java separate themselves from each other when we begin to define PHP as a scripting language. PHP, or any scripting language, is one which can alter the way other programs work. Its purpose is not necessarily to create standalone applications or content, but to adjust other applications to function in a different way. More technically, PHP supports the writing of scripts for internet environments rather than applications.

The most prominent difference between Java and PHP is that Java uses static data typing (“Introduction to Static and Dynamic Typing”, 2004) and PHP has dynamic typing. Steve Ferg (n.d.) gives an excellent distinction between static types and dynamic types:

In Java, all variable names (along with their types) must be explicitly declared. Attempting to assign an object of the wrong type to a variable name triggers a type exception. That’s what it means to say that Java is a *statically* *typed* language.

In [dynamic languages], you never declare anything. An assignment statement binds a name to an object, and the object can be of any type. If a name is assigned to an object of one type, it may later be assigned to an object of a different type. That’s what it means to say that [a language] is a dynamically typed …

More succinctly, this means that all variables in Java must be given a *type* upon declaration, whereas variables in PHP require no such binding. For example, if an integer of value 5 is needed in Java or C, one would write:

int anInteger = 5;

Thus, binding the variable “anInteger” to the data type “int” forever, and although the value may change, the variable can never be given a value of any type other than int. In PHP, however, to create any type of variable one needs only write:

var $aVariableExample = 5

In PHP, as opposed to Java, the value of the variable “aVariableExample” change throughout execution of the program, but the type of variable can be changed as well. If the variable “aVariableExample” was needed later as a string, a programmer need only update the variable to that new value:

aVariableExample = “a string”

This makes PHP a lot easier to write and work with, but also makes it easier to create unintended errors. Where Java is strict and PHP is lenient, Java is stable and PHP is insecure.

Java and PHP both have numerous built-in data types, both include numeric types, text types, map types, and both languages allow the use of a Boolean value, a value which can be set either to True or False. These Boolean data types are invaluable when it comes to conditional code execution and is it is no wonder that they are so straight forward.

PHP and Java both have integer and floating-point data types. For Java, int, by default, is an integer whose minimum and maximum available values (due to memory constraints) are -32,768 and 32,768 respectively (“Java Tutorials: Primitive Data Types”, n.d.). For PHP, an integer is a representation of a decimal, hexadecimal, octal or binary number which which can optionally be preceded by + or - (The PHP Group, “Integers”). A floating-point value, for both PHP and Java, as well as most programming languages, is a representation of a floating point number, a number whose structure resembles that of scientific notation. In Python, a float can be declared either as a positive or negative precise number to a certain decimal point, or as a combination of a precise number to a certain decimal point and an exponent, i.e. 0.314E1 (Tatroe, MacIntyre, Lerdorf, p. 23. In Java a float is almost exactly the same as a double, which is much more common, although a float can only hold 32-bit numbers, whereas double values can hold 64-bit numbers (“Short and Float Variables”, n.d.).

Java and PHP have one major built-in mapping, or in the case of Java, sequential, data type: an array, though they function quite differently. In Java, an array is an immutable, and is given, at instantiation, a size that it must maintain throughout the entirety of its lifetime. It is also bound to a single data type, such as an int, or String, and it is also possible to have 2-dimensional arrays, or an array of arrays, or even a 3-dimensional array, an array of arrays of arrays, and so on and so forth. It is basically a container for objects that is of a fixed size and a single data type (“Java Tutorials: Arrays”, n.d.). However, in PHP, because variables are always dynamically typed, the array does not need to have a pre-defined data type upon instantiation, nor even a data type which it adheres to throughout the entirety of its contents. This means that a single array in PHP may house many different data types within it. Another stark contrast between arrays in Java and arrays in PHP is that PHP arrays may have indices which are defined by the programmer. In Java, the indices of an array are static and are defined automatically as integers in strict numerical order starting from index 0 through index n-1 on an array of n objects. PHP may have an array whose indices are 1, “foo”, 100, “eighty-nine”, etc. The only requirement PHP has regarding the input of its custom indices is the index must be either a string or an integer. Arrays in Java may be viewed as mapped types or sequential types, for the contents of an array are “mapped” to their numeric index, and the contents of an array are examined sequentially. Accessing a specific index in an array in Java requires a sequential examination of each index up until the specified index is reached. In PHP, an array is mapped to its index, but is certainly not necessarily sequential.

Java and PHP are extremely similar in their ways of typing text, although once again PHP proves to be more lenient with its functions. In Java there are both the String data type and the char data type. In Java, a char is a single 16-bit Unicode character (“Java Tutorials: Primitive Data Types”, n.d.). A String, in Java, is most easily described as an array of chars, which is literally how a string is defined in C. Strings are immutable, just as arrays are, although the String’s functionality is much more diverse than an array. Arrays have limited functions, and to adjust sizing or content, the programmer must specify a copy of the array with new dimensions or attributes. Strings, on the other hand, do most of this without the programmer’s knowledge. Changing a String’s content calls an internal function that copies the new content into a new String, destroys the old String data and makes a new variable with the same variable name with the content of the new String.

PHP does not care whether an object is a string or a character, both are declared exactly the same way. Strings are basically lists of characters, and to define a character, you would simply define a string with that one character. Strings can be denoted by either double quotation marks or by single apostrophe punctuation, and this notation has different purposes. As discussed in a previous paper, strings in PHP which are declared using double quotes may be interpolated, meaning that the text which comprises these strings may be expanded.

Another major similarity between PHP and Java is the way in which they differentiate code sections. They both utilize brackets to highlight where sections of code begin or end, such as in a method or function, specifically, both use 1TBS, “the one true brace style” indentation. Comment etiquette and line ending should be covered as well. Java and PHP have different forms of commenting, although they all allow for comments to be written in any part of the code. Comments in Java and PHP can be written as either single, in-line notation, or as blocks of text. Every line of code, with the exception of lines opening up a new section of code (i.e. wherever a bracket is placed), must end in a semicolon (;) in order for the compiler to recognize that it is the end of an instruction. In both languages, the beginning of a single line comment is denoted by two forward slashes (//) and comment blocks are denoted by a forward slash with an asterisk (/\*) to begin, and an asterisk and forward slash to end (\*/).

Java and PHP code is constructed very similarly. Both are compilations of variables and functions. In PHP, a function is called just that, a function. Tutorials Point defines a function as “a piece of code which takes one more input in the form of parameter and does some processing and returns a value” (“PHP - Functions”, n.d.), although this definition implies that all PHP functions must return a value, but functions may simply execute some code and end, just as in Java. Java methods functions are defined similarly; they are sections of code which execute a specific action that can be reused however frequently or infrequently as necessary.

PHP and Java functions are declared much the same way, except the access level is removed. In Java, these functions are called ‘methods’, and they are declared by an access level, which other files have access to them, a return type, a method name (conventionally named in camel caps), and a list of parameters to be passed to the method for use within that method. If parameter(s), such as an integers or strings, needed to be used within the method, the method declaration would be rewritten to include the type of variable it could take and a name for that method to use to refer to the parameter. In PHP the keyword “function” is added to begin the definition of a function, and the return type is removed. If a parameter is needed, it does not need to be preceded by a data type, since PHP uses dynamic typing.

In conclusion, Java and PHP are both powerful languages that are similar in many ways, but in general used for very different purposes. PHP and Java can both be described as verbose and robust languages. Java may also be described as an inflexible and intricate language that excels at large, general applications, both client-side and server-side (“Comparison of Programming Languages”, n.d.). PHP is used for web development as opposed to general applications (as in Java). Java and PHP have their assets and their disadvantages, but all are incredibly practical languages for use in the modern world.

Works Cited

"Arrays." *(The Java™ Tutorials Learning the Java Language Language Basics)*. N.p., n.d. Web. 19 Aug. 2016.

Bellaachia, A. "Imperative Programming Languages (IPL)." *Imperative Programming Languages (IPL)* (n.d.): n. pag. *George Washington University*. Web. 19 Aug. 2016.

"Comparison of Programming Languages." *Wikipedia*. Wikimedia Foundation, n.d. Web. 19 Aug. 2016.

"PHP Functions." *Www.tutorialspoint.com*. N.p., n.d. Web. 19 Aug. 2016.

"PHP: Integers - Manual." *PHP: Integers - Manual*. The PHP Group, n.d. Web. 19 Aug. 2016.

Pillai, Premshree. "Introduction to Static and Dynamic Typing." *RSS*. N.p., 17 June 2004. Web. 19 Aug. 2016.

"Primitive Data Types." *(The Java™ Tutorials Learning the Java Language Language Basics)*. N.p., n.d. Web. 19 Aug. 2016.

"Programming Languages: Consolidation and Growth." *Wikipedia*. Wikimedia Foundation, n.d. Web. 19 Aug. 2016.

"Short and Float Variables." Home and Learn, n.d. Web. 19 Aug. 2016.

"What Is Object-oriented Programming (OOP)? - Definition from WhatIs.com." *SearchSOA*. N.p., n.d. Web. 19 Aug. 2016.