CS330: Programming Language Project (PLP)

Assignment 3: Data types and naming conventions

Understanding how different types of data are represented in your programming language is a key step in being able to use it for bigger projects. Research the naming conventions in your language for variables (i.e., do they have to start with lowercase letters? Can they start with numbers? Symbols? do programmers use underscores, as in "last\_name", or do they use camel case (lastName))? Write a piece of code that creates variable of each of these common data types and follows the naming conventions:

• int

• string

• floating-point number

• boolean

• array/list

• hash/dictionary

Even if your programming language only has one data type, or if it doesn't require that types be declared, you should still be able to create variables that store these types of information (well, maybe not the hash table). If your language doesn't have variables or doesn't differentiate between data types, then find out how it stores information and do that.

In your code, experiment with doing different things with the data types: can you add ints and floats? If you do, is the resulting variable an int (narrowing conversion) or a float (widening conversion)? Can you put different data types in the same array or list? Can one data type be converted to another (int to float, string to int, etc)?

Discussion questions:

1. What are the naming requirements for variables in your language?

1. Must include $ sign before the variable name
   1. i.e : $a = “Hello”;
2. No need to declare type of variable in front of the name, unlike Java
3. Valid variable name starts with a letter or underscore, followed by any number of letters, numbers, or underscores. [Source](https://www.php.net/manual/en/language.variables.basics.php)
4. Characters in the name must be in extended ASCII 228
   1. Valid: $var, $Var, $var1, $vaaaaaaaaaaaar111111111, $var\_1, $\_var, $\_1var, $\_var1\_, $\_var\_\_\_\_, $vár, ect
   2. Invalid: $1var, $111

1. What about naming conventions? Are they enforced by the compiler/interpreter, or are they just standards in the community?

Besides the $ sign (which is similar to Perl), the naming convention for PHP is pretty much the same as the other programming languages. However, it can be a little different from some of the other languages (like Java or Pascal), PHP allows underscores in the name of variables (also similarly to C). These are the rules that enforced by the compiler/interpreter. Despites the fact that a name of a variable can be however long the developers want. It is also common sense and standard that naming should be neat, easy to comprehend and simple.

2. Is your language statically or dynamically typed?

PHP is a dynamically typed language as it does not require the type of the variable to be declared right in front of the name.

3. Strongly typed or weakly typed?

PHP is a weakly typed. For example, I can do: echo "Integer: ", $int, "\n\n"; without any issue even though there is a mix of integer and string in one command. If I do that to the other strongly typed languages such as Python and Java, this will surely and instantly give me an error. For strongly typed languages, I have to convert $int into string before “combining” them with a string.

4. If you put this line (or something similar) in a program and try to print x, what does it do? If it doesn't compile, why? Is there something you can do to make it compile? x = "5" + 6

This will give 11 as the result. The plus sign in PHP automatically change string into numbers.

5. Describe the limitations (or lack thereof) of your programming language as they relate to the coding portion of the assignment (adding ints and floats, storing different types in lists, etc). Are there other restrictions or pitfalls that the documentation mentions that you need to be aware of?

1. In order to print an array or a dictionary, I must do loop through the whole thing to indicate their location (index for array, key for dictionary).

for($i = 0; $i < sizeof($arr); ++$i){

echo "Item of array ", $i, "\t", $arr[$i], "\n\n";

}

foreach($dict as $key=>$val) {

echo "Items of dictionary: ", "$key=>$val", "\n\n";

}

1. While Python I can just simply do: print(arr)
2. Must include $ everytime I use that variable
3. There is not really a dictionary or hashtable in PHP unlike Python or Java. The PHP “dictionary” is a array:

$dict = array("a"=>"1","b"=>"2","c"=>"3");

Where “a”,”b”,”c” are the keys and “1”,”2”,”3” are the value respectively

PHP does not really distinguish between string number (“2”) and an integer (2), 2.00 and 2

Say $a = “2”, $b=2. Then $a+$b will be 4. Despite the fact that these are different type.

Even when I convert 2 into float by using floatval, I still got 2 but not 2.0 or 2.00. In order to get 2.00 I must do echo sprintf("%.2f", $var);

6. How do type conversions work in your language? Are the conversions narrowing or widening, and do they work by default or do they have to be declared by the programmer?

String and int can be treat really interchangeably. I can combining a string and an integer without going through any issue. Same goes for float and integer. Some languages would prohibit when combining different type of variable in one command, that even require users to convert before committing any command. Moreover, in PHP, I do not have to declare the type of variable before assigning any value (unlike Java). Boolean values are also quite interesting to mention. When I print a True value out, it will be 1. While a False value will become, surprisingly, empty. PHP pretty much treats everything into numbers, by default. Which I would say their conversions are widening. Conversion seems useless in PHP.

Make sure that your answers are clear, accurate, and fully-formed: remember that these tutorials are public, and GitHub users don't have the context of the assignment that you do. Explain the reasoning behind the answers as much as possible. If there is no clear-cut answer to a question, explain why not. And cite your sources! You can incorporate code into your tutorial to show examples, but you should also have a file in your repository that is just code, which someone could download and run.