

Python: Variables and Data Types

Variables and Data Types are two important concepts in the Python Programming Language. “Variables” are terms that hold a given piece of data, whether from the user or hard coded in the program. A “Data Type” refers to the category the programmer intends to assign to a particular piece of data. This handout will further explain what each of them are, how they work, and when to use them.

You can navigate to specific sections of this handout by clicking the links below.

[Variable Guidelines/Rules](#): pg. 1

[Data Types](#): pg. 2

[Different Data Types and What They are Used For](#): pg. 2

Variable Guidelines/Rules

The process of creating variables with data is referred to as “assignment.” A variable assignment usually consists of a name, followed by an equal sign (=), and the value assigned by the programmer. The data is not actually assigned to the variable itself, but rather the variable is a reference to the given data. Once assigned, variables can have a wide variety of characteristics. There are a few guidelines that should be followed to produce a variable that is efficient for a program. For example, when naming a variable, it is important to ensure that the name is meaningful for the data to which it is assigned; otherwise, the program might be confusing to another programmer if they need to review the code later. W3Schools provides the following guidelines for variable names:

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age, and AGE are three different variables)

Example:

```
1 |
2 | #variableName = value
3 |
4 | firstNum = 7
5 | secondNum = 4
6 |
7 | product = firstNum * secondNum
```

The code in red font written after the hashtag (line number 2) is called a “comment.” Comments are used to explain the programmer’s intention or the intended function of the line(s) of code. They are completely ignored by the compiler when the code is written and are helpful when a programmer wants to look back at their original code at a later date and understand what they were trying to do. Comments are also useful if another programmer wants to look at the program and understand how it works.

In the example code, there are three variables. Two of them are assigned values directly, and the third is assigned the value of the first two multiplied together. If the programmer decided to print the third variable, Python would display 28.

Data Types

Data Types are very specific. For example, there is a difference between “Seven,” 7, and 7.0, which are the data types **str (String)**, **int (integer)**, and **float (floating-point number)** respectively. Data types are used with variables to let the computer know how to process given data. If the programmer wanted to request a number from the user, they would use either the **int** or **float** data types, but if they wanted a sentence, or another series of characters, then they would use the **str** data type.

Different Data Types and What They Are Used For

A table of different data types is listed below including examples of how they are used. A list of the common data types and their definitions are shown on the next page.

Example	Data Type
x = “Hello World”	str
x = 20	int
x = 20.5	float
x = [“apple”, “banana”, “cherry”]	list
x = (“apple”, “banana”, “cherry”)	tuple
x = range(6)	range
x = {“name” : “John”, “age” : 36}	dict

Figure 1: "Python Data Types." W3Schools, 2021, https://www.w3schools.com/python/python_datatypes.asp

str (String) Data Type is used when the programmer wants the data to be represented as exact words that are either hard coded into the program or gained from the user. A String data type is always enclosed in quotation marks.

int (Integer) Data Type is used when the programmer wants the data to be represented by a whole numerical number.

float Data Type is used similarly to the **int** data type, except instead of being represented as a whole number, the data would be represented by a decimal or floating-point number.

list Data Type is used to represent and organize a series of data, whether it be numbers, letters, or words, and assign them to a single variable. This series of data can be changed later, but they must always be enclosed in brackets [].

tuple Data Type is like the **list** data type in the sense that it stores multiple values in a single variable. Unlike a list, however, once tuples are created, neither the order nor the contents of it can be changed. It is also enclosed in parentheses rather than brackets.

range Data Type is most often used in loops to represent a series of numbers that start with 0 (by default) and increase by one (by default) until it reaches a given number ("Python range() Function").

dict (Dictionary) Data Type is similar to both the **tuple** and **list** data types, except rather than holding one value in each part of it (like 1, 2, 3, 4, 5), dictionaries hold a value pair, one acts as a key, and the other the value associated with it (like animal: cat, name: fluffy). These value pairs are changeable, so the **dict** data type acts like an actual dictionary, except instead of pages, it is enclosed in curly brackets { }.

Example:

```
1 |
2 | #variableName = value
3 | #print (dataType (variableName))
4 |
5 | testVariable = "Hello"
6 | print (str (testVariable))
7 |
8 | testVariable2 = 3.14
9 | print (float (testVariable2))
10 |
```

Not only are variables important when writing code, but declaring the desired variable type is also an important concept to keep in mind when writing it. Doing so specifies how to store or display the value(s) when the program reads it, which makes the whole code easier to understand. The code in the example will print “Hello” and “3.14” respectively, and since “Hello” is a **String**, and 3.14 is a **floating-point** number, the data types **str** and **float** are used.

Works Cited

- “Python – Variable Names.” *W3Schools*, 2021,
https://www.w3schools.com/python/python_variables_names.asp
- “Python range() Function.” *W3Schools*, 2021,
https://www.w3schools.com/python/ref_func_range.asp