**Word Count: 1577**

Section 4,

In this section, we will cover the following topics

* Basics of Python Programming
* Typecasting & loops in Python
* Conditional Branching & Functions in Python
* Modules in Python
* Create a GUI with Python

Video, Basics of Python Programming

In this video, we will learn about the need for a programming language, the difference between a compiled & an interpreted language, and later we compare the text editor with the IDE. Then, we will learn variables in python, and Finally, we will learn the different data types in python with examples

“I think everybody in this country should learn to program a computer,” said Apple’s co-founder Steve Jobs, “because it teaches you how to think.”

Learning to code is one of the most profoundly life-changing things you can do. Code is a critical layer in our lives that sits between us and the digital world. Learning to use code and hardware is incredibly empowering. It enables you to automate a whole bunch of boring and repetitive tasks in your life, freeing you up to concentrate on the important things. The good news is that you don’t need to be a genius to know coding, just like don’t have to be a genius to read and write. It’s actually pretty simple once you learn a few simple concepts. We’ll demystify the whole process of coding in Python, so you can better understand what goes on inside your Raspberry Pi.

At the heart of a Raspberry Pi 4, there are billions of voltage switches called bits that can be set high and low at super fast speeds. This is the language of the machines. It takes a very long time to code in this language by humans. This is why we use a coding language that can convert human-understandable language to machine language.

There are a lot of programming languages out there, and they all offer something special. Python is a great option for beginners. It has an easy to read syntax that’s free from clutter, and you don’t have to worry about curly braces that come in more complicated languages like Java. And it scales all the way up to industrial, medical, and scientific purposes, so it’s ideal for beginners and experts also.

Depending on the way the code is converted to machine language, programming languages are divided into two. Compiled language and Interpreted language. Python is an interpreted language”. That is, you write code and then run the program. Under the hood, it’s translated and runs on the fly. Some languages like C and Java, are compiled. That is you write the program, then compile it to get a machine code, then you run the machine code. To know more about this please check out the resources section.

You don’t need to do anything to set up Python on your Raspberry Pi, as it comes preinstalled with the operating system. Just open the terminal and enter python3. We will use Python 3 as Python 2 will be out of service in 2020. Here you can see the python shell symbol , where you can type out instructions line by line. Using the shell is known as Interactive Mode, as you can interact directly with the code. This is mainly used to debug and test out instructions. Generally, coders create python programs with a regular text editor and save the files with “.py” extension. But as you are a beginner, it's better if you use an IDE aka Integrated Development Environment. IDEs are an all in one solution that combines a text editor, debugger, syntax checker, auto text complete and many such features.

We will use the Thonny Python IDE to learn python on the Raspberry Pi 4. So let’s begin our programming journey with understanding the concept of variables in Python. If you’ve created a science project or experiment, you may have come across variables. In science, a variable is any factor that you can control, change, or measure. In computer programming, variables are used to store things in your program. In Python, you write the name of a variable, then a single equals sign and the word, number, or object you want to put in it. Enter this code directly into the Shell:

a = 1

b = 2

If you ever want to retrieve the contents stored in the variable, you can do this by just typing the variable name like this. You can also print out the variables by passing them into a print function.

Now let’s look at the naming rules for variables.

1. Variable names should only have a combination of lower case, upper case, digits, or an underscore.
2. Don’t start a variable name with a digit. It can start with a letter or an underscore
3. Variable names are case sensitive
4. Do not use reserved keywords as variables

With variables, you define the variable once and then use it in your code. Then it’s ready for changing at any time. This will say a lot of time for the programmers, rather than going to each line of the code and modifying it.

You can assign multiple values to multiple variables in a single line of instruction. Enter the following in the shell

a, b, c = 5, 3.2, “Hello”

Now enter

print(a)

Then print(b)

And later print(c)

We can assign the same value to multiple variables at once also by entering following instruction

x = y = z = 1

Now printing each variable reveals the same output

Python has five standard data types. Numbers, Strings, Lists, Tuple, and Dictionary.

Integer, Floating point numbers, and complex numbers all falls under the Number datatype. Python automatically assigns the correct data type based on the number type. For examples,

Type a = 5 in the shell, followed by type(a) will reveal that it is of integer class. Similarly check the datatype for a = 2.0 and a =1 + 2j.

Variables can also be used to contain ‘strings’. These are groups of letters and other characters that form words, phrases, or other text. Creating a string variable in Python is pretty much the same as creating an integer, except you surround the text with single or double quotes. Using double quotes makes it easier to include apostrophes, such as

print("Don’t worry. Be Happy").

This line would break after ‘Don’ if you used single quotes

print('Don’t worry, Be Happy') – so use double quotes for now.

Next, let's look at lists in Python. A List is an ordered sequence of items. It is one of the most used data types in Python and is very flexible. All the items in a list do not need to be of the same type. Declaring a list is pretty straight forward. Items separated by commas are enclosed within square brackets and assigned to a variable.

For example, type the following

a= [5,10,15,20,25,30,35,”Forty”]

To access an element from this list, let's say to access 15 you would need to enter a[2], as the list is indexed from 0 onwards.

Similar to get a slice of this list, you need to use a slicing operator to extract a range of items. For example to extract the first three elements from the list, you should enter a[0:3]. This will extract elements from index 0 to index 2.

Lists are mutable, meaning, the value of elements of a list can be altered.

For example, if you enter a[0]=”Five” and again access the “a” list you will see that the list was updated with the new entry.

Tuples are just like lists, but they are immutable, that is once a tuple is created, it cannot be modified. Tuples are used to write-protect data. It is defined within parentheses () instead of square brackets, where items are separated by commas like shown here. The same slicing operations of lists are applicable for tuples also.

The final data type of python is named a Dictionary. It is an ordered collection of key-value pairs.

We use the key to retrieve the respective value. But not the other way around. It is used when we have a huge amount of data. Dictionaries are optimized for retrieving data. We must know the key to retrieve the value. In Python, dictionaries are defined within curly braces, with each item being a pair in the form key:value. Key and Values can be of any type.

Now create a dictionary as shown here:

d= {1:”Hai”,”Five”:2}

Now can you guess what will be the output of d[2].

Yes, it will be an error. If you type d[“Five”] it will give 2 as the output.

If you type d[1] it will give “Hai” as the output.

Thus we have finished an overview of different data types in Python

Summary

In this video, we have covered the following

* What is the need for a Programming Language
* Difference between Compiled and Interpreted Language
* Text editor vs. IDE
* Variables in Python
* Data Types in Python

In the next video, we will learn about type casting and loops in python.