**Word Count: 872**

Video, Conditional Branching & Functions in Python

In this video, we will learn Conditional Branching & Functions in Python. Finally, we will create a Simple Calculator Project

We will learn about conditional branching in Python, then we will talk about functions, and later, we will implement a simple calculator project.

Most of the programs are more than just a simple list of commands to be run over and over again. Programs can make decisions and execute some commands only if certain circumstances apply.

They do this using a special Python keyword called “if” and its optional counterpart, “else”. In some cases, you will need extra conditions in between if and else. In that case, we use the keyword “else if”.

To understand this, we will create a simple program that accepts 2 numbers as inputs, and checks whether the first number is greater than, equal to, or less than the second number.

Open the Thonny IDE and load the conditional statement.py file from the downloaded repository. Now run the script. First, I am going to enter x as 5 and y as 7. The output shows “x is less than y”

Let’s test one more condition before moving to the code explanation.

Entering x and y as equal to 5 returns an output: “x is equal to y”

In the code, first, we prompted the user to enter the numbers x and y using the input() function. We then stored the values after typecasting them to an integer.

Next, the keyword “if” followed by the condition “x greater than y” is given. This means that only if the condition is true, any statements inside the “if” will work. Here the output is obviously the string “x greater than y”. As there are multiple conditions involved, we need to use the else if condition in between. In this case, we need to check for the equality condition. Please remember that x equals y and x double equals y are different. While the Former assigns x to y, the later checks whether x and y are the same and return either True or false. Finally, we use the “else” keyword. Note that here we haven’t put any conditions for this. Please remember to put the colon and proper indents for an error-free program.

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reuse. As you already know, Python gives you many built-in functions like print(), etc. but you can also create your own functions. These functions are called user-defined functions.

There are simple rules to define a function in Python.

* Function blocks begin with the keyword def, followed by the function name and parentheses ( ( ) ).
* Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
* The code block within every function starts with a colon (:) and is indented.
* The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

Thus the syntax of a function definition will look like this.

Defining a function only gives it a name, specifies the parameters that are to be included in the function, and structures the blocks of code. Once the basic structure of a function is finalized, you need to call the function inorder to execute it.

Open a new blank script and type the following

def my\_function():

print("Hello from a function")

Now save and run the script. You won’t be getting any output as there is no function call. Thus add

my\_function()

at the end of the code, and rerun it to get the output.

To learn more about functions, please check out the links in the resources section.

A project is always the best way to understand the implementation of concepts. So, now we will create a simple calculator with all the programming concepts we learned till now.

Open the Simple Calculator.py file from the downloaded repository in Thonny IDE and run it. Select the multiply operation by entering 3. Now enter the first number as 30 and second as 2. You should get an output 30 \* 2 = 60. Now let’s check out the code.

For each mathematical operation, we have defined separate functions. Each function accepts two parameters as arguments. They are the first and second numbers. Each function uses the return keyword to give back the result of the operation. Next, we use the print function to print out the options. It then takes in the input from the user. Next, we have to collect the numbers. Here we need to typecast string to an integer so that we can later use the numbers to do mathematical operations. Next, we created a conditional branching for each operation. The condition was checked with a double equal to the operator. In the end we also added an else statement, to prompt the user that the input is invalid.

Summary

In this video, we covered the following topics

* Conditional Branching
* Functions in Python
* A Simple Calculator Project

In the next video, we will learn about modules in python.