**Word Count : 858**

Video, Modules in Python

In this video, we will learn about modules, how to import them, how to work with them, and finally, we will learn to install python packages from raspbian archives and with the “pip” tool.

There is no point in reinventing the wheel. It is true in the world of programming that you can stand on the shoulders of giants by importing other programmer's code seamlessly by using the concept of Modules in Python.

Your programs can import code created by other people using the import statement. This enables you to import modules and use their functions. The only difference is that they are now called methods instead of functions. To use a module in a program, first, you need to import the module to your current program. Furthermore, you need to import all the modules needed for the program first itself in the code.

A common module used in many programs is the math module. This allows you to access lots of math methods, which allows you to do factorial, truncation, power, logarithm, trigonometric operations, angular conversion, and much more. You can go to do a quick google search to see all the available methods of any python module.

First let's try out modules in the shell. In the Shell of Thonny IDE, enter “import math”. You now have access to all the methods in math. You won’t notice any difference, but if you type the “type(math)” it will say that it is a module class. Now let's try to access the methods aka functions of the module. This is done by the dot operator. Type “math” followed by a dot and the name of the function you want to use inside the module. Type “math.sqrt(16)” which will give you 4. Similarly, some methods can take more than one argument. For example, typing “math.pow(64,3)” will return the 64 raised to 3, which is equal to 262144.0

Just like accessing methods from a module, you can also access constants from a module, which are fixed variables contained in the module, by just replacing the function after the dot operator with the constant name. For example, you can type “math.pi” to get the value of pi to 15 decimal places.

Do you feel that typing “math dot then the method name every time is time-consuming, then you can use the “from” keyword to import methods and constants directly to your program. This avoids the need to use “math dot” every time.

Open a new shell and type the following 3 instructions

from math import pi

from math import e

from math import pow

Now, if you type “e” or ‘pi” or “pow(64,3)” you will get the output.

Did you know that you can create your own modules and reuse them in your code? By creating your own modules and importing those created by other people, you can vastly improve the capabilities of your code. If you want to know more about creating your own modules, please visit the links in the resources.

It’s true that with Python, you can build just about anything, from simple scripts to full applications. The Python language, however, doesn’t come pre-installed with all of the fancy features you might want or require in the Raspberry Pi.

The preferred method to install python packages is from the raspbian archives, as it means that the modules you install can be kept up to date easily with the usual “sudo apt update” & “sudo apt full-upgrade” commands. To install a package to your python from the raspbian archives, you can use “apt”

For example, let's install a python package for working with the picamera as an example.

First, you need to update the archive list with “sudo apt update”

Then you type “sudo apt install python3-picamera” To install for python 2 version,

You need to replace “python3” with just “python”.

Not all Python packages are available in the Raspbian archives, and those that are can sometimes be out-of-date. If you can't find a suitable version in the Raspbian archives, you can install packages from the Python Package Index (PyPI). To do so, use the “pip” tool. The “pip” tool comes pre-installed in raspbian except on “lite” version.

Let's install a GUI toolkit package called guizero as an example.

To install guizero package using pip tool type

“sudo pip3 install guizero”

For python 2 version replace “pip3” with just “pip”

You can uninstall python modules with “sudo pip3 uninstall” followed by the module name.

If you have created some useful or cool modules and want to share them with others, you can upload it to the Python Package Index. Please check out the resource section to learn how to do it.

Summary

In this video, we have covered the following

* Why use Modules
* How to work with Python Modules
* Installing python packages using Raspbian archives & the pip tool

In the next video, we will learn to create a graphical user interface with Python.