

1.2 Transcript: Getting started with Hands on programming with Python and beyond

Here you will start hands on programming with Python and its ecosystem. Through the course of this module you are using Anaconda. Anaconda is a package manager, which allows you to install many programs such as Python and Jupyter Notebook at once so that you don't need to install them separately on your computer. Anaconda is also used for installing Math and Science libraries if you need them.

First you must make sure you have installed Python or Anaconda and its associated libraries and can use them in your computing machine. If you have already installed them in your computing machine or have access to them then you may skip this section.

Below you will find a series of pictorial instructions for installing Anaconda.

1. Go to Anaconda website: you can either go to <https://anaconda.org/> and then click on “Download Anaconda” menu or directly go to <https://www.anaconda.com/products/individual> and click “Download” or scroll down to the bottom of its page where you can find Anaconda Installer as shown below:
2. Click on your platform operating system (Windows, MacOS, Linux for 32-Bit or 63-Bit Graphic Installer) to download its related Anaconda. As you can see you can only have Python 3 (version 3.8 here) but not Python 2. Depends on your operating system a dialog box (see figure below for a 64-Bit Windows operating system) will appear to save the associated exe file and execute it afterward.
3. Anaconda setup will then appear (see figure below), which guides you to install and customize Anaconda through a series of dialogue boxes, so that you can simply progress through them by pressing “Next >” button to next stage. This will probably take a while.

4. After successful installation, you are all set to launch Jupyter. To do so you need to launch an application called “Anaconda Navigator” just as usual way that you launch another application in your specific platform. In “Anaconda Navigator” you can see “Jupyter Notebook” (see figure below for a Windows installation) beside many other interesting and helpful tools. In this module you don’t need to learn them but If you wish you can explore these tools by clicking on them and learn about them in your own time.

5. Click on “Jupyter Notebook” and launch it. You will see a Web browser tab or window will appear (see figure below for a Windows installation). Jupyter stands for Julia, Python and R. They are all provided by this interface and if required you will use them. However, the primary focus here is Python. Depends on your default Web browser in your system, this could be Chrome, Mozilla Firefox, Microsoft Edge or any other appropriate web browser. The URL on your web browser should be something similar to: <http://localhost:8888/tree>

6. You are now ready to start your first Python programming.

7. To start your Python programming, first create a new folder of your choice. You may create this new folder under Desktop in your system and call it “PythonTutorial” in anyway you wish depends on your platform. If you go back to “Jupyter Notebook” interface and click on your Desktop there, you should see your newly created folder as it is at the top right there (see figure below for a Windows installation).

8. Click “Python Tutorial” in “Jupyter Notebook” interface. This directs you to new folder location in “Jupyter Notebook” (see figure below for a Windows installation).

9. Ok, let's create your first program. In the Jupyter notebook interface click “New” at the top right-hand corner above file location and then click python 3. This might take a short time, but a new notebook tab will be created for you. Change the name of this notebook from “Untitled” (next to Jupyter symbol) to let say “hello world” by clicking on word “Untitled” and change it to “hello world” and press “Rename” button (see figure below for a Windows installation). You can verify that a new file called “hello world.ipynb” was created under “PythonTutorial” folder in your system. Please note that letters nb are added to ipy extension to emphasis on algebraic notebook.

10. Cells: The box you see in the heart of above figure is called a Cell. Each cell represents a place holder that you can type a set of code so you can type any Python code in that box. For example type: `print(` . Note that Jupyter notebook interface will automatically add the closed bracket and make it like `print()` . Inside parentheses type a double quote and the notebook will add the second double quote for you. Type string `Hello World` between double quotes so that your first code should be
`print("Hello World")`
Run this code by pressing “Run” button at the top and it executes all the code within the cell. You see that the string `Hello World` is printed. Congratulations you have now developed your first Python program. Please get note of colour of words in the cell box and appearance of a new cell box.

In programming `Hello World` is called *string* which is basically a set of characters usually surrounded by double quotes. In Python you can also use single quotes interchangeably with double quotes and anything between them treated as string even if it is a number.

11. As an exercise you can try to print `Hello World` using single quotes. Please note that `print` is a keyword code in Python and it prints whatever is in the parentheses, for example as another exercise you can type a number say 12 by typing `print(12)` or typing `print("12")` in one cell. You may notice that both produce the same outputs but they are different from your computer point of view as you may note that the colour differences between `"12"` and `12` in `print` statement. What do you think about the differences between them and especially their memory usages? Yes you are right, `"12"` is considered as a string while 12 as number or more precisely an integer. Integers require much less memory space than strings in a computer. Assuming that this is your first Python codes, if you don't type in the exact set of characters that you see on the screen it might not run correctly and produce errors. For example, if you use square brackets instead of parentheses you will get an error as shown in figure below. Jupyter notebook provide this facility for you so that you can go back to the same cell, correct the error and execute it again. Can you try this?

12. As it is clear, you can directly interact with the code. For example, this is handy for doing math's. You may think multiplying two number such as 1920 and 1080 using following code:

```
print (1920 * 1080)
```

however, this can done directly by inserting numbers only without any code:

```
1920 * 1080
```

Or you can add (concatenate) two words (strings) together and create a new one: by adding them directly or or assigning each to a variable and then add them together:

```
"hello"+"world"      This prints 'helloworld'
word1 = "Hello "
word2 = "World"
print(word1+word2)   This prints 'Hello World'
```

You may realize that there no need to use quote or double quote for word1 and word2 inside print parentheses in above example as they are both type strings with some values. When you are testing above code don't forget to include the space after 'Hello '. You may also able to mix same type values and initialized variables:

```
word1+"world"      This prints 'Hello world'
```

13. You can determine the type of your data in your program by using `type ()` statement. As you tried following statements `print (12)` and `print ("12")` in two cells and answered a question about type of print arguments, you can now verify your answer by following statements:

```
type (12)
type ("12")
```

in the same cell.

14. Working with Jupyter Notebook is quite strict forward and intuitive. If you require new Cell, from menu you can see "Insert" item. Click and you see two choice i.e., "Insert Cell Above" (or press "A" key in your keyboard) and "Insert Cell Below" (or press "B" key in your keyboard), so you can determine the location of new Cell.

If you want to remove a Cell, you have to select it first by clicking on it and then go to "Edit" and press "Delete Cell", or press D Key twice in your keyboard.

If you go back to previous tab (i.e, Jupyter Notebook interface) you can see list of files you have created so far. You can also use your system to view "hello world.ipynb" file under "PythonTutorial" folder and to see how this file look like. Please note that codes in this file is not Python. What is it then? You may perform a google search about it. Please don not edit this file directly.

However if you want to access your Pyton file, click "File" then "download as", which provide many options such as "AsciiDoc (.asciidoc) , HTML (.html) , ... Python (.py) ...". click on "Python (.py)" a subdiagouge at the bottom of your web browser will appear, which ask you to "open, save or cancel". You can click "save ^", then "save as" (to chose a location). As you can see this file was saved as "hello world.py"