

Lab Work #12

Aim: → Study of use of Jupyter Notebook & open CV.

Objective → The objective of this lab work is.

- 1] Students will be able to list use of Jupyter notebook.
- 2] Configure Jupyter notebook & execute simple functions.
- 3] Students will be able to list use of OpenCV.

Outcomes → After completing the Lab Work #12 student able to

- 1] Configure & use Jupyter notebook.
- 2] Understand use of OpenCV and its applications.

Pre-Requisites: →

- 1] Basic syntax of python.

Theory: →

What is Jupyter Notebook?

The Jupyter Notebook is an incredibly powerful tool for interactively developing & presenting data science projects. This article will walk you through how to use Jupyter Notebooks for data science projects & how to set it up to on your local machine.

First, though: What is a "notebook"?

A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. In other words, it's a single document where you can run code, display the output and also add explanations, formulas, charts & make your work more transparent, understandable, repeatable & shareable. Using Notebooks is now a major part of the data science workflow at companies across the globe. If your goal is to work with data, using a Notebook will speed up your workflow & make it easier to communicate & share your results.

Best of all, as part of the open source project Jupyter, Jupyter Notebooks are completely free. You can download the software on its own, or as part of the Anaconda data science toolkit.

Although it is possible to use many different programming languages in Jupyter Notebooks, this article will focus on python as it is the most common use case.

Installation

The easiest way for a beginner to get started with Jupyter Notebook is by installing [Anaconda].

Anaconda is the most widely used python distribution for data science and comes pre-loaded with all the most popular libraries & tools.

Some of the biggest python libraries included in Anaconda include Numpy, pandas & Matplotlib though the full list is exhaustive.

***** Printers Nanded

Anaconda thus lets us hit the ground running with a fully stocked data science workshop without the hassle of managing countless installations or worrying about dependencies and OS-specific (read: Windows-specific) installation issues.

To get Anaconda, simply:

- ① Download the latest version of Anaconda for python 3.8
- ② Install Anaconda by following instructions on download page & lot in the executable.

If you are a more advanced user with python already installed & prefer to manage your packages manually, you can just use PIP:-

PIP install Jupyter.

Open CV

Open cv was started in Intel in 1999 by Gary Bradsky & the first release came out in 2000. Vadim Pisarevsky joined Gary ~~DARPA~~ Bradsky to manage Intel's Russian software Open CV team. In 2005, Open CV was used on Stanley, the vehicle that won the 2005 DARPA Grand challenge. Later its active development continued under the support of Willow Garage with Gary Bradsky and Vadim Pisarevsky leading the project. Open CV now supports a multitude of algorithms related to computer vision & Machine Learning and is expanding day by day.

Open CV supports a wide variety of programming languages such as C++, Python, Java etc and is available on different platforms including windows, linux, os x, Android & ios. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development.

Open CV - Python is the Python API for open CV, combining the best qualities of the open CV C++ API & the Python language.

OpenCV-Python

OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

Python is a general purpose programming language started by Guido Van Rossum that became very popular very quickly mainly because of its simplicity & code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability.

Compared to languages like C/C++, Python is slower. That said, Python can be easily extended with C/C++, which allows to write computationally intensive code in C/C++ & create Python wrappers that can be used as Python modules. This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) & second, it is easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.

OpenCV-python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy & Matplotlib.



Welcome To Colaboratory

File Edit View Insert Runtime Tools Help [Cannot save changes](#)

+ Code + Text Copy to Drive

✓ [1] 45+60

0s

105

✓ [2] 5*(12.7)/7

1s

9.071428571428571

✓ name= input("Enter your name:")
print("Hello",name)

9s

Enter your name:varad
Hello varad

✓ [4] def add(a,b):
 return a+b
val= add(28,29)
val

0s

57



```
In [ ]: -MIT
        * Marathwada Institue of Technology
        * Maharashtra Institue of Technology
        * Poly
        -India
        * Pune
        * Mumbai
        * Aurangabad
```

```
In [ ]:
```

 Run Markdown Download GitHub Binder

Memory:

-MIT * Marathwada Institue of Technology * Maharashtra Institue of Technology * Poly -India * Pune * Mumbai * Aurangabad

First Heading
Second Heading
Third Heading
Fourth Heading
*Fifth Heading*
*Sixth Heading*

First Heading

Second Heading

Third Heading

Fourth Heading

Fifth Heading

Sixth Heading

ID	Name	Class	Marks
--	----	-----	-----
01	Ram	SY	80
02	varad	SY	45

In []:

ID	Name	Class	Marks
01	Ram	SY	80
02	varad	SY	45


```
In [7]: from ipywidgets import widgets
lb1 = widgets.Label('Enter the First number')
display(lb1)
text1 = widgets.Text()
display(text1)
lb2 = widgets.Label('Enter the Second number')
display(lb2)
text2 = widgets.Text()
display(text2)
btn = widgets.Button(description = "add")
display(btn)
lb3 = widgets.Label()
display(lb3)
def add(x):
    a = int(text1.value)
    b = int(text2.value)
    lb3.value = 'result='+str(a+b)
btn.on_click(add)
```

Enter the First number

Enter the Second number

result=57

[Visit repo](#)[Copy Binder link](#)

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3



Code

Download



GitHub



Binder

Memory: 134.5 MB / 2 GB

```
In [5]: from ipywidgets import widgets
lb1 = widgets.Label('Enter the First number')
display(lb1)
text1 = widgets.Text()
display(text1)
lb2 = widgets.Label('Enter the Second number')
display(lb2)
text2 = widgets.Text()
display(text2)
btn = widgets.Button(description = "Multiply")
display(btn)
lb3 = widgets.Label()
display(lb3)
def Mult(x):
    a = int(text1.value)
    b = int(text2.value)
    lb3.value = 'result='+str(a*b)
btn.on_click(Mult)
btn = widgets.Button(description = "add")
display(btn)
lb3 = widgets.Label()
display(lb3)
def add(x):
    a = int(text1.value)
    b = int(text2.value)
    lb3.value = 'result='+str(a+b)
btn.on_click(add)
btn = widgets.Button(description = "divide")
display(btn)
lb3 = widgets.Label()
display(lb3)
def div(x):
    a = int(text1.value)
    b = int(text2.value)
    lb3.value = 'result='+str(a/b)
btn.on_click(div)
btn = widgets.Button(description = "Subtract")
```

Battery status: 38% remaining



Code



Download



```
btn.on_click(div)
btn = widgets.Button(description = "Subtract")
display(btn)
lb3 = widgets.Label()
display(lb3)
def sub(x):
    a = int(text1.value)
    b = int(text2.value)
    lb3.value = 'result='+str(a-b)
btn.on_click(sub)
btn = widgets.Button(description = "Square")
display(btn)
lb3 = widgets.Label()
display(lb3)
def sqr(x):
    a = int(text1.value)
    lb3.value = 'result='+str(a*a)
btn.on_click(sqr)
```

```
lb3 = widgets.Label()
display(lb3)
def sqr(x):
    a = int(text1.value)
    lb3.value = 'result='+str(a*a)
btn.on_click(sqr)
```

Enter the First number

Enter the Second number

Multiply

add

divide

Subtract

Square

result=9

In [7]: `from ipywidgets import widgets`