

Python Programming for Data Analysis

Edward Salakpi Ph.D.
Wise Osagie

Course Outline

Week One

- Overview of Python Programming Language
- Setting Up Python and Python Environment on PC (Windows/Mac/Linux)
- Using Integrated Development Environment (IDE)
- Variables, Data types and Booleans, Strings
Lists, Tuple, Dictionaries, Sets
- Prints and String Formatting
- Loops, Conditional Statements

Week Two

- Python Functions, and Lambda**
- Object Oriented Programming (Intro)
- Introduction to python Modules (Packages)
- Working NumPy and SciPy Module
- Data Processing with Pandas Module
 - Data Visualization with Pandas
 - Data Visualization with Matplotlib and Seaborn Modules
 - Basic Statistics in python

Python Programming Language

What is it:

A high-level, easy to read and understand, **general-purpose** programming language. Python is a popular choice for both beginners and experienced programmers, making it a versatile tool for various applications.

Key Features:

- **Readability:** Clear and concise syntax and make code easy read and maintainable.
- **Versatility:** can be used for web/software development and data science to machine learning and scientific computing.
- **Interpreted Language:** Uses an interpreter, instead of a compilation. Enables faster development and prototyping
- **Dynamic Typing:** No need to explicitly declare data types, saving time and effort.
- **Garbage Collection:** Automatically manages memory allocation and deallocation,
- **Object-Oriented Programming (OOP):** Supports OOP principles, enabling modular programming, and reusable code.



Python Programming Language

Applications of Python:

Python is suitable for a wide range of applications, including:

- Web Development: Using frameworks like Django and Flask.
- **Data Science:** Using Python libraries like NumPy, Pandas, and Matplotlib.
- **Machine Learning:** Developing machine learning and AI frameworks, using TensorFlow and scikit-learn.
- Scientific Computing: With numerical libraries like NumPy and SciPy for scientific calculations and data analysis.
- Scripting and Automation: Automating tasks, data pipelines, integrating systems.



General Concepts of Programming

Some general concepts that are common to most programming languages:

- **Variables:** These are named storage locations that hold data, like numbers, text, or even lists of other data. Think of them as containers with labels that you can use to store and retrieve information later in your program.
- **Data types:** These define the kind of data that can be stored in a variable, such as integers, floats, strings, or booleans (true/false). Imagine different types of containers, like bowls for fruits, cups for liquids, and boxes for books, each holding a specific type of data.
- **Operators:** These are symbols that perform operations on data, such as addition, subtraction, comparison, or logical operations. They're like the tools you use in your recipe, like mixing, chopping, or baking, to manipulate the data in your program.



General Concepts of Programming

- **Control flow:** This determines the order in which instructions are executed. It allows you to make your program follow different paths based on conditions or user input.
- **Functions:** These are reusable blocks of code that perform specific tasks. You can call them from different parts of your program to avoid repeating the same code.
- **Loops:** These allow you to repeat a block of code multiple times, as long as a certain condition is met. Think of it like a loop in your recipe where you knead the dough repeatedly until it reaches the desired texture.

Setting Up Python Environment on PC with Anaconda

- Windows/Mac/Linux

- **Anaconda** is a popular open-source distribution of Python. It enables the easy to installation of Python and Python packages.
- Anaconda provides a convenient environment management system:
 - **Anaconda Navigator:** a Graphical User Interface (GUI) and
 - **Conda:** a Command Line Tool (CLI),that allows you to create and switch between different Python environments.
- Python environments are used to isolate different versions of Python and packages from each other.



Setting Up Python Environment on PC with Anaconda

- Windows/Mac/Linux

Step 1: Download and Install Anaconda

<https://www.anaconda.com/download>

Step 2: Create a Python Environment

With Anaconda Navigator (GUI)

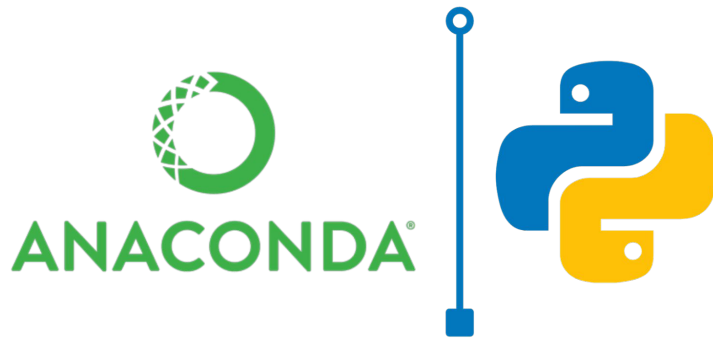
or

Open a terminal window (Windows) or command prompt (macOS and Linux) and run the following command:

```
conda create -n myenv python=3.8
```

↖
Your preferred name

↖
Your preferred python version



Setting Up Python Environment on PC with Anaconda

- Windows/Mac/Linux

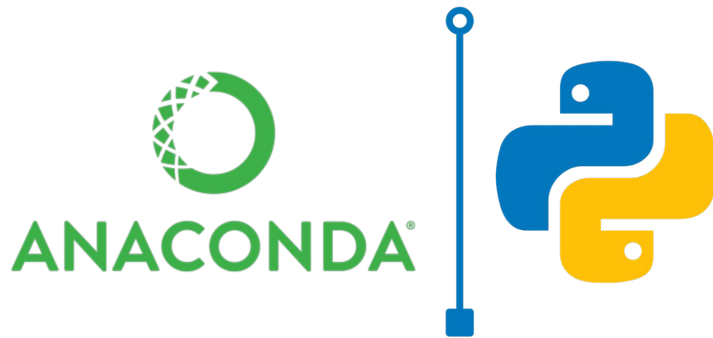
Step 3: Select or Activate the Environment

To use the Python environment, you need select it from the GUI or to activate it with conda .

```
conda activate myenv
```

You can deactivate the environment by running:

```
conda deactivate
```



Step 4: Install Packages

To install a package, search and install with GUI or run the following command:

```
conda install package_name
```

E.g. `conda install pandas`

Or install multiple packages

E.g. `conda install numpy pandas matplotlib`

Applications on base (root)

Channels



Datalore

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

Launch



IBM Watson Studio Cloud

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

Launch



JupyterLab

3.2.1

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch



Jupyter Notebook

6.4.5

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch



Qt Console

5.1.1

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical celltips, and more.

Launch



Spyder

5.1.5

Scientific PYTHON Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Launch



VS Code

1.80.2

Streamlined code editor with support for development operations like debugging, task running and version control.

Launch



Glueviz

1.0.0

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install



Orange 3

3.32.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install



PyCharm Professional

A full-fledged IDE by JetBrains for both Scientific and Web python development. Supports HTML, JS, and SQL.

Install



RStudio

1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R

Anaconda Notebooks

Cloud notebooks with hundreds of packages ready to code.

Learn More

A Full Python IDE directly from the browser

Documentation

Anaconda Blog



Setting Up PC (Working Environment)

The screenshot shows the Anaconda Navigator desktop application. The interface includes a sidebar on the left with navigation options: Home, Environments, Learning, and Community. The main area is divided into two panes. The left pane shows a list of environments: base (root), biopy, data_procs, and ds_env. The right pane shows a list of installed packages with columns for Name, Description, and Version. Four blue callout boxes with white text and arrows point to specific elements: Step 1 points to the 'Environments' button in the sidebar; Step 2 points to the 'Create' button in the bottom toolbar; Step 3 points to the 'base (root)' environment in the list; and Step 4 points to the 'Update index...' button in the top right of the package list pane.

Step 1

Step 2

Step 3

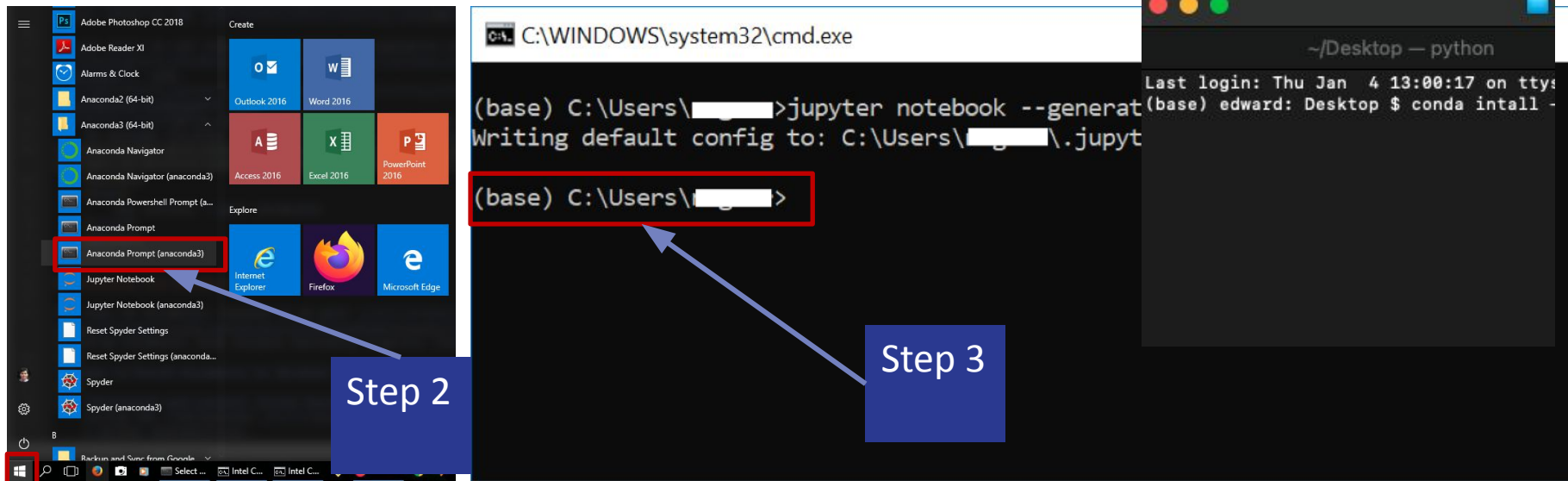
Step 4

Name	Description	Version
✓ _ipyw_jlab_nb_ext...	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
✓ alabaster	Configurable, python 2+3 compatible sphinx theme.	0.7.12
✓ anaconda	Simplifies package management and deployment of anaconda	2021.11
✓ anaconda-client	Anaconda cloud command line client library	1.9.0
✓ anaconda-project	Tool for encapsulating, running, and reproducing data science projects	0.10.1
✓ anyio	High level compatibility layer for multiple asynchronous event loop implementations on python	2.2.0
✓ appdirs	A small python module for determining appropriate platform-specific dirs.	1.4.4
✓ applaunchservices	Simple package for registering an app with apple launch services to handle uti and url	0.2.1
✓ appnope	Disable app nap on os x 10.9	0.1.2
✓ appscript	Control applescriptable applications from python.	1.1.2
✓ argh	The natural cli.	0.26.2
✓ argon2-cffi	The secure argon2 password hashing algorithm.	20.1.0
✓ arrow	Better dates & times for python	0.13.1
✓ asn1crypto	Python asn.1 library with a focus on performance and a pythonic api	1.4.0
✓ astroid	A abstract syntax tree for python with inference support.	2.6.6
✓ astroquery	Community-developed python library for astronomy	4.3.1
✓ async-generator		1.10
✓ async_generator	Async generators and context managers for python 3.5+	1.10
✓ atomicwrites	Atomic file writes	1.4.0

392 packages available

Setting Up PC (Working Environment)

- Create Python environment and install required packages (Terminal , Conda)



At step 3 the “Conda” the command line tool for anaconda can be used to create and manage python environment.

Link to “Conda” documentation: <https://docs.conda.io/en/latest/>

Link to “Conda” user guide: <https://tinyurl.com/conda-user>

Setting Up PC (Working Environment)

- Create Python environment and install required packages (Terminal , Conda)

QUICK START	
Tip: It is recommended to create a new environment for any new project or workflow.	
verify conda install and check version	conda info
update conda in base environment	conda update -n base conda
install latest anaconda distribution (see release notes)	conda install anaconda=2022.05
create a new environment (tip: name environment descriptively)	conda create --name ENVNAME ← Step 1
activate environment (do this before installing packages)	conda activate ENVNAME ← Step 2

<https://tinyurl.com/conda-cheat>

Setting Up PC (Working Environment)

- Create Python environment and install required packages (Terminal , Conda)

CHANNELS AND PACKAGES

Tip: Package dependencies and platform specifics are automatically resolved when using conda.

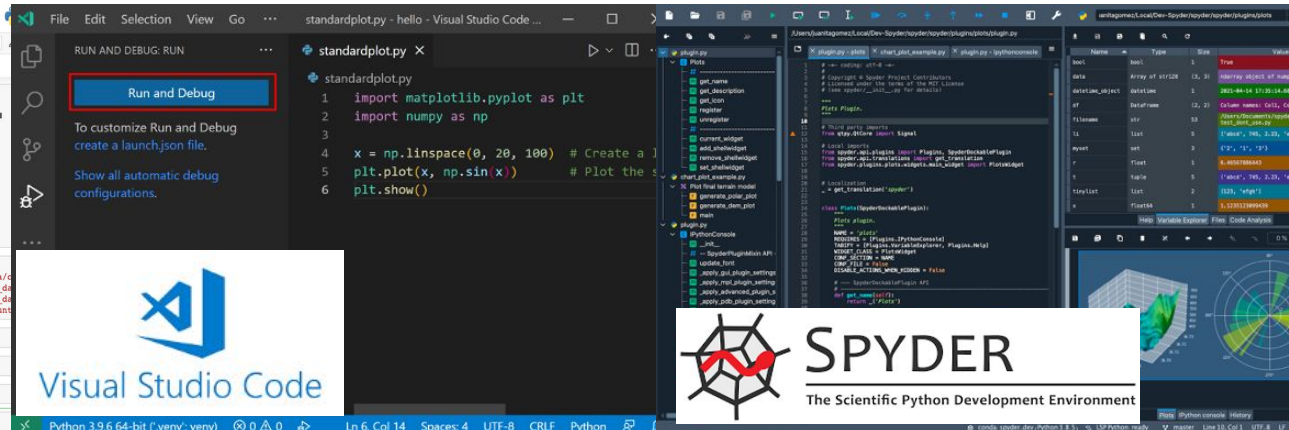
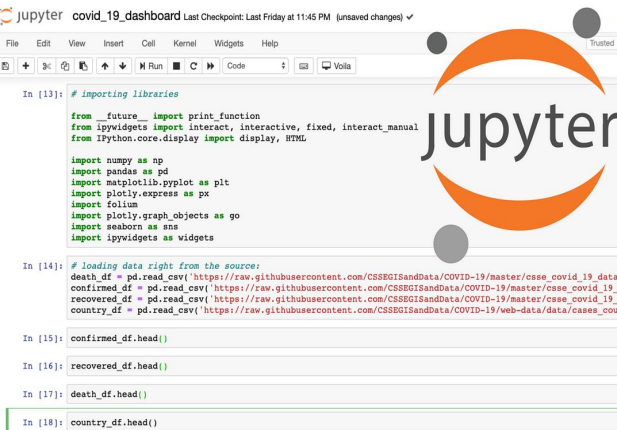
list installed packages	<code>conda list</code>
list installed packages with source info	<code>conda list --show-channel-urls</code>
update all packages	<code>conda update --all</code>
install a package from specific channel	<code>conda install -c CHANNELNAME PKG1 PKG2</code> ←
install specific version of package	<code>conda install PKGNAME=3.1.4</code>
install a package from specific channel	<code>conda install CHANNELNAME::PKGNAME</code>
install package with AND logic	<code>conda install "PKGNAME>2.5,<3.2"</code>
install package with OR logic	<code>conda install "PKGNAME [version='2.5 3.2']"</code>
uninstall package	<code>conda uninstall PKGNAME</code>

Step 3

<https://tinyurl.com/conda-cheat>

Integrated Development Environments (IDE)

- An Integrated Development Environment (IDE) is a software application designed to make programming easier and more efficient.
- It combines essential tools for writing, editing, running, debugging, and managing Python code in a single interface.
- Popular Python IDEs > PyCharm, Visual Studio Code, Spyder, Jupyter Notebook and JupyterLab



Integrated Development Environments (IDE)

Popular Python IDEs:

- **PyCharm:** Powerful and feature-rich, excellent for professional development.
 - **Visual Studio Code:** Versatile and customizable, widely used for various languages.
 - **Spyder:** Scientific Python Development Environment, designed for data science and scientific computing.
- **Jupyter Notebook and JupyterLab:** *An interactive computing IDE, especially useful for data analysis, visualization, and machine learning. But here's a quick breakdown of their key differences*

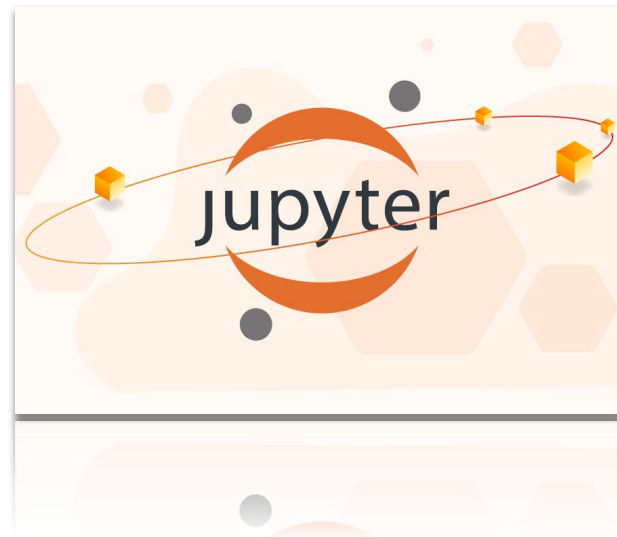
Jupyter IDE

Jupyter Notebook:

- Simple, Lightweight and focused on **python notebooks**
- Good for beginners and quick exploration
- Can feel limited for complex workflows

JupyterLab:

- Modern and flexible interface ,offers a workspace with tabs for notebooks, terminals, text editors
- More feature-rich and customizable, extensible with plugins for additional functionality and workflows.
- Ideal for advanced users and complex projects



Loading Jupyter Notebook IDE

Via Navigator

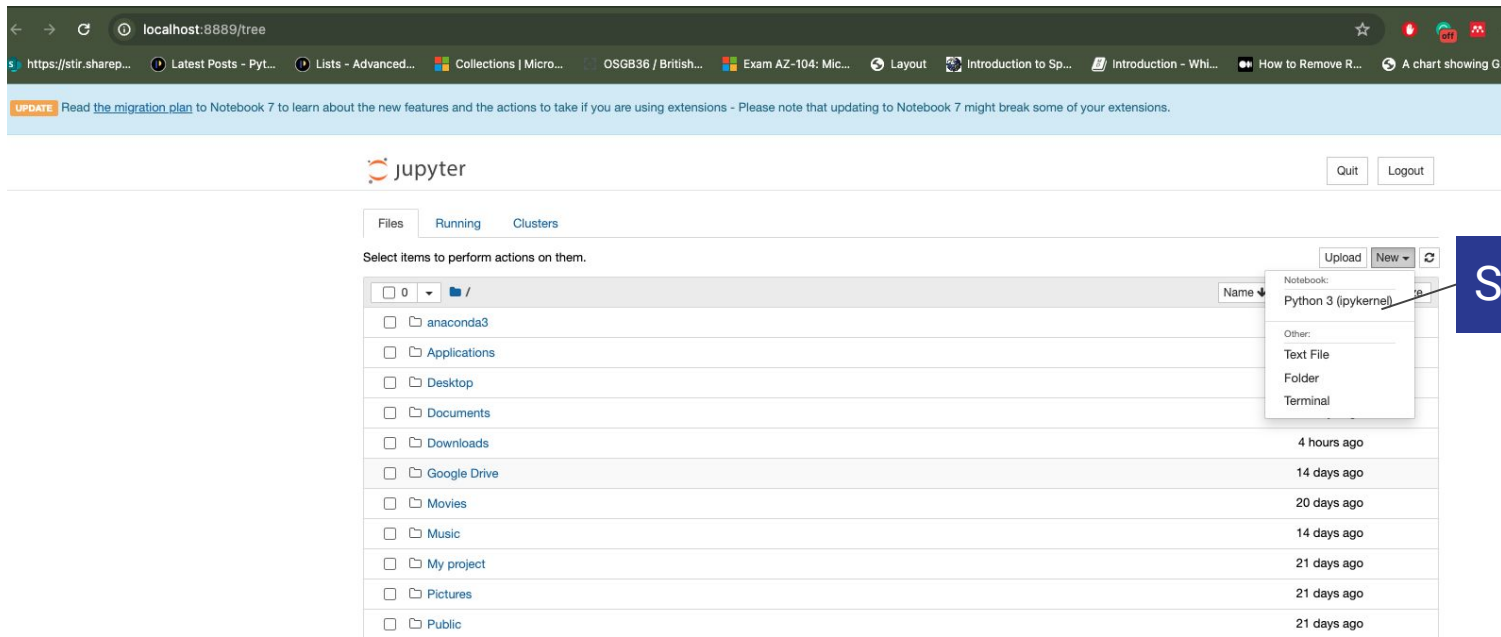
Click to start IDE

Via Command Line

```
C:\WINDOWS\system32\cmd.exe
(base) C:\Users\>jupyter notebook --generate-config
Writing default config to: C:\Users\>
(base) C:\Users\>
```

```
edward ~-zsh- 80x24
Last login: Sun Jan 14 18:55:55 on ttys004
(base) edward: ~ $ jupyter-lab
```

Loading Jupyter Notebook



The screenshot shows the Jupyter Notebook web interface in a browser. The address bar shows 'localhost:8889/tree'. The interface includes a top navigation bar with the Jupyter logo and 'Quit' and 'Logout' buttons. Below this is a tabbed interface with 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file browser. A message at the top says: 'UPDATE: Read the [migration plan](#) to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.' The file browser shows a list of files and folders. A dropdown menu is open, showing options: 'Notebook: Python 3 (ipykernel)', 'Other: Text File', 'Folder', and 'Terminal'. An arrow points from the 'Select Kernel' text to the 'Python 3 (ipykernel)' option.

localhost:8889/tree

https://stir.sharep... Latest Posts - PyL... Lists - Advanced... Collections | Micro... OSGB36 / British... Exam AZ-104: Mic... Layout Introduction to Sp... Introduction - Whi... How to Remove R... A chart showing G...

UPDATE: Read the [migration plan](#) to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

0 /

Name	Modified
anaconda3	
Applications	
Desktop	
Documents	
Downloads	
Google Drive	
Movies	
Music	
My project	
Pictures	
Public	

4 hours ago

14 days ago

20 days ago

14 days ago

21 days ago

21 days ago

21 days ago

New

Notebook: Python 3 (ipykernel)

Other: Text File Folder Terminal

Select Kernel

Loading Jupyter Notebook

localhost:8889/notebooks/Untitled.ipynb?kernel_name=python3

UPDATE Read [the migration plan](#) to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

jupyter Untitled Last Checkpoint: a minute ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [1]: `print('Hello DSIS Class')`

Hello DSIS Class

In []:

Jupyter Lab (Advanced Notebook)

localhost:8888/lab/tree/Desktop/OSiri-DS/Docs

https://str.sharep... Latest Posts - Pyt... Lists - Advanced... Collections | Micro... OSGB36 / British... Exam AZ-104: Mic... Layout Introduction to Sp... Introduction - Whi... How to Remove R... A chart showing G... All Bookmarks

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ ... / OSiri-DS / Docs /

Name	Last Modified
Osiri University...	7 days ago
Python Program...	3 days ago
Python Program...	3 days ago
Python Program...	3 days ago
Python Program...	3 days ago

Launcher

Desktop/OSiri-DS/Docs

Notebook

Python 3 (ipykernel)

Console

Python 3 (ipykernel)

Other

Terminal Text File Markdown File Python File Show Contextual Help

Simple 1 5

Launcher 1

Jupyter Lab (Advanced Notebook)

