

# Python basics

# Python is a script language

A scripting language or script language is a **programming language** for a **runtime system** that automates the **execution** of tasks that would otherwise be performed individually by a human operator.

Scripting languages are usually **interpreted** at **runtime** rather than **compiled**.

# setting up conda environment

- <https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>
- connect to [adroit.princeton.edu](https://adroit.princeton.edu)

```
$module load anaconda3/2021.11
```

```
$conda create --name usrp
```

```
$conda activate usrp
```

```
$conda install jupyterlab pip matplotlib numpy astropy
```

# python/ipython/jupyter-notebook

- <https://jakevdp.github.io/WhirlwindTourOfPython/01-how-to-run-python-code.html>
- The most basic way to execute Python code is line by line within the **Python interpreter**.
- An alternative interpreter called **IPython (for Interactive Python)** is bundled with the Anaconda distribution, and includes a host of convenient enhancements to the basic Python interpreter.
- for more complicated programs it is more convenient to save code to file, and execute it all at once.
- A useful hybrid of the interactive terminal and the self-contained script is the **Jupyter notebook**, a document format that allows executable code, formatted text, graphics, and even interactive features to be combined into a single document.


# Whirlwind Tour of Python

- highly-recommended python tutorial
- clone the GitHub repository at <https://github.com/jakevdp/WhirlwindTourOfPython>
- or open it on Google colab
  - <https://colab.research.google.com/github/jakevdp/WhirlwindTourOfPython/blob/master/Index.ipynb>

SAO/NASA ADS    게임 천문학자    job ▾    folders ▾    grants ▾    conferences ▾    documentation ▾    CCA ▾    green card ▾    git-repos ▾    Jeanne ▾    Calendar    Chang-Goo Kim    NASA HECC portal    Astro-Coffee@Princeton    mews    >>

Dashboard - MyAdroit - Adroit cluster

MyAdroit - Adroit cluster    Files ▾    Jobs ▾    Clusters ▾    Interactive Apps ▾    ? ▾    User Icon    Share Icon

 **PRINCETON**  
UNIVERSITY

Desktops

- Desktop (Visualization Node)    interface to Adroit cluster
- Desktop (Compute Node)

GUIs

- MATLAB
- Mathematica
- XStata

Servers

- Jupyter
- Jupyter for Classes
- Jupyter on Adroit Vis
- RStudio Server

powered by  
**OPEN OnDemand**

OnDemand version: v2.0.20

Display a menu for "https://myadroit.princeton.edu/pun/sys/dashboard/batch\_connect/sys/jupyter-vis/session\_contexts/new"

## Interactive Apps

## Desktops

Desktop  
(Visualization Node)

Desktop (Compute Node)

## GUIs

 Mathematica XStata

## Servers

 Jupyter Jupyter for Classes

Jupyter on Adroit  
Vis

 RStudio Server

## Jupyter on Adroit Vis

This app will launch a **Jupyter** server using **Python** on the **Adroit cluster** visualization node.

For visualization and light interactive work that might require internet, Adroit-vis is a shared login node with limited resources and therefore please use regular [Jupyter session on compute nodes](#) for larger runs.

☒ Use JupyterLab instead of Jupyter Notebook?

JupyterLab is the next generation of Jupyter, and is completely compatible with existing Jupyter Notebooks.

Number of hours

6

- 2021.5
- ✓ 2021.11
- 2020.7
- 2020.2
- 2020.11
- 2019.3
- 2019.10
- 2018.12
- custom

Use your conda env usrp



SAO/NASA ADS    게임 천문학자    job    folders    grants    conferences    documentation    CCA    green card    git-repos    Jeanne    Calendar    Chang-Goo Kim    NASA HECC portal    Astro-Coffee@Princeton    mews    >>

My Interactive Sessions - MyAdroit - Adroit cluster    python-intro... (2) - JupyterLab

File   Edit   View   Run   Kernel   Tabs   Settings   Help

+   +   +   ↺

Filter files by name

/ usrp-sciprog / day2 /

Name	Last Modified
exercises	2 days ago
arguments...	2 days ago
hello.py	2 days ago
numpy-int...	2 days ago
python-int...	2 days ago
readme.md	2 days ago

Launcher    python-intro.ipynb

Python 3 (ipykernel)

Python intro

In this section, we are going to see the basic blocs that structure how we communicate in python and how we can write complex codes in a way that python will be able execute AND that we will be able to understand. This includes how to declare variable, write functions and use built-in commands that allow us to write algorithms in python ('for' loops, 'if' statements, and more).

Select Kernel

Select kernel for: "python-intro.ipynb"

usrp [~/conda/envs/usrp/]

Cancel    Select

```
[ ]: #Integers
a = 1
a

[ ]: #floats
b = 2.
print(b)
print(type(b))

[ ]: #strings
s = 'list of letters'
print(s)
print(type(s))

[ ]: #lists
l = [4,5,2., 'hello', 'world']
l
```

Display a menu    0    \$    2    Python 3 (ipykernel) | Idle    Mode: Command    Ln 1, Col 1    python-intro.ipynb