DS-GA 1007 | Lecture 4

Programming for Data Science

Jeremy Curuksu, PhD
NYU Center for Data Science
jeremy.cur@nyu.edu

December 17, 2022

DS-GA 1007 Curriculum

Programming for Data Science:

- Introduction to Programming in Python
- Best Practice Programming and Software Engineering
- Program Efficiency
- **Interacting with Programs**
- Array Manipulation for Scientific Computing
- Data Visualization
- Advanced Data Objects ($\times 4$)
- Environments for Collaborative Programming
- Industrial Applications

Interacting with Programs

Last week:

- Run Time and Algorithmic Complexity
- Examples of Iterative and Recursive Algorithms
- Examples of Searching and Sorting Algorithms

Today:

- **Python Distributions, Editors and IDEs**
- **Python Libraries and Virtual Environments**
- **Operating System Command Line Interface**

Interacting with Programs

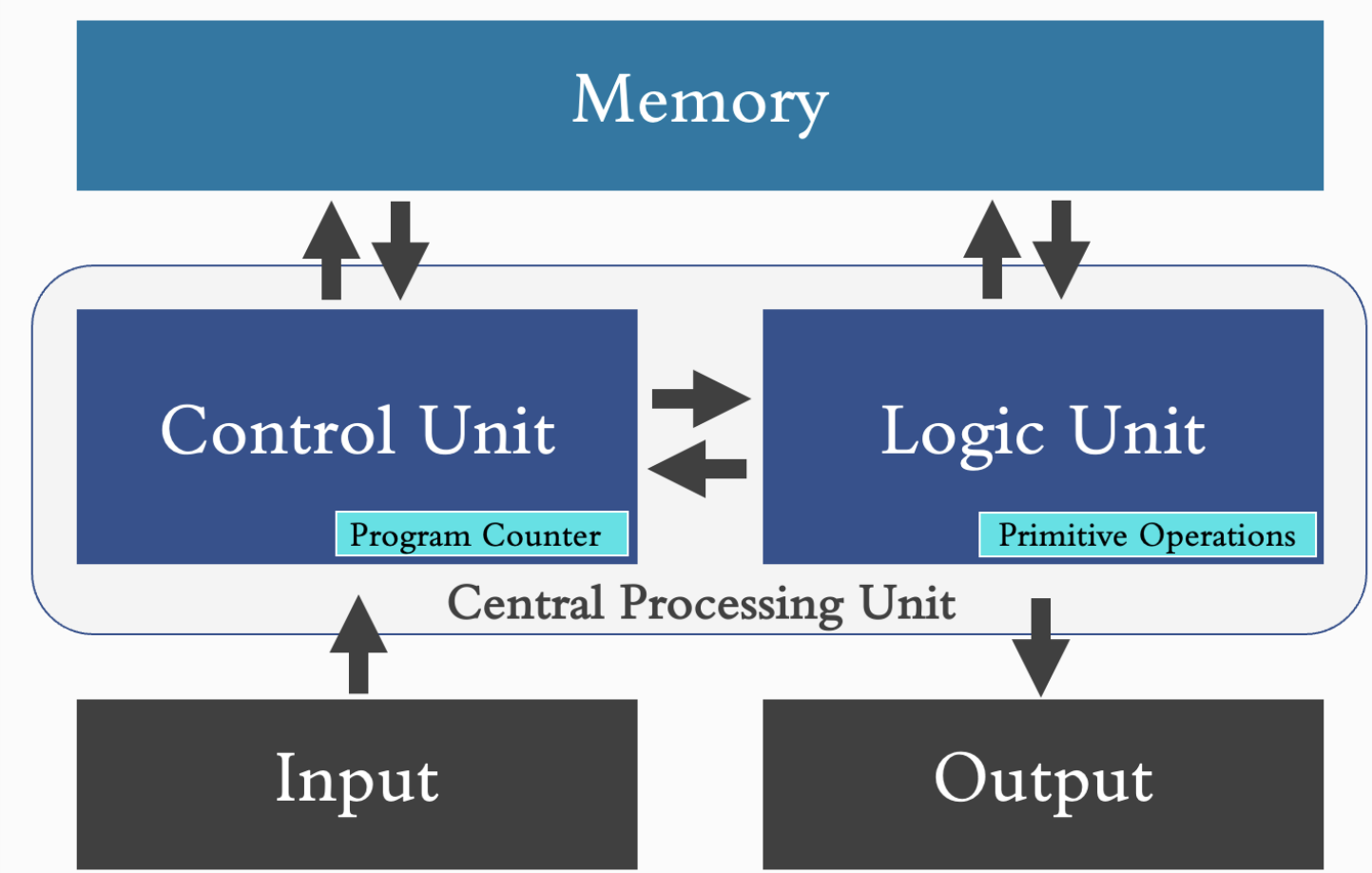
Do not forget:

- Today's lecture includes practice code examples in an accompanying Jupyter notebook

Interacting with Programs

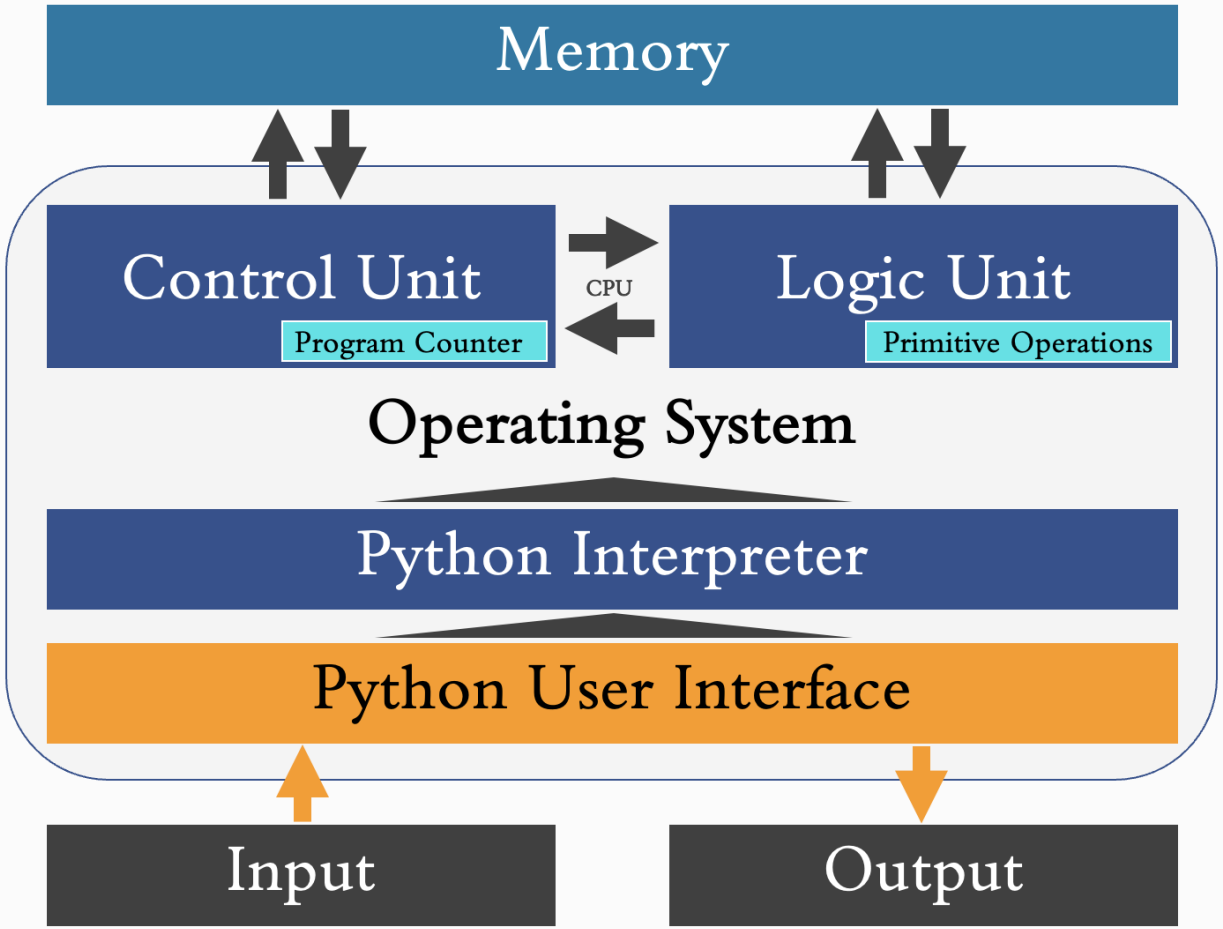
Architecture of *stored program* computers

Programs Interface with the Operating System



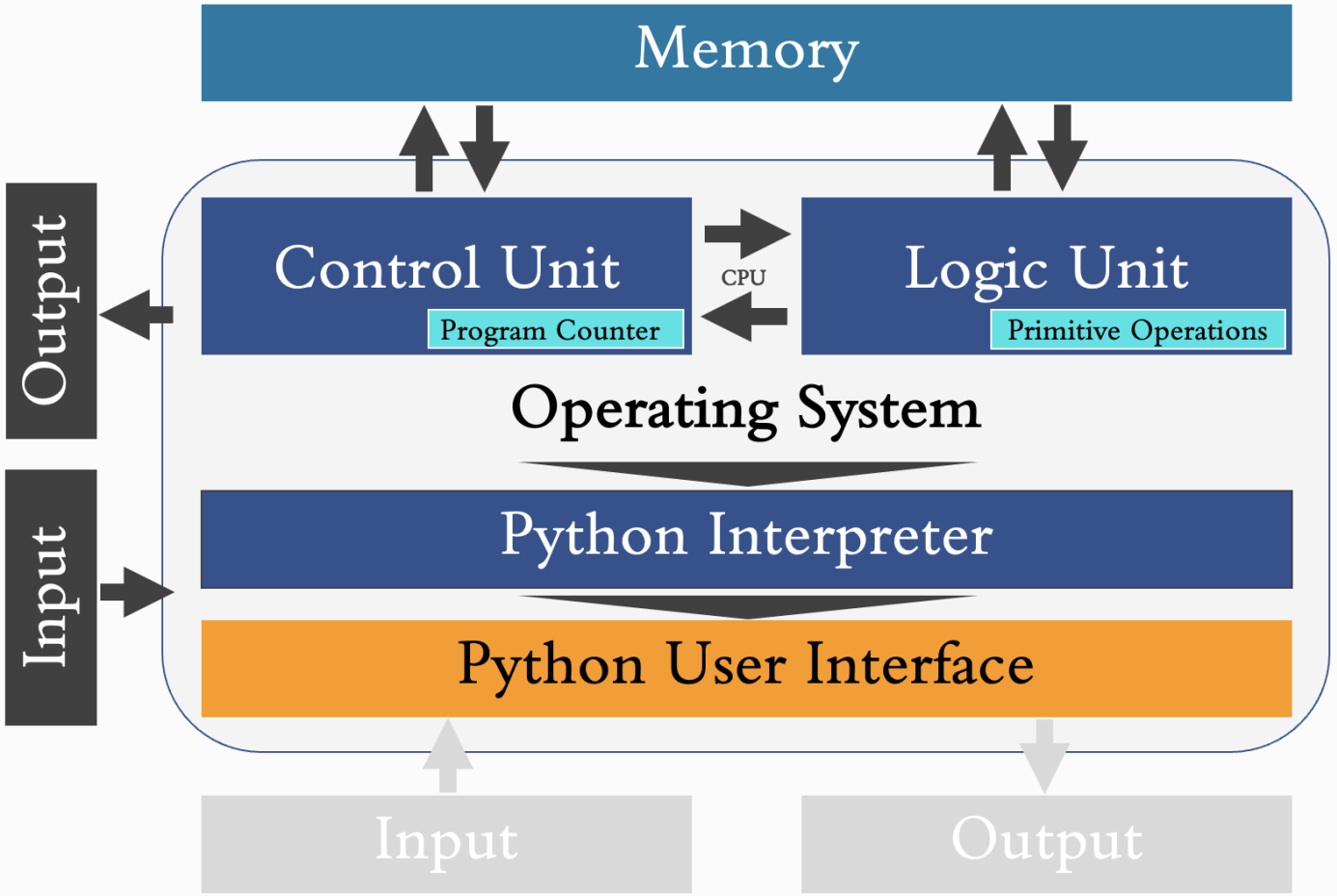
Architecture of *stored program* computers

Programs Interface with the Operating System



Architecture of *stored program* computers

Programs Interface with the Operating System



Python Distributions, Editors and IDEs

Python Distributions

Minimum required:

- **C-Python source distribution**
  - Python interpreter
  - Basic Python packages and package installer (PIP)
  - Basic editor (IDLE)

For this course:

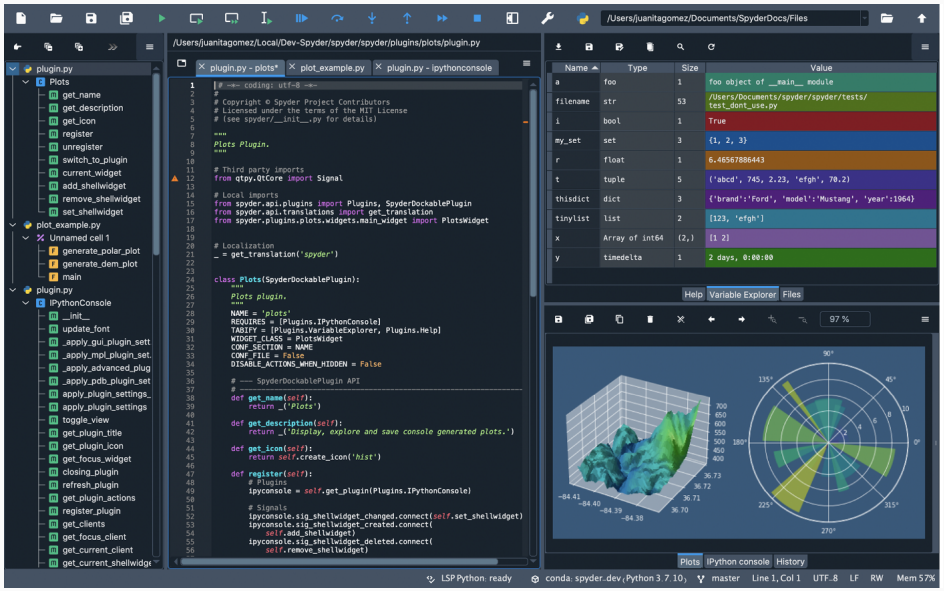
- **Anaconda distribution**:
  - C-Python source distribution
  - Packages for data science (NumPy, Pandas, etc)
  - Package and Environment Manager (Conda)
  - GUI (Anaconda Navigator)
  - Integrated Development Environments (Jupyter, Spyder)

Python Editors and IDEs

The Top 6...

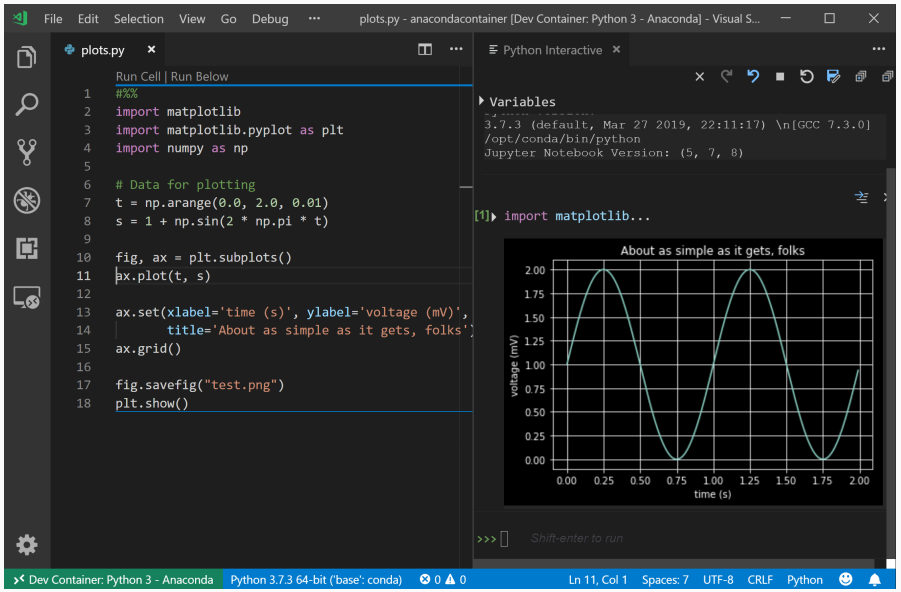
Python Editors and IDEs

Spyder



spyder-ide.org

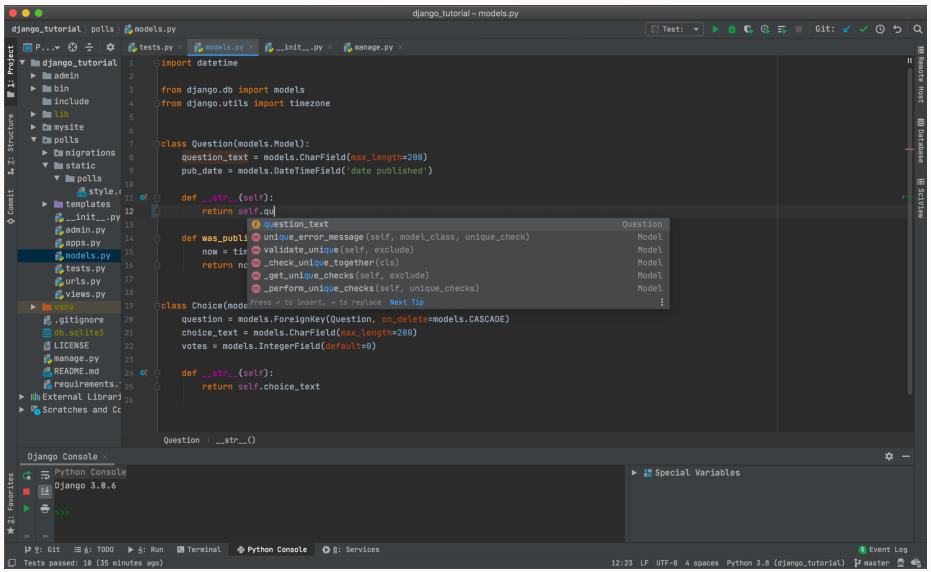
Visual Studio Code



code.visualstudio.com

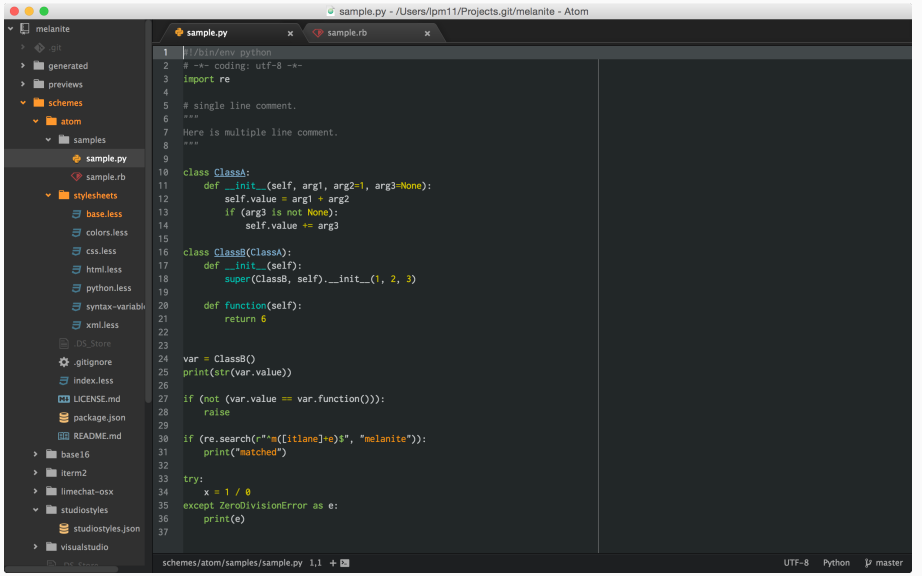
Python Editors and IDEs

PyCharm



jetbrains.com/pycharm

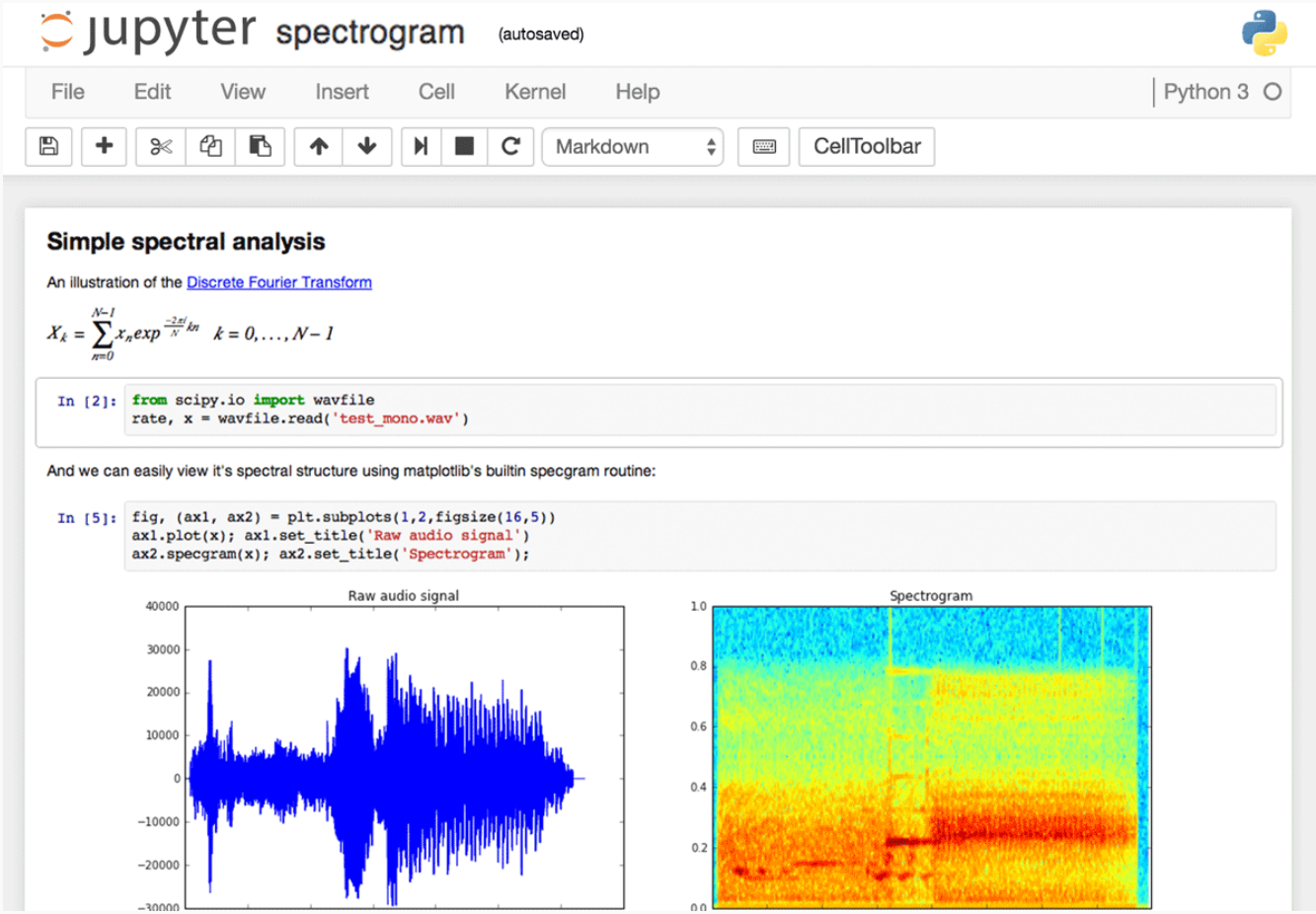
Atom



atom.io

Python Editors and IDEs

Jupyter Notebook

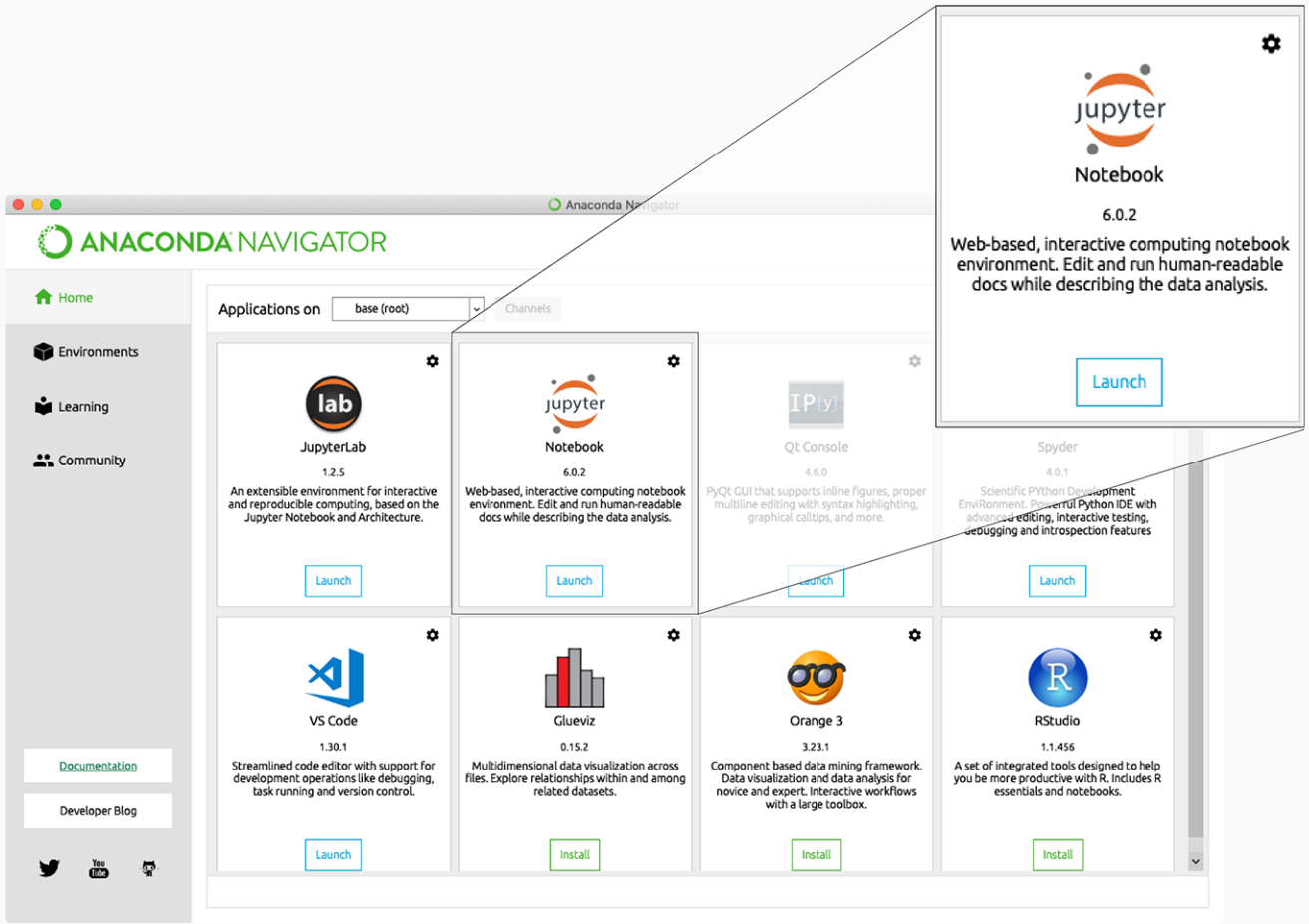


jupyter.org

Open Jupyter Notebook

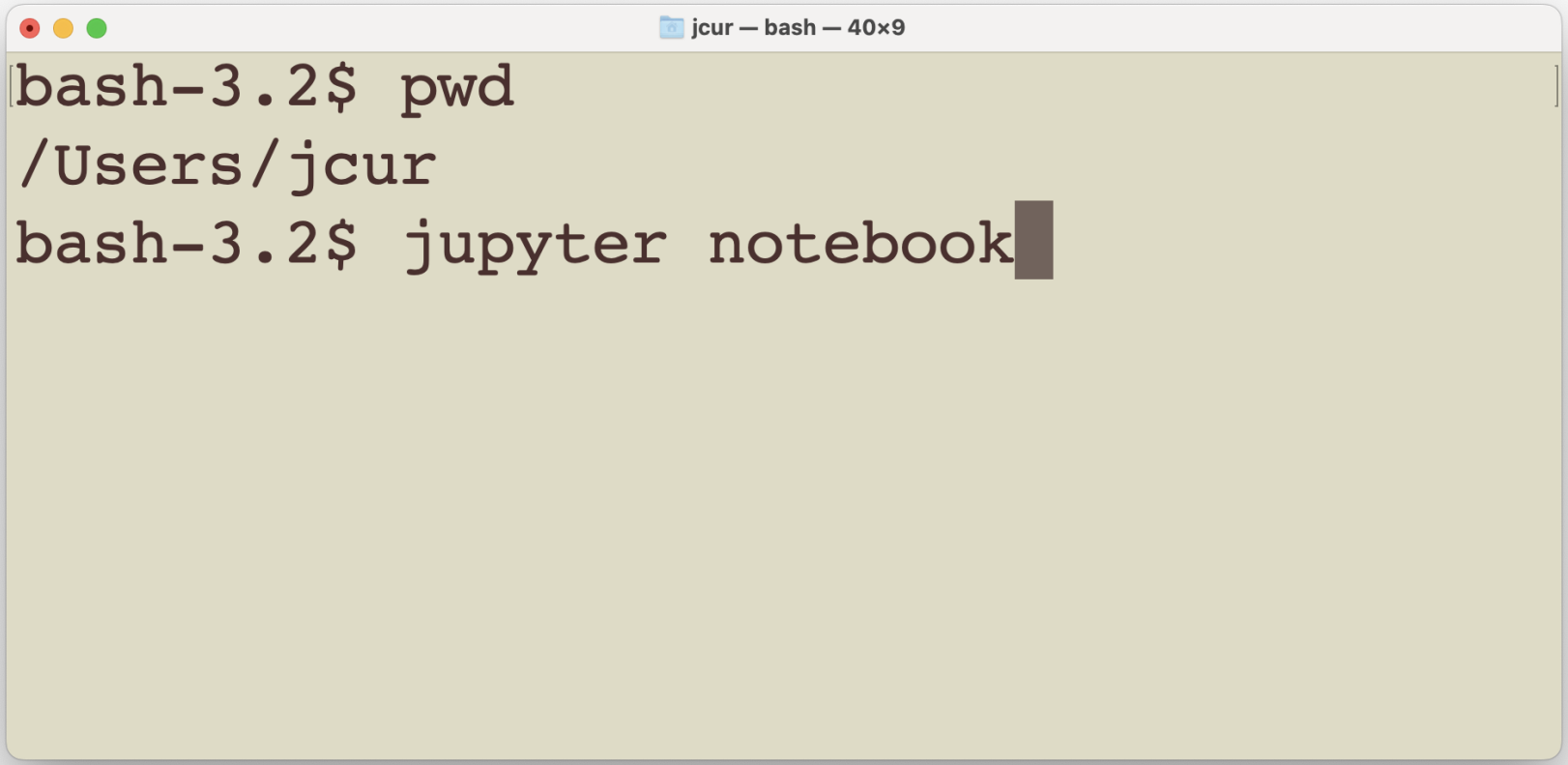
Graphical User Interface (GUI)

Open Jupyter Notebook from the Anaconda Navigator



Command Line Interface (CLI)

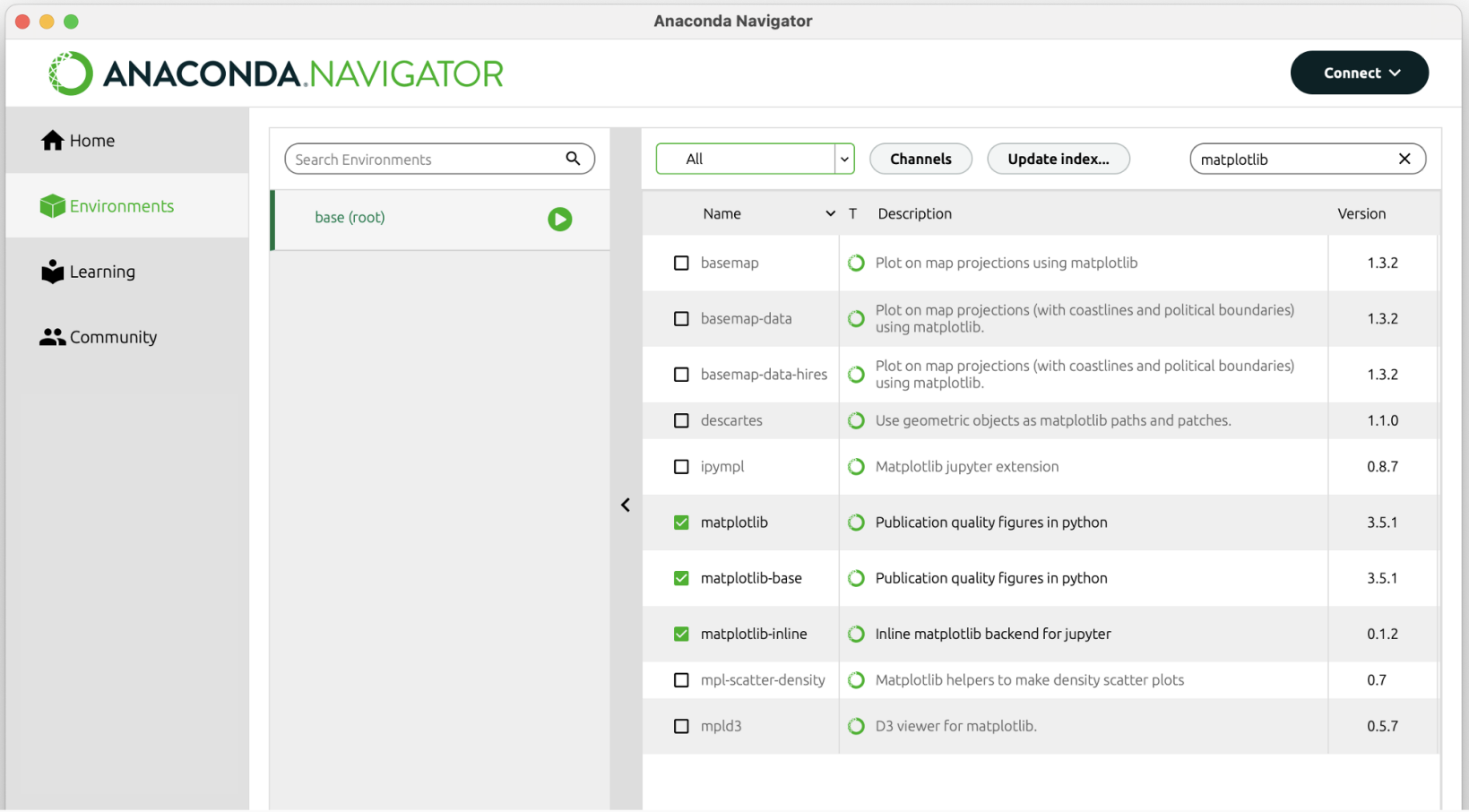
Open Jupyter Notebook from the OS Console



Python Libraries and Virtual Environment

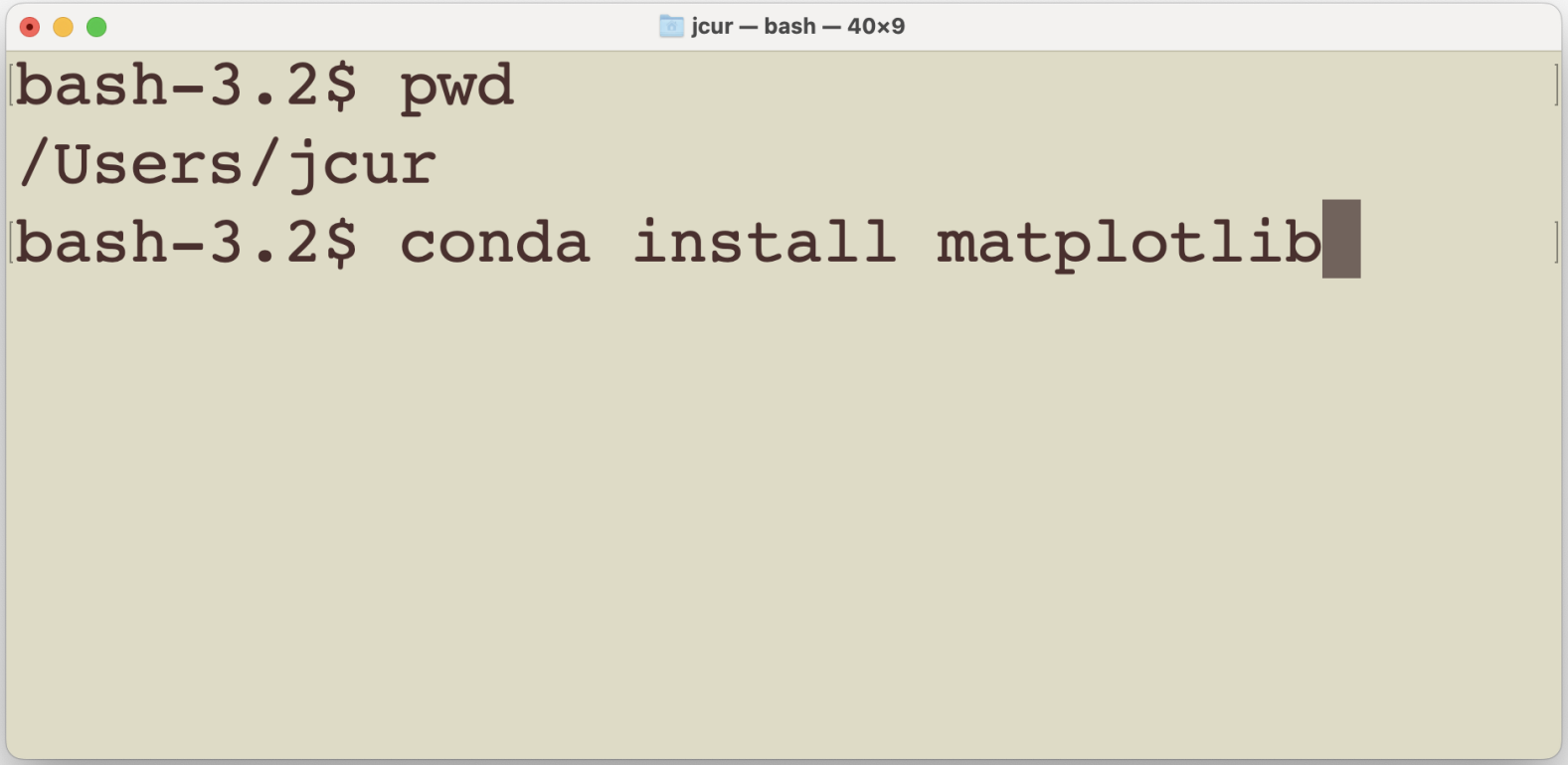
Installing a New Python Library

From the Anaconda Navigator (GUI)



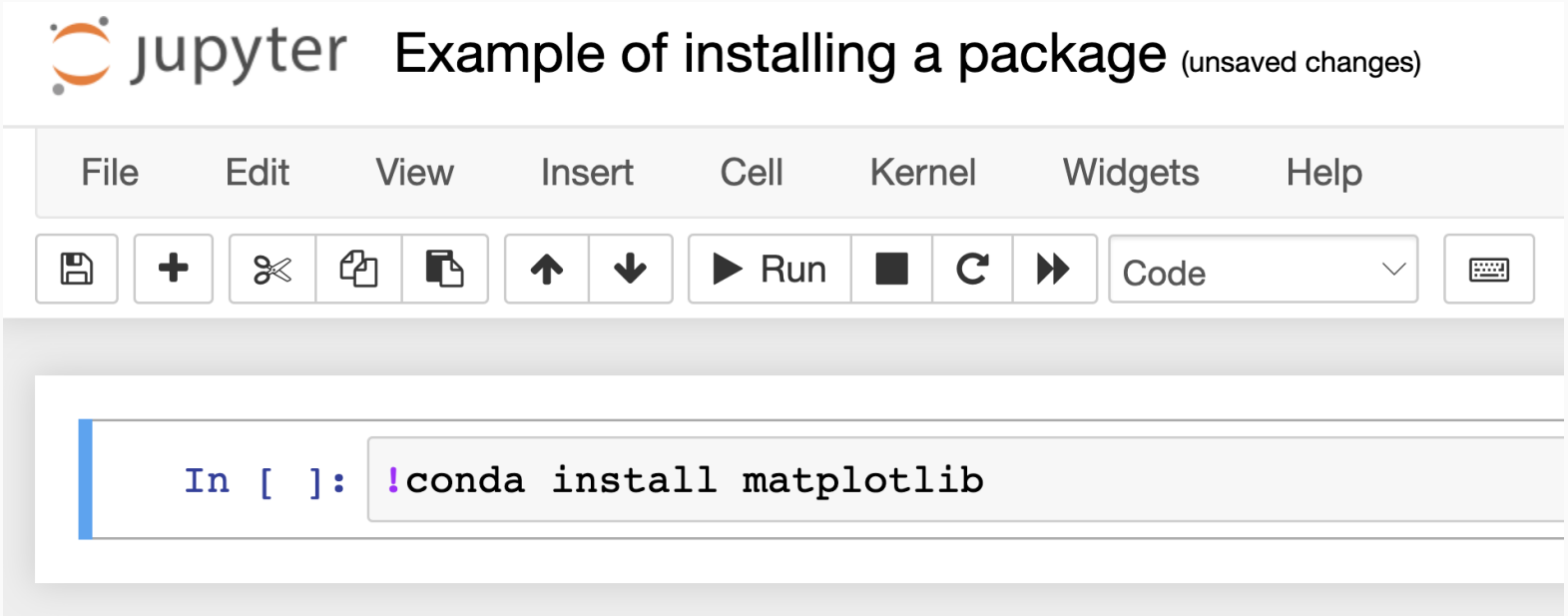
Installing a New Python Library

From the Command Line Interface (CLI)



Installing a New Python Library

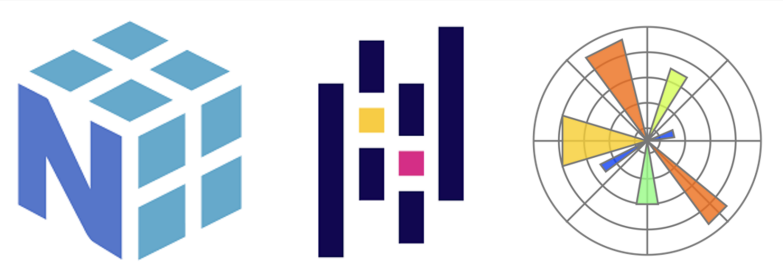
From the Jupyter Notebook



Python Libraries for Data Science

DS-GA 1007 essential libraries:

- **NumPy**: Fast operations on arrays of numerical data
- **Pandas**: Manipulation and analysis of complex data frames (builds on NumPy)
- **Matplotlib**: Graphical vizualization of data (builds on NumPy)



Versions of Python Libraries

- **Python libraries have different versions** that correspond to different public releases (solve bugs, add features, improve features)
- **Latest version is generaly best**, installed by default (a specific version can be explicitly installed too)
- **Risk of compatibilty issues**: A program that uses many libraries may not work after changing the version of these libraries

Python Virtual Environment

Virtual Environment: Container with specific versions of libraries and interpreter needed to execute a program

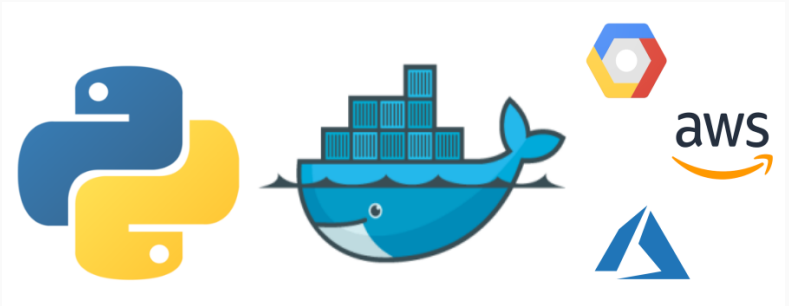
Why use a Virtual Environment?

- If you create different Python programs that need different versions of Python libraries or interpreter
- You can have multiple Virtual Environments on your computer. They allow Python libraries to be installed in an isolated location for a particular application, rather than being installed globally

Python Virtual Environment

What a Virtual Environment is and is not:

- ✓ A **Virtual Environment** includes specific versions of Python libraries/interpreter needed for a program
- × A **Docker Container** (beyond scope of this course) includes all OS dependencies needed for a program
- × A **Virtual Machine** (beyond scope of this course) includes all OS and hardware-related dependencies to isolate ('virtualize') a guest OS on a host machine



Interacting with the Operating System

Interacting directly with the OS

Graphical User Interface

- ✓ Easy to use by clicking mouse on visual designs
- × Lots of manual redundant activities
- × Not portable, ”know-how” can become obsolete
- × Only control options shown

Command Line Interface

- × Need to know what to type in the terminal
- ✓ Can automate everything, at scale
- ✓ Highly portable, remains valid over time
- ✓ Full control on everything

This isn't just about OS...

Visual Design Interface

- × *WYSIWYG* ("What You See Is What You Get") => What You See Is All You've Got
- × Execution of commands happens in real time as you click on buttons
- × No record: Not Portable, Manual, Obsolescent

Logical Design Interface

- ✓ You have full control: type in the exact desired logical structure
- ✓ Entire set of commands can be typed, executed and changed any time
- ✓ Program code: Portable, Automatable, Timeless

How to access a CLI on your OS

The Shell (ex: *bash*) is an interpreter: It reads keyboard commands from the Unix (or Linux) language and passes them to the OS to carry out. All modern OS come with a pre-installed Shell terminal emulator program

To access the Shell on...

- **Windows:** Open **cmd** (*Command Prompt*)
- **Mac:** Open **Terminal**
- **Linux:** Open **Linux Console**
- **Python Interpreter:** Precede command by "**!**"
- **(Web-) Applications:** It varies (on Jupyter: **Terminal**)

Linux commands to navigate files

Navigate files and directories	
pwd	Print name of current directory
cd	Change directory
ls	List directory contents
file	Determine file type
less	View text file contents
head/tail	Output first/last part of file

Linux commands to manipulate files

Manipulate files and directories	
cp	Copy files and directories
mv	Move/rename files and directories
mkdir	Create directories
rm	Remove files and directories
chmod	Change a file's permissions

Linux commands to find things

Find what you are looking for	
man	Display a command's user manual page
find	Find files whose names match a pattern
grep	Find lines of text file matching a pattern
sed s/x/y/ f	Find and replace x by y in file f
history -n	Print last n commands typed in

Linux commands to parse text files

VI editor commands	
vi, :q	Open VI, quit VI
:w	Save file
0	Move cursor to begining of line
$	Move cursor to end of line
9G	Move cursor to line 9
x, 9x	Delete current character, delete 9 characters
dd, 9dd	Delete current line, delete 9 lines
/text	Find string 'text' in entire file
:2,9,s/s1/s2/	From line 2 to 9, find and replace s1 by s2

Linux commands to manage jobs

Manage execution of programs	
command > filename	Redirect output to file (>> to append)
command | command2	Pipe output to input of command2
cat filename	Print file contents to output
jobs	List active processes
top	Monitor processes dynamically
bg, fg	Place process in background/foreground
kill	Terminate a process
sudo	Execute command as another user

Advanced Linux commands

Advanced Linux commands	
alias	Create alias for a command
gzip	Compress or uncompress files
tar, zip	Package files and directories
ssh, scp	Remotely log in another computer
#!/bin/bash	Invoke interpreter within script
if ; then ; else ; fi	Branching control
for : do ; done	Looping control

Conda commands

Commands are *programs*. A program is a command when executed in the CLI. Conda offers *commands* to manage Python environments

Manage Python environments on your OS	
conda install	Install a Python package
conda update	Update a previously installed package
conda list	List all packages in environment
conda env list	List all environments
conda create	Create a new environment
conda activate	Activate a specific environment

Conda commands

Commands are *programs*. A program is a command when executed in the CLI. Conda offers *commands* to manage Python environments

Manage Python environments on your OS	
conda install	Install a Python package
conda update	Update a previously installed package
conda list	List all packages in environment
conda env list	List all environments
conda create	Create a new environment
conda activate	Activate a specific environment

Click here for more information: conda-cheatsheet.pdf

Thank you!!