Python

leangaurav

Tools

Tools and Softwares to use

- Anaconda
  - Jupyter
  - Sypder
- Editor + CMD/Terminal
  - Notepad++ (windows only)
  - Sublime
  - Atom etc.
- Git and Github
  - Create your **github** account
  - On windows install **gitbash**

Anaconda

- Anaconda is distribution of Python which comes with bunch of tools
- We will use Anaconda3 and tools which come along:
  - Jupyter (mostly)
  - Spyder
- Install Python 3.x version from here:
  - https://www.anaconda.com/distribution/

* For windows if asked to add to PATH, put tick in check box

Notepad++ and Terminal

- You can write python code/scripts in any text editor and run from terminal.
- In any text editor, write your python code and save file with extension *.py*
- Now open a terminal in same folder as your python script and run like this:

  python <filename>

replace file name with your script name and don't put <>

Git and Github

- Git is a tool for version control
- Github is a website that allows people to collectively work on a project
- On Mac/Linux ***git*** comes preinstalled or you can install if running **git** on terminal gives error
- On windows install ***gitbash*** to use git.

Other Tools and Stuff

- Try other **IDE** to write python code like **PyCharm**.
- Books:
  - Learning Python (Beginners)
  - Programming Python (Intermediate)
  - Python Cookbook, Fluent Python (Advanced)
- **pip**: This is important to understand packaging and dependency management

What is Python

Python: Language

- Python is a Programming Language
- Python is an interpreted language.
- But it uses a hybrid model to improve performance
- Dynamically typed and case sensitive

Compiled vs Interpreted

- Compilation:
  - Convert to binary and save
  - Run saved binary
- Interpretation:
  - Read one instruction at a time
  - Convert to binary and run
  - Repeat till done
- **Compiled** languages are ***faster*** than interpreted.
- **Interpreted** one give ***platform independence***.

Which python are we using

- Anaconda: collection of some tools(Jupyter etc.) and libraries (pandas, numpy etc.) along with Cpython. That justifies the size difference
- Cpython: official Python implementation (available at python.org)
- Source code of Cpython is written in C programming language
- There are other implementations like Jython, PyPy etc.

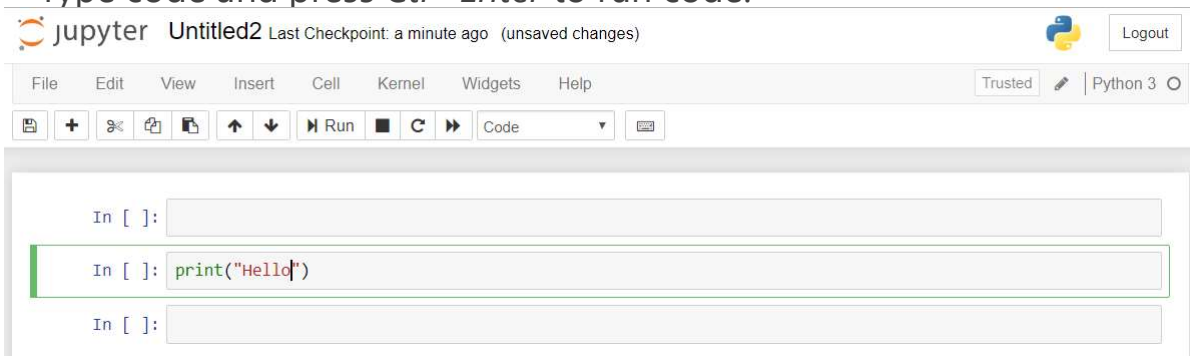
Running Python Code

```
>>> print("Hello World")
```

Jupyter Notebook

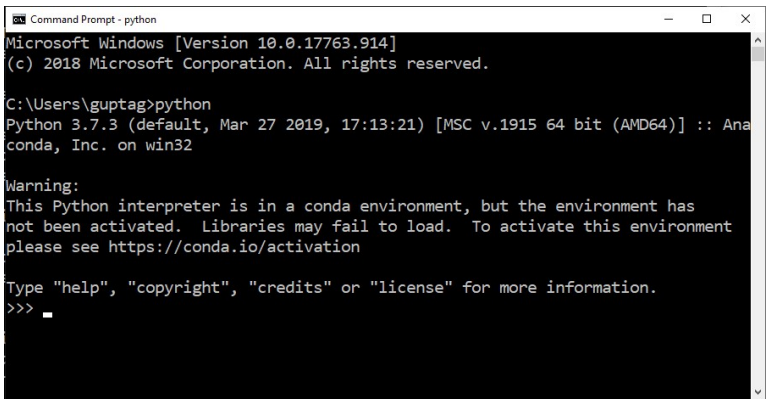
- Jupyter Notebook opens in browser
- On top right there is new button: with Python3 and Folder options

- *New → Python3 :* Creates new notebook (with extension .ipynb)
- Cells: a block where you write code
- Type code and press *Ctr+Enter* to run code.



Python Interpreter

- Open a terminal or CMD and type python (in small)
- Notice **>>>**
- Check Version 3.x.x 3.7.7 here
- Should not be 2.x.x



Notepad++ and Terminal

- You can write python code in a text editor and run in a terminal.
- In a text editor, write your code and save file with extension ***.py***
- Now open terminal in folder where python script is saved and run like this:

  python <filename>

  * replace file name with script name and don't put <>

  Example: if file is saved as test.py

  python test.py

Building Blocks of Code

Tokens

- Four kind of Tokens:
  - Keywords
  - Identifiers
  - Literals
  - Operators
- All code elements fall into one of these category

- Keywords: Special reserved words predefined by language

- List of keywords (Python 3.8)

False	*await*	*else*	*import*	*pass*
None	*break*	*except*	*in*	*raise*
True	*class*	*finally*	*is*	*return*
and	*continue*	*for*	*lambda*	*try*
as	*def*	*from*	*nonlocal*	*while*
assert	*del*	*global*	*not*	*with*
async	*elif*	*if*	*or*	*yield*

- **Identifiers**: These are names.
- Rules:
  - Use letters in lowercase (a to z) or uppercase (A to Z)
  - Digits (0 to 9) allowed but not in beginning
  - Underscore (_) allowed anywhere
  - Should not be name of keyword

- Variable name, Class name, Function name and Module name are all identifiers

- Python has special identifiers (having two underscores):
  - `__*__` : Special Reserved system defined names
  - `__*` : Used to define private class members

DIY 1

- Read about operator precedence

  https://docs.python.org/3/reference/expressions.html#operator-precedence

- Literals: Constant of fixed values

- Type of literals:

int	:	1,-1,0….
float	:	-1.0, 0.0, 3.14
string	:	‘’, ‘ ’, ‘a’, ‘abcd’
bool	:	True, False
None	:	Empty

- Operators: Python has Unary and Binary operators
  - **
  - +x, -x, ~x
  - *, @, /, //, %
  - +,-
  - in, not in, is, is not, <, <=, >, >=, !=, ==
  - not, and, or
  - >>, <<

DIY 2

- Read python naming convention
  https://www.python.org/dev/peps/pep-0008/#naming-conventions
- Write Python code to import and print list of all keywords
- What is **PYTHONPATH**

DIY 2

- Print List of Keywords

```
C:\Users\guptag>python
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on
win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue'
, 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', '
in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with'
, 'yield']
>>>
```