Introduction to Python

GITAA
Transforming careers

Popular tools used in data science

- Data pre-processing and analysis
  - Python, R, Microsoft Excel, SAS, SPSS
- Data exploration and visualization
  - Tableau, Qlikview, Microsoft Excel
- Parallel and distributed computing incase of big data
  - Apache Spark, Apache Hadoop

Evolution of Python

- Python was developed by Guido van Rossum in the late eighties at the 'National Research Institute for Mathematics and Computer Science' at Netherlands
- Python Editions
  - Python 1.0
  - Python 2.0
  - Python 3.0

Python as a programming language

- Supports multiple programming paradigm
  - Functional, Structural, OOPs, etc.
- Dynamic typing
  - Runtime type safety checks
- Reference counts
  - Deallocates objects which are not used for long
- Late binding
  - Methods are looked up by name during runtime
- Python's design is guided by 20 aphorisms as described in Zen of Python by Tim Peters

Python as a programming language

- Standard CPython interpreter is managed by “Python Software Foundation”
- There are other interpreters namely JPython (Java), Iron Python (C#), Stackless Python (C, used for parallelism), PyPy (Python itself JIT compilation)
- Standard libraries are written in python itself
- High standards of readability

Python as a programming language

- Cross-platform (Windows, Linux, Mac)
- Highly supported by a large community group
- Better error handle

Python as a programming language

- Comparison to Java
- Python vs Java
  - Java is statically typed i.e. type safety is checked during compilation (static compilation)
  - Thus in Java the time required to develop the code is more
  - Python which is dynamically typed compensates for huge compilation time when compared to Java
  - Codes which are dynamically typed tend to be less verbose therefore offering more readability

Advantages of using python

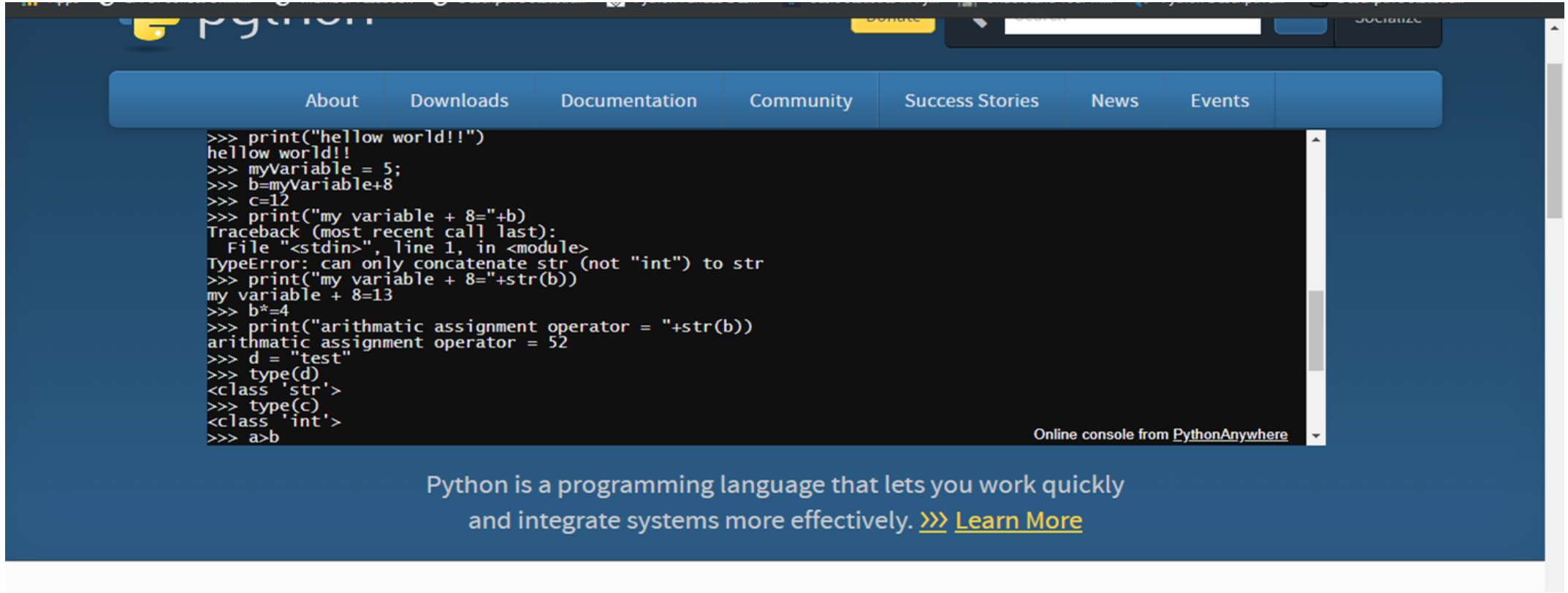
- Python has several features that make it well suited for data science
- Open source and community development
  - Developed under Open Source Initiative license making it free to use and distribute even commercially
- Syntax used is simple to understand and code
- Libraries designed for specific data science tasks
- Combines well with majority of the cloud platform service providers

Coding environment

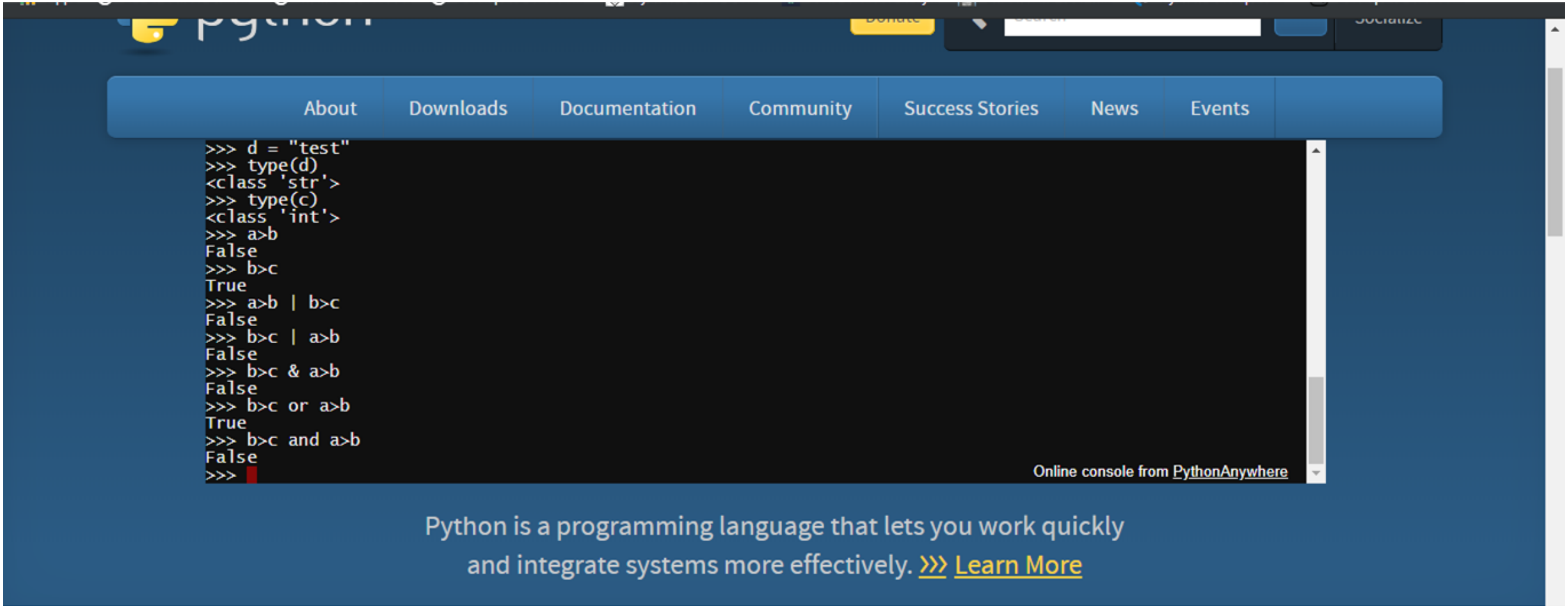
- A software program can be written using a terminal, a command prompt (cmd), a text editor or through an Integrated Development Environment (IDE)
- The program needs to be saved in a file with an appropriate extension (.py for python, .mat for matlab, etc...) and can be executed in corresponding environment (Python, Matlab, etc…)
- Integrated Development Environment (IDE) is a software product solely developed to support software development in various or specific programming language(s)

Coding environment

- Python 2.x support will be available till 2020
- Python 3.x is an enhanced version of 2.x and will only be maintained from 3.6.x post 2020
- Install basic python version or use the online python console as in https://www.python.org/
- Execute following commands and view the outputs in terminal or command prompt
  - Basic print statement
  - Naming conventions for variables and functions, operators
  - Conditional operations, looping statements (nested)
  - Function declaration and calling
  - Installing modules



https://www.python.org/



https://www.python.org/

Integrated development environment (IDE)

- Software application consisting of a cohesive unit of tools required for development
- Designed to simplify software development
- Utilities provided by IDEs include tools for managing, compiling, deploying and debugging software

Coding environment- IDE

- An IDE usually comprises of
  - Source code editor
  - Compiler
  - Debugger
  - Additional features include syntax and error highlighting, code completion
- Offers supports in building and executing the program along with debugging the code from within the environment

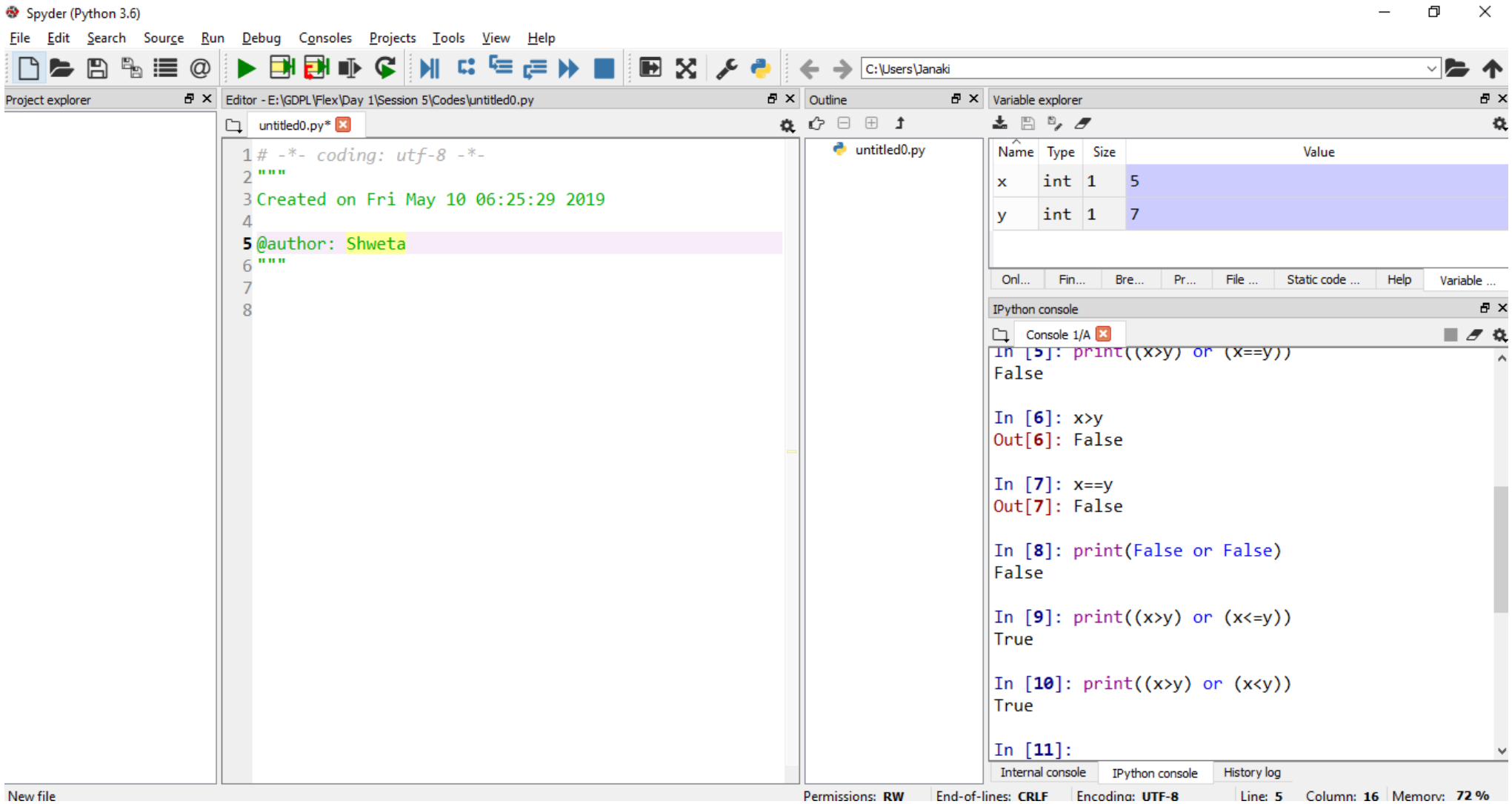
Coding environment- IDE

- Best IDEs provide version control features
- Eclipse+PyDev, SublimeText, Atom, GNU Emacs, Vi/Vim, Visual Studio, Visual Studio Code are general IDEs with python support
- Apart from these some of the python specific editors include Pycharm, Jupyter, Spyder, Thonny

Spyder

- Supported across Linux, Mac OS X and Windows platforms
- Available as open source version
- Can be installed separately or through Anaconda distribution
- Developed for Python and specifically data science
- Features include
  - Code editor with robust syntax and error highlighting
  - Code completion and navigation
  - Debugger
  - Integrated document
- Interface similar to MATLAB and RStudio

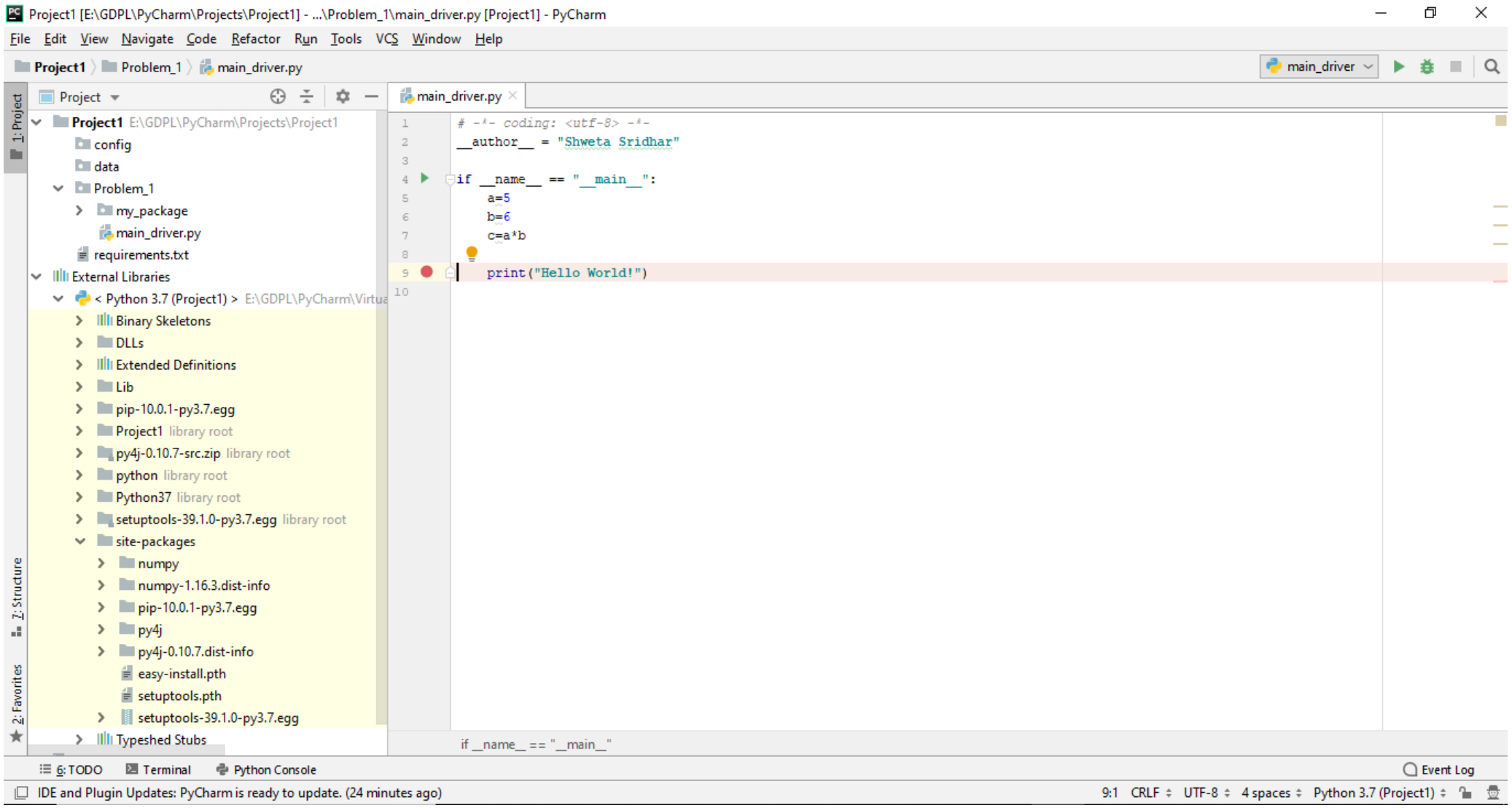
Spyder


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Janaki
Project explorer
Editor - E:\GDPL\Flex\Day 1\Session 5\Codes\untitled0.py
untitled0.py*
1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri May 10 06:25:29 2019
4
5 @author: Shweta
6 """
7
8
Outline
untitled0.py
Variable explorer
Name Type Size Value
x int 1 5
y int 1 7
Onl... Fin... Bre... Pr... File ... Static code ... Help Variable ...
IPython console
Console 1/A
In [5]: print((x>y) or (x==y))
False
In [6]: x>y
Out[6]: False
In [7]: x==y
Out[7]: False
In [8]: print(False or False)
False
In [9]: print((x>y) or (x<=y))
True
In [10]: print((x>y) or (x<y))
True
In [11]:
Internal console IPython console History log
New file
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 5 Column: 16 Memory: 72 %

PyCharm

- Supported across Linux, Mac OS X and Windows platforms
- Available as community (free open source) and professional (paid) version
- Supports only Python
- Can be installed separately or through Anaconda distribution
- Features include
  - Code editor provides syntax and error highlighting
  - Code completion and navigation
  - Unit testing
  - Debugger
  - Version control

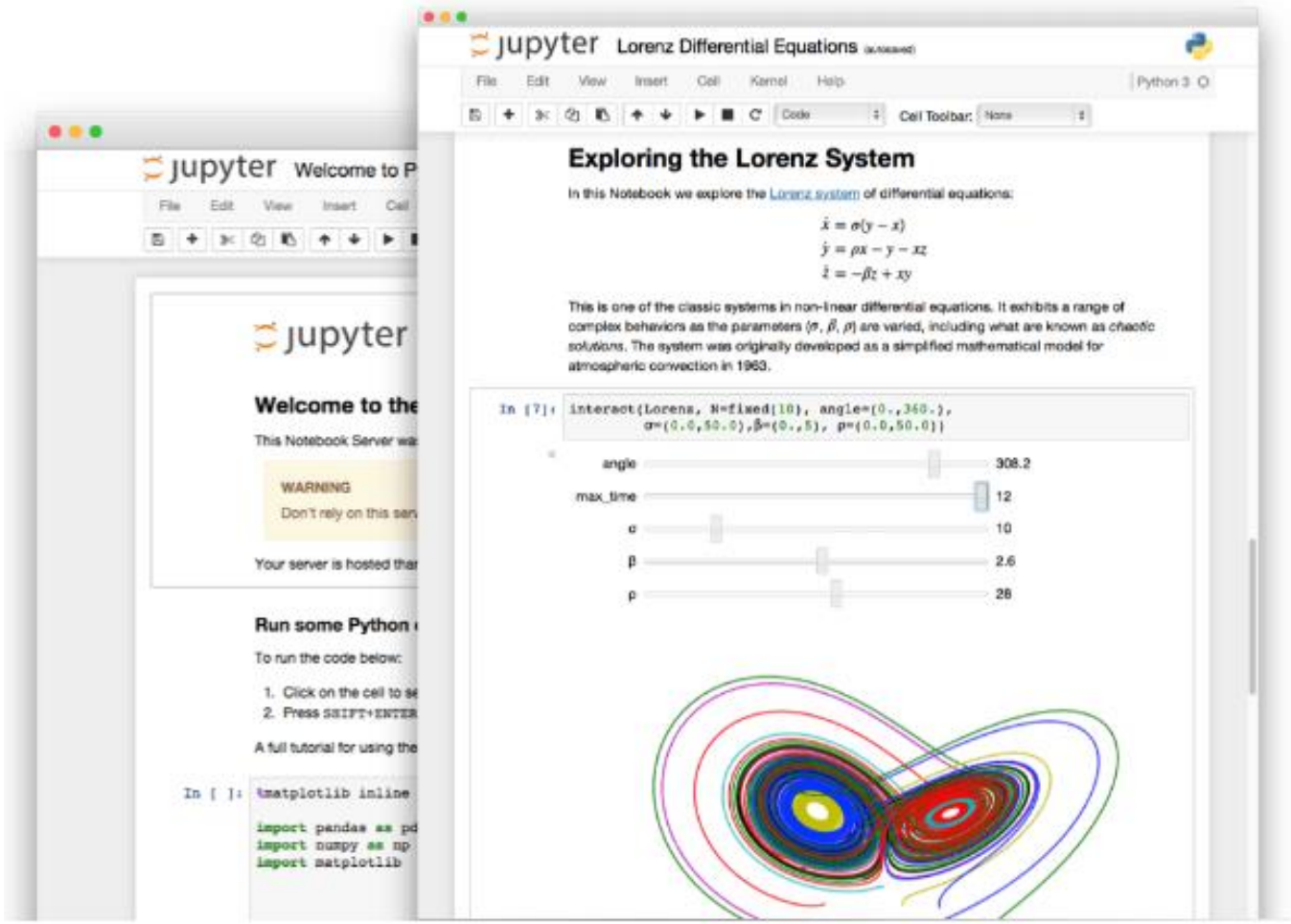
PyCharm


Project1 [E:\GDPL\PyCharm\Projects\Project1] - ...\Problem_1\main_driver.py [Project1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Project1 Problem_1 main_driver.py
main_driver
Project
Project1 E:\GDPL\PyCharm\Projects\Project1
config
data
Problem_1
my_package
main_driver.py
requirements.txt
External Libraries
< Python 3.7 (Project1) > E:\GDPL\PyCharm\Virtua
Binary Skeletons
DLLs
Extended Definitions
Lib
pip-10.0.1-py3.7.egg
Project1 library root
py4j-0.10.7-src.zip library root
python library root
Python37 library root
setuptools-39.1.0-py3.7.egg library root
site-packages
numpy
numpy-1.16.3.dist-info
pip-10.0.1-py3.7.egg
py4j
py4j-0.10.7.dist-info
easy-install.pth
setuptools.pth
setuptools-39.1.0-py3.7.egg
Typeshed Stubs
main_driver.py
# -*- coding: <utf-8> -*-
__author__ = "Shweta Sridhar"
if __name__ == "__main__":
a=5
b=6
c=a*b
print("Hello World!")
if __name__ == "__main__"
1: Project
2: Favorites
7: Structure
6: TODO
Terminal
Python Console
Event Log
IDE and Plugin Updates: PyCharm is ready to update. (24 minutes ago)
9:1 CRLF UTF-8 4 spaces Python 3.7 (Project1)

Jupyter Notebook

- Web application that allows creation and manipulation of documents called ‘notebook’
- Supported across Linux, Mac OS X and Windows platforms
- Available as open source version

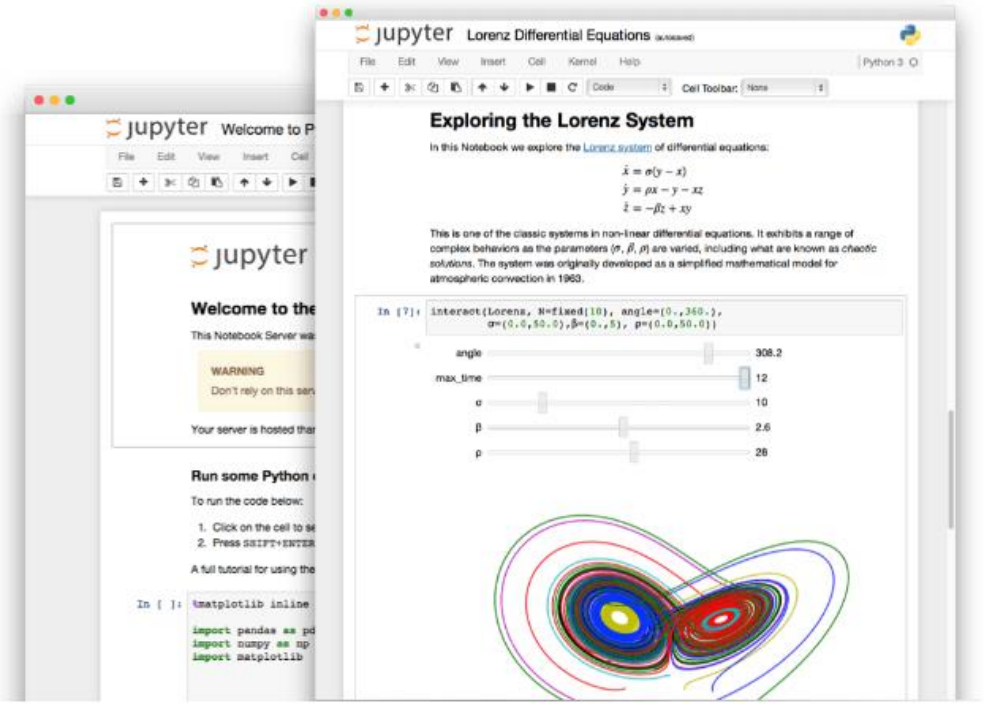
Jupyter Notebook



Source-https://jupyter.org/

Jupyter Notebook

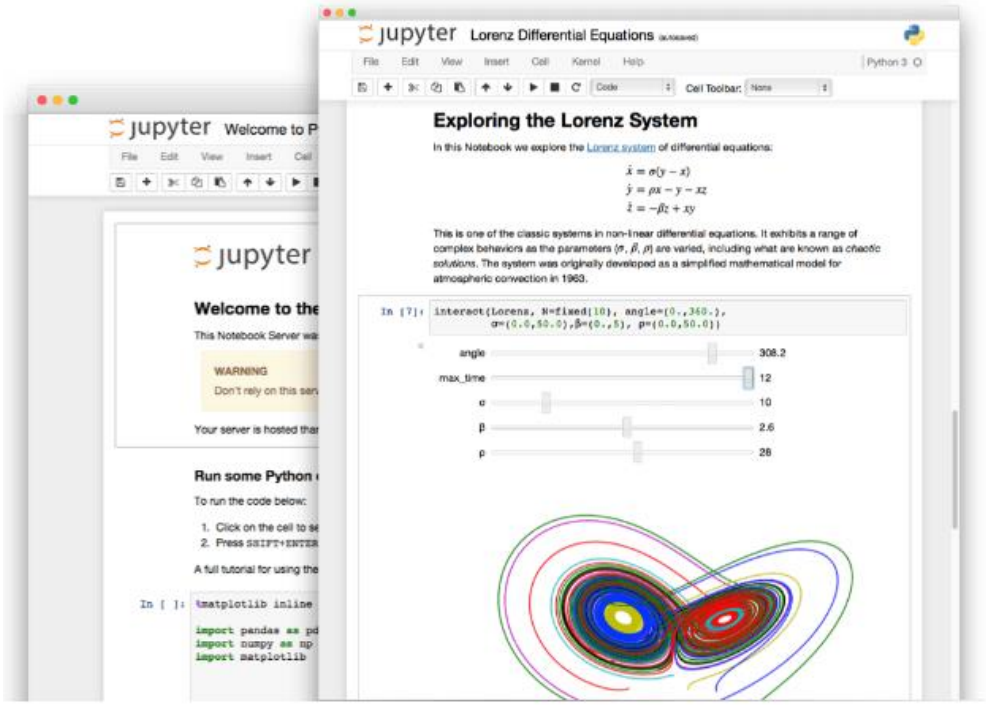
- Bundled with Anaconda distribution or can be installed separately
- Supports Julia, Python, R and Scala
- Consists of ordered collection of input and output cells that contain code, text, plots etc.



Source-https://jupyter.org/

Jupyter Notebook

- Allows sharing of code and narrative text through output formats like PDF, HTML etc.
  - Education and presentation tool
- Lacks most of the features of a good IDE



Source-https://jupyter.org/

How to choose the best IDE?

- Requirements
- Working with different IDEs helps us understand our own requirement


operation == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True
selection at the end -add
ob.select= 1
er_ob.select=1
ntext.scene.objects.acti
("Selected" + str(modifier
irror_ob.select = 0
bpy.context.selected_ob
ata.objects[one.name].se
int("please select exact

THANK YOU