



Introduction to Python

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

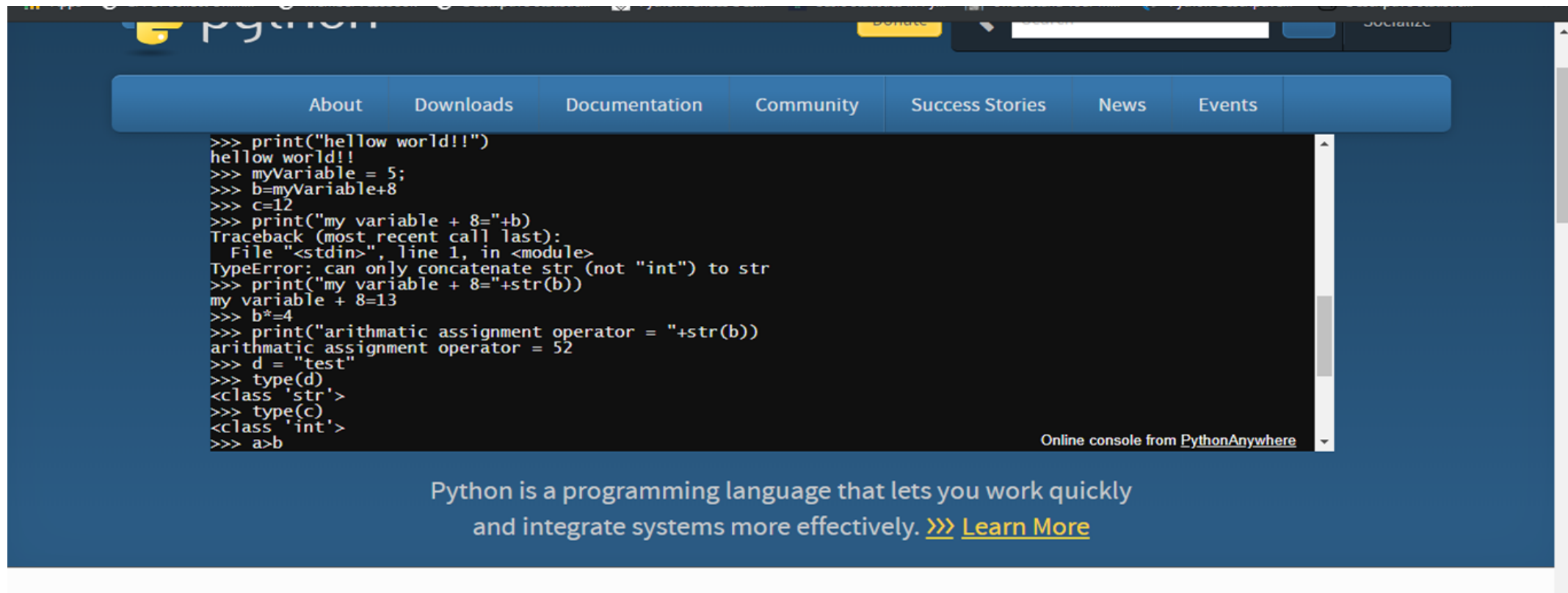
- Python has several features that make it well suited for data science
- Open source and community development
 - Developed under Open Source Initiative license making it free to use and distribute even commercially
- Syntax used is simple to understand and code
- Libraries designed for specific data 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

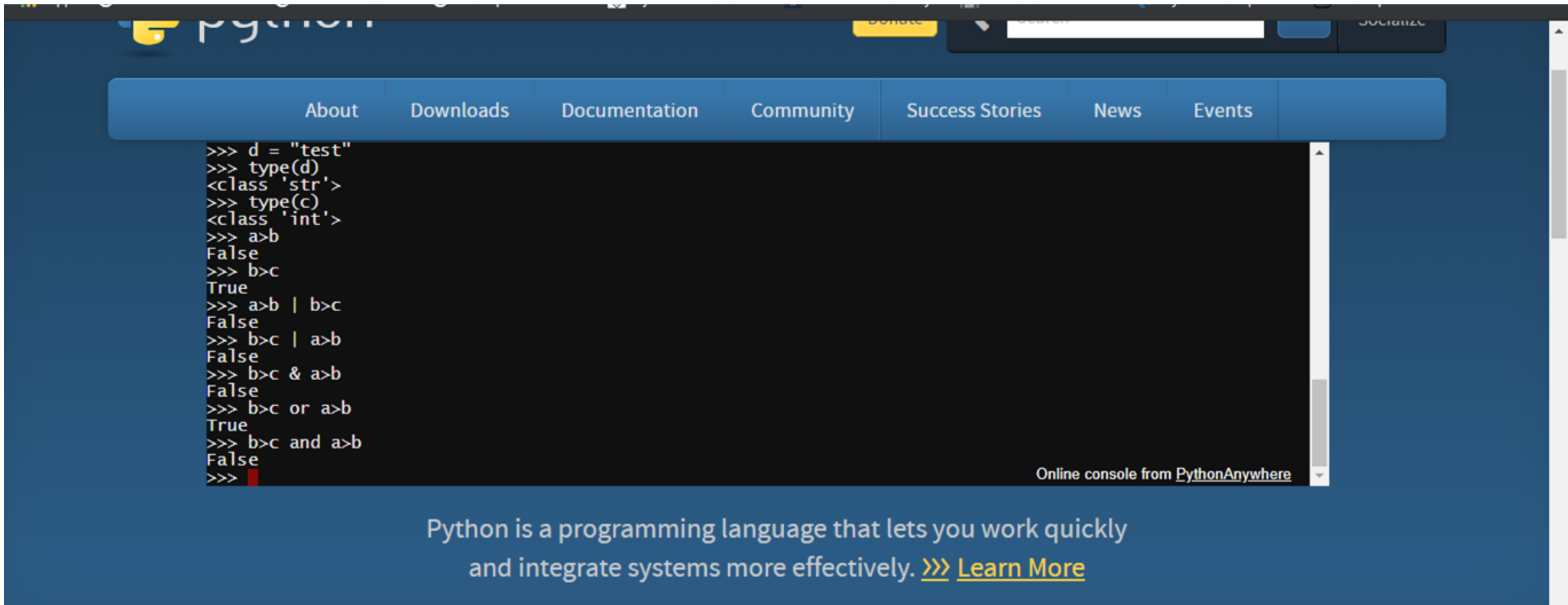


The screenshot shows the Python.org homepage. At the top, there's a navigation bar with links: About, Downloads, Documentation, Community, Success Stories, News, and Events. Below this is a large dark blue area containing a Python REPL console. The console shows a series of commands and their outputs, including a traceback for a `TypeError` when trying to concatenate a string and an integer. The commands and outputs are as follows:

```
>>> print("hellow world!!")
hellow world!!
>>> myVariable = 5;
>>> b=myVariable+8
>>> c=12
>>> print("my variable + 8="+b)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
>>> print("my variable + 8="+str(b))
my variable + 8=13
>>> b*=4
>>> print("arithmetic assignment operator = "+str(b))
arithmetic assignment operator = 52
>>> d = "test"
>>> type(d)
<class 'str'>
>>> type(c)
<class 'int'>
>>> a>b
```

At the bottom of the console area, it says "Online console from [PythonAnywhere](#)". Below the console, there's a text block that reads: "Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)".

<https://www.python.org/>



The screenshot shows the Python.org homepage. At the top, there's a navigation bar with links: About, Downloads, Documentation, Community, Success Stories, News, and Events. Below this is a large dark blue area containing a Python REPL console. The console shows the following code and output:

```
>>> d = "test"
>>> type(d)
<class 'str'>
>>> type(c)
<class 'int'>
>>> a>b
False
>>> b>c
True
>>> a>b | b>c
False
>>> b>c | a>b
False
>>> b>c & a>b
False
>>> b>c or a>b
True
>>> b>c and a>b
False
>>>
```

At the bottom right of the console, it says "Online console from [PythonAnywhere](#)". Below the console, there's a text block that reads: "Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)".

<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

- 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

Project explorer Editor - E:\GDPL\Flex\Day 1\Session 5\Codes\untitled0.py Outline Variable explorer

untitled0.py*

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri May 10 06:25:29 2019
4
5 @author: Shweta
6 """
7
8
```

untitled0.py

Name	Type	Size	Value
x	int	1	5
y	int	1	7

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

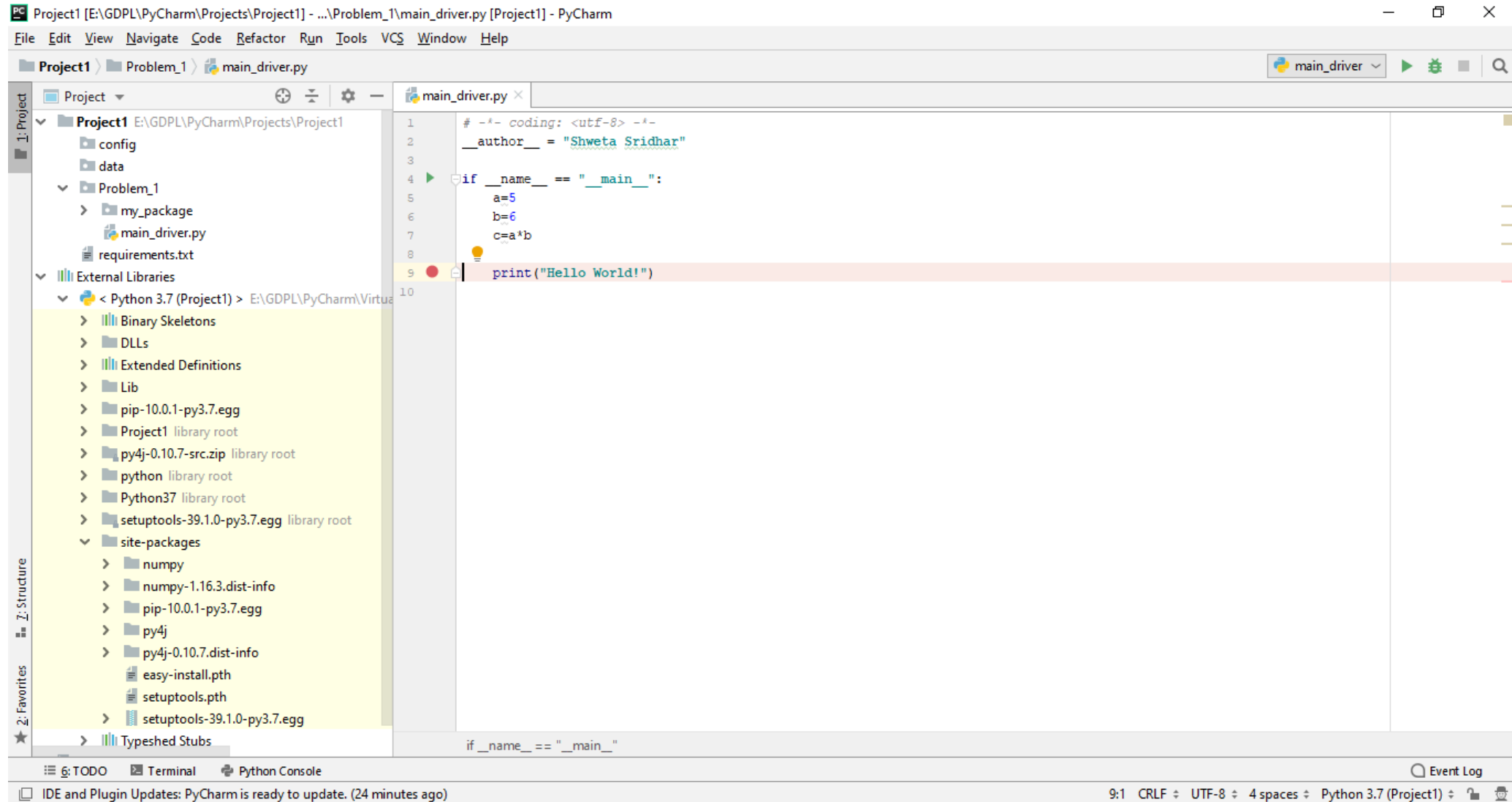
In [9]: print((x>y) or (x<=y))
True

In [10]: print((x>y) or (x<y))
True

In [11]:
```

Permissions: RW End-of-lines: CRLF Encodina: UTF-8 Line: 5 Column: 16 Memory: 72 %

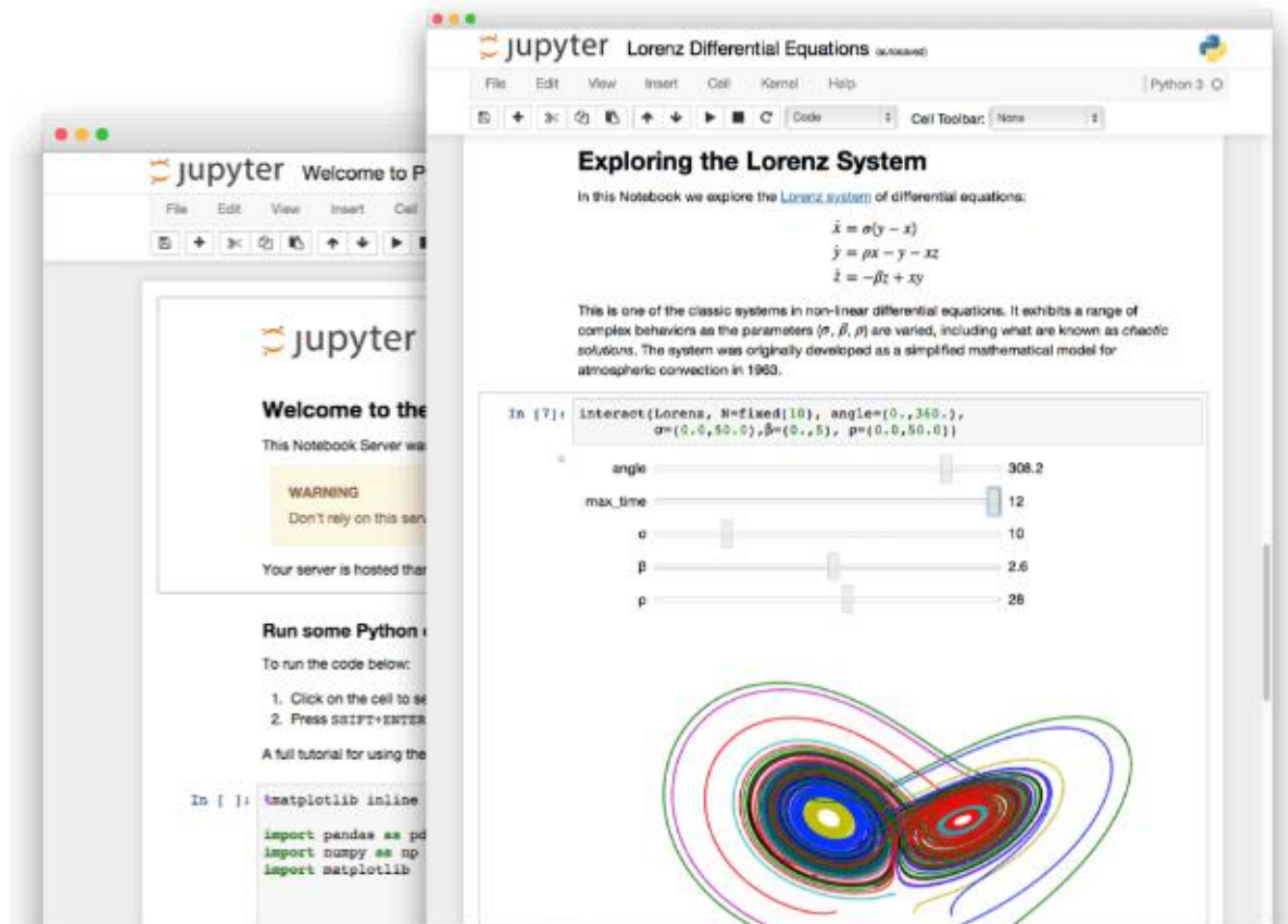
- Supported across Linux, Mac OS X and Windows platforms
- Available as community (free open source) and professional (paid) version
- Supports only Python
- Can be installed separately or through Anaconda distribution
- Features include
 - Code editor provides syntax and error highlighting
 - Code completion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



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

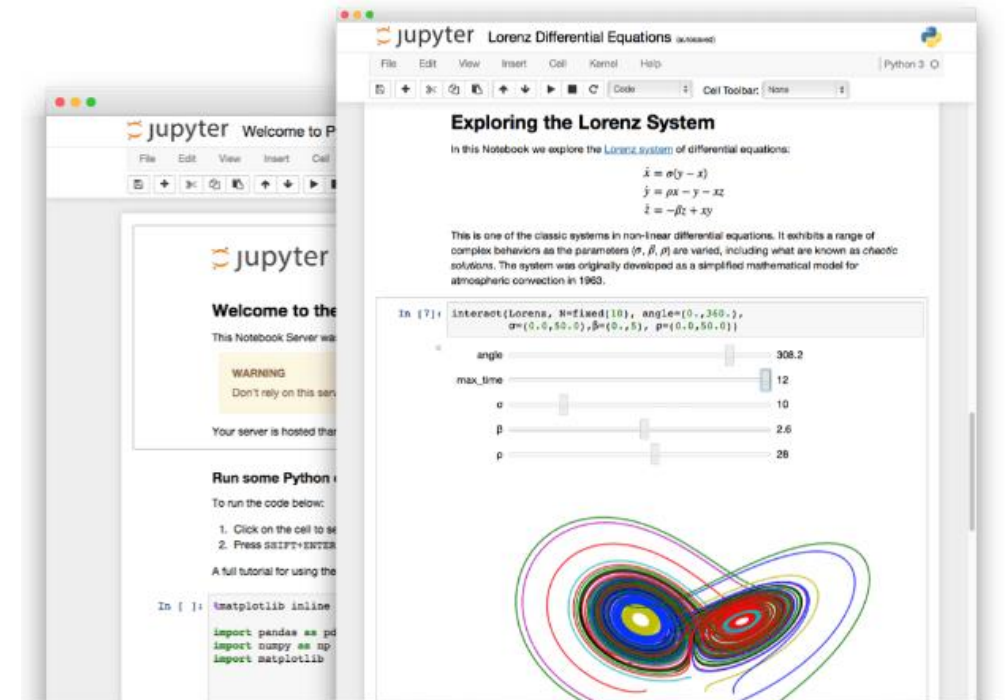
Jupyter Notebook



Source-<https://jupyter.org/>

Jupyter Notebook

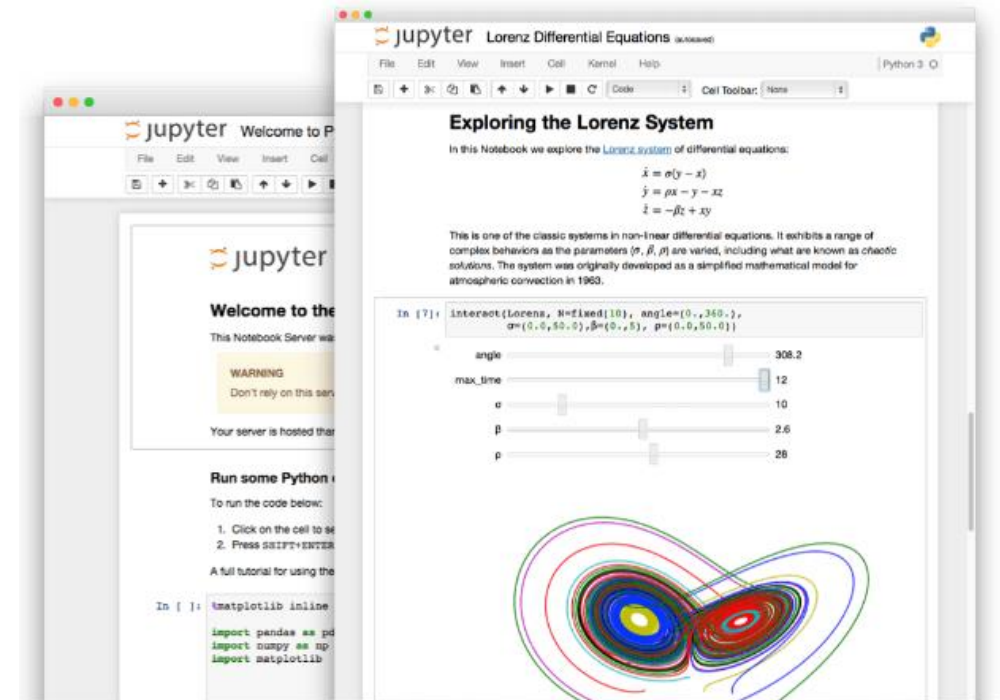
- Bundled with Anaconda distribution or can be installed separately
- Supports Julia, Python, R and Scala
- Consists of ordered collection of input and output cells that contain code, text, plots etc.



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":  
    mirror_mod.use_x = True  
    mirror_mod.use_y = False  
    mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
#selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
= ("Selected" + str(modifier_ob.name))  
mirror_ob.select = 0  
= bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exactly one mirror")
```

WILLIAM C. LEE

```
def mirror(modifier):  
    #add mirror to the selected  
    #object -mirror_x  
    mirror_ob = bpy.context.selected_objects[0]  
    mirror_mod = modifier
```

THANK YOU