



**BITS Pilani**

Pilani | Dubai | Goa | Hyderabad

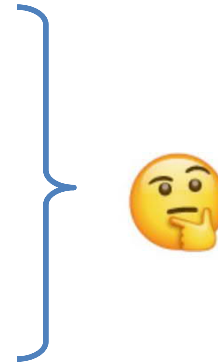
# Introduction to Python for Data Science

## DSECLPFDS

Prof. Parthasarathy  
BITS Pilani  
WILP Division

# Agenda for CS #1

- 1) Ground Rules
- 2) Introduction to *DSECLPFDS*
  - Motivation & Objective of *DSECLPFDS*
  - Courseware
  - Books & Evaluation components
  - Pedagogy for DSECLPFDS ?
- 3) Course Schedule
- 4) Getting started with Module 1



# Ground Rules!



- Mentally present – Observe!! Listen!!
- Keep your questions for the Q&A section / Discussion Forum
- Use the Discussion Forum in Canvas effectively
- Solve the exercises regularly!
- Go that “extra mile” 😊

$$1^{365} = 1$$

$$1.01^{365} = 37.8$$

# Motivation for this course ?



## Motivation

- As of now, Python is one of the most widely used programming languages in the Data Science field.
- Data Scientists just love Python! ❤️
- Python is easy to learn & has a great community for support!
- We would use Python for all the assignments / case-studies (For all the subjects in MTech DSE).

# Course Objectives




## What is this course about ?

- Introduce the fundamental programming concepts of Python
- Enable you to solve data problems using Python
- Act as a kick-start / bridge for participants of the MTech DSE programme who are *new* to Python.

## What is this course *not* about ?

- Comprehensive, in-depth discussion about Python programming.
- Comprehensive, in-depth discussion about data analysis using Python and related packages, libraries, and tools.

➤ Available at : <https://bits-pilani.instructure.com/courses/855/files/155743/download?wrap=1>


**BITS Pilani**  
Pilani | Dubai | Goa | Hyderabad

Introduction to Python for Data Science

### Table of Contents

- Course Objectives
- Modular Course Structure
- Text & References
- Additional Readings
- (Sample) Labs / Assignments

### Course Objectives

CO1	Introduce students with fundamental programming concepts of Python
CO2	Enable students to solve data problems using Python

### Text and References

T1	Charles Severance: Python for Everybody, Exploring Data in Python 3, Creative Commons, 2016
T2	Jake VanderPlas: Python Data Science Handbook, Essential Tools for Working with Data, O'Reilly Media, 2016
T3	Edouard Duchesnay: Statistics and Machine Learning in Python Release 0.2, 2018
T4	Wes McKinney: Python for Data Analysis, Agile Tools for Real World Data, O'Reilly Media, 2013

### Additional Reading

1. [Python 3.\\* documentation](#)
2. [Numpy Documentation](#)
3. [Pandas Documentation](#)
4. [Matplotlib documentation](#)
5. [seaborn: statistical data visualization documentation](#)
6. [Scikit-learn documentation](#)

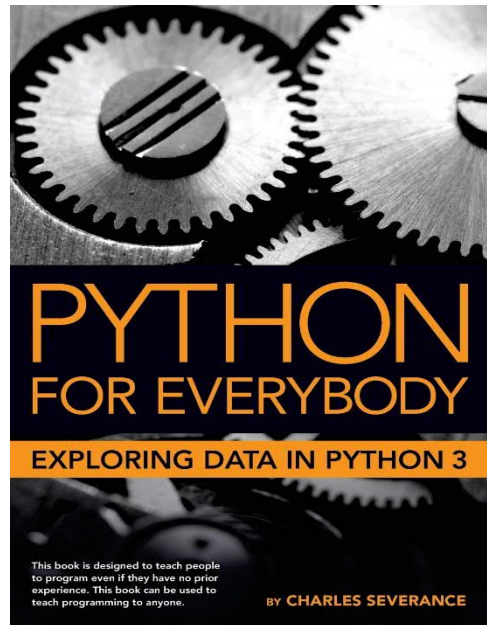
1 | Page

### Modular Content Structure

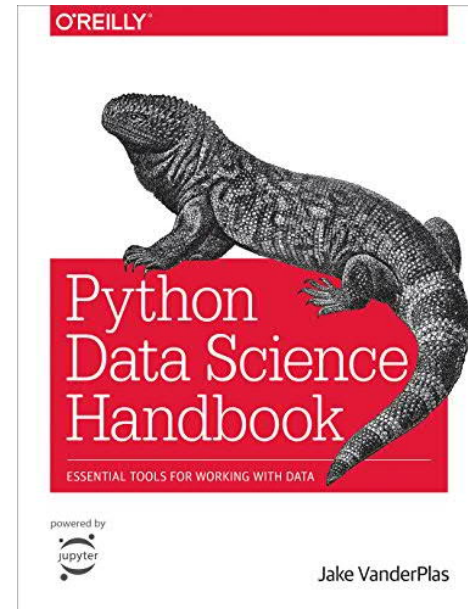
Session	Topics	Reference
<b>Saturday, April 17, 2021 – S1</b>		
<b>1</b>	<b>Python Basics</b>	
<b>1.1</b>	<b>Setting up Python Environments</b>	<b>Python Documentation</b>
1.1.1	Anaconda Distribution	
1.1.2	Spyder IDE	
1.1.3	Jupyter Notebooks	
1.1.4	Input / Output with Python	
<b>1.2</b>	<b>Getting familiarity with basic code constructs</b>	<b>T1 : Ch 2, Class Notes</b>
1.2.1	Package imports	
1.2.2	Data Types & Type Casting	
1.2.3	Variables, Expressions & Statements	
1.2.4	Object Orientation (Introduction)	
<b>Saturday, April 17, 2021 – S2</b>		
<b>2</b>	<b>Python Data Structures</b>	
<b>1.3</b>	<b>Immutable Data Structures</b>	<b>T1 : Ch 6, 10, Class Notes</b>
1.3.1	Immutable Data Structures	
1.3.2	Strings	
1.3.3	Operations on String	
1.3.4	Familiarity with Tuples	
	<b>Python Data Structures</b>	
<b>2.1</b>	<b>Mutable Data Structures</b>	<b>T1 : Ch 8, 9, Class Notes</b>
2.1.1	List	
2.1.2	List operations	
2.1.3	Familiarity with Sets	
2.1.4	Dictionary operations	
	<b>Python Programming Constructs</b>	
<b>2.2</b>	<b>Expressions, Operations, and Decision Structures</b>	<b>T1 : Ch 2, 3, Class Notes</b>
2.2.1	Boolean Expressions and Logical Operators	
2.2.2	Conditional and Alternative execution	
2.2.3	Chained and Nested execution	
2.2.4	Catching Exceptions with try and except	
<b>Sunday, April 18, 2021 – S1</b>		
<b>2.3</b>	<b>Iterative Executions</b>	<b>T1 : Ch 5, Class Notes</b>
2.3.1	While loops	
2.3.2	Infinite loops, break, continue	
2.3.3	For loops	
2.3.4	Loop patterns	

2 | Page

# Text Books



[Charles Severance: Python for Everybody, Exploring Data in Python 3](#)



[Jake VanderPlas: Python Data Science Handbook](#)

eBooks of both are made available in Canvas

*Note: These are the prescribed ones. Please feel free to explore any Python materials that suits you.*

# Evaluation Components



- This course is **NOT evaluated** !!
- You will **not** have any exams for this course 😊
- Nevertheless, there would be some exercises for you to try and hone your skills.



# Pedagogy for this Course



## Step 01: Class Session

- **We** learn Fundamentals !
- Look at few examples for each concept.

## Step 02: Explore

- **You** explore the additional notebooks. Get your hands dirty with Python
- Practice more examples for each concept.

## Step 03: Doubts

- Put your queries in Discussion Forum.
- Peers and TA to answer ...

**Non-Beginners:** You can directly start with Step 02 and also use this phase for additional learning which might help in future ... You play an important role in Step 3 as well in answering your peer's queries.

# Course Schedule



Today – S1 (9AM)	Today – S2 (11:30AM)	Tomorrow – S3 (9AM)
<ul style="list-style-type: none"> <li>○ Motivation &amp; Agenda</li> <li>○ Python Basics</li> <li>○ Setting up Python Environment</li> <li>○ Getting familiarity with basic code constructs</li> <li>○ Introduction to Object Orientation (This is more for self exploration as its not too widely used in DSE)</li> </ul>	<ul style="list-style-type: none"> <li>○ Python Data Structures</li> <li>○ Immutable Data Structures</li> <li>○ Mutable Data Structures</li> <li>○ Expressions, Operations &amp; Decision Structures</li> </ul>	<ul style="list-style-type: none"> <li>○ Iterative Constructs</li> <li>○ Functions</li> <li>○ Files</li> </ul>
(Sat) 24/4/2021 – S4 (9AM)	(Sat) 24/4/2021 – S5 (11AM)	(Sun)25/4/2021–S6(9AM)
<ul style="list-style-type: none"> <li>○ SciPy Ecosystem</li> <li>○ NumPy</li> <li>○ Pandas Basics</li> </ul>	<ul style="list-style-type: none"> <li>○ Data Exploration with Pandas</li> <li>○ Visualization with Matplotlib</li> </ul>	<ul style="list-style-type: none"> <li>○ Visualization with Seaborn</li> <li>○ Introduction to ML / scikit-learn</li> </ul>

# Program & Programming Language



## Computer Program

- Set of instructions that perform a specific task executed by computer
- Required by computer to function
- Written by programmer using programming languages
  - Like C, C++, Java, Python etc.
- Executed with compiler and interpreter

# Python as a Programming Language

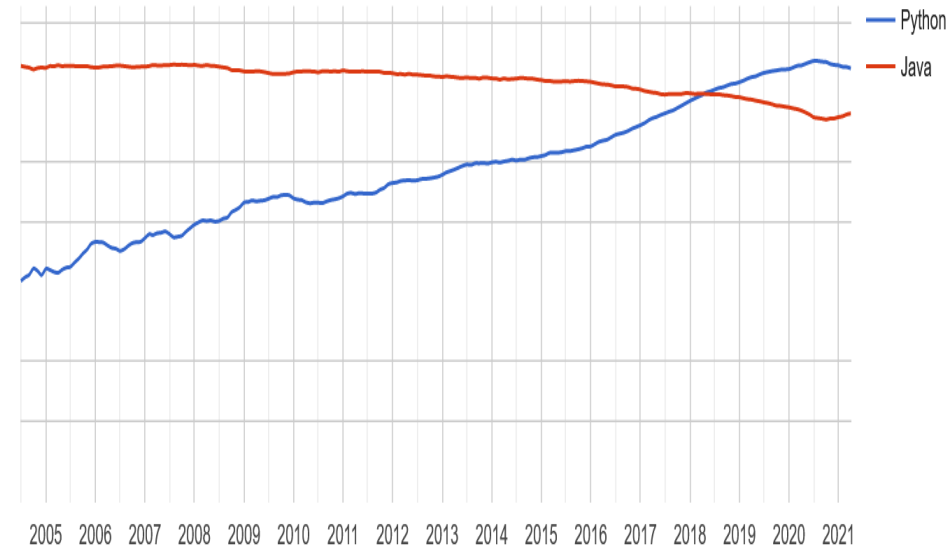


## Why Python ?

Worldwide, Apr 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	29.5 %	-1.0 %
2		Java	17.51 %	-0.6 %
3		JavaScript	8.19 %	+0.2 %
4		C#	7.05 %	-0.2 %
5	↑	C/C++	6.73 %	+1.0 %
6	↓	PHP	6.23 %	+0.0 %
7		R	3.86 %	+0.0 %

PYPL Popularity of Programming Language



Worldwide, Python is the most popular language ...

Source : <http://pypl.github.io/PYPL.html>

# Python as a Programming Language



## Python

- Designed by Guido van Rossum around 1990
- Not just a scripting language
- Easy to learn, read, use
- Extensible (add new modules)
- Highly readable
- Latest Version 3.9
- Most fond of language for Data Scientists

## Touchy Feel Properties

- Open Source
  - copyrighted but use not restricted
  - owned by independent non-profit, PSF
- Mature (29 years old)
- Supportive user community
  - plenty of good books, too
  - Active user community
- Simple design, easy to learn
  - reads like “pseudo-code”
  - Suitable as first language
  - Suitable as last language :-)  
(Hopefully)

# Python Applications



## Use Python for...

**Web Development:** Django , Pyramid , Bottle , Tornado , Flask , web2py

**GUI Development:** tkinter , PyGObject , PyQt , PySide , Kivy , wxPython

**Scientific and Numeric:** SciPy , Pandas , IPython

**Software Development:** Buildbot , Trac , Roundup

**System Administration:** Ansible , Salt , OpenStack

# Python Ecosystem



## Components of Python World :

- Core Python
- Distributions
- Frameworks / IDEs
- Third party Libraries

### Core Python

- Programming Language itself
- Some standard modules are available
- Other packages needs to be explicitly installed

### Python Distribution

- Python + packages
- Majority of packages, libraries are already available
- Package management is simplified
  - Anaconda from Continuum Analytics
  - IPython and its IPyKit variant

# Python Ecosystem



## Frameworks / IDEs

- Use frameworks to create code and develop applications
- Provides a defined structure to the developers so that they can focus on the core logic of the application rather than on other elements
- Python web framework
  - ✓ Django
  - ✓ Web2py
  - ✓ Flask
- Python IDEs
  - ✓ IDLE
  - ✓ PyCharm
  - ✓ Spyder
  - ✓ Jupyter Notebooks

## Third party Libraries

- Makes life of developers very simple
- Just need to know the right library to carry out a task
  - NumPy
  - Scipy
  - Pandas
  - Matplotlib
  - Seaborn
  - Bokeh
  - ScikitLearn
  - And List goes on ...



# Python Installation



## Three Ways :

- Install Python directly
  - Install the Python language with installer
  - Need to install other packages explicitly using pip install
  - <https://www.python.org/downloads/>
- Use Python distribution
  - The open-source Anaconda Distribution is the easiest way to perform Python coding
  - Works on Linux, Windows, and Mac OS X
  - <https://www.anaconda.com/distribution/>
- Use Cloud based services
  - The simplest of all but needs internet connectivity to use
  - Microsoft Azure Notebooks
  - Google Collab



*Post your queries in the Discussion Forum!!*

# Feedback

---

😊 👍 : 5

😏 🙅 : 3

😞 👎 : 1

Thank You for your  
time & attention !

Contact : [parthasarathypd@wilp.bits-pilani.ac.in](mailto:parthasarathypd@wilp.bits-pilani.ac.in)