Introduction to Python for Data Science

DSECLPFDS / AIMLCPFDS

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Parthasarathy

Agenda for CS #1

1) Ground Rules
2) Introduction to *DSECLPFDS / AIMLCPFDS*
    - Motivation & Objective
    - Courseware, Canvas Walkthrough
    - Books & Evaluation components
    - Pedagogy for this course ?
3) Course Schedule
4) Getting started with Module 1
5) Q&A
6) Feedback

Ground Rules!

- Mentally present – Observe!! Listen!!
- Keep your questions for the Q&A section / Discussion Forum
- Use the Discussion Forum in Canvas effectively
- Do not post unrelated messages in MS Teams. React using the "Like" feature if your intended message has already been posted.
- Solve the exercises regularly!
- Go that "extra mile" ☺

$$1^{365} = 1$$

$$1.01^{365} = 37.8$$

Motivation for this course ?

innovate achieve lead

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 / MTech AIML).

Course Objectives

innovate achieve lead

What is this course about ?

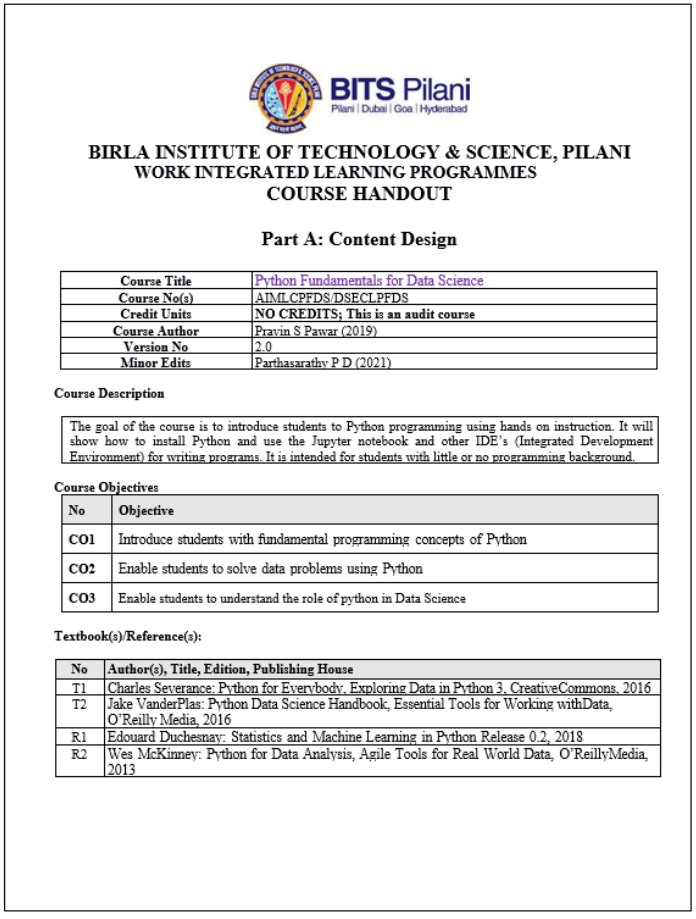
- Introduce the fundamental programming concepts of Python
- Enable you to solve data problems using Python
- Act as a kick-start / bridge for participants of the MTech DSE/AIML programme who are ***new*** to Python.

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.

Courseware

- Available on canvas (Home page).



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
WORK INTEGRATED LEARNING PROGRAMMES
COURSE HANDOUT

Part A: Content Design

Course Title	Python Fundamentals for Data Science
Course No(s)	AIML CPFDS/DSECL PFDS
Credit Units	**NO CREDITS; This is an audit course**
Course Author	Pravin S Pawar (2019)
Version No	2.0
Minor Edits	Parthasarathy P D (2021)

Course Description

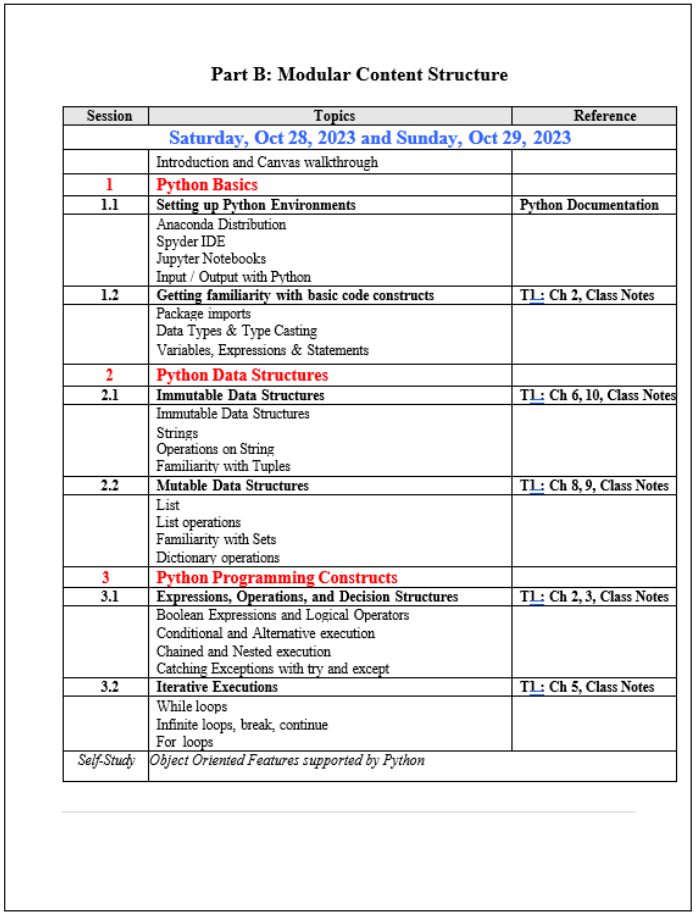
The goal of the course is to introduce students to Python programming using hands on instruction. It will show how to install Python and use the Jupyter notebook and other IDE's (Integrated Development Environment) for writing programs. It is intended for students with little or no programming background.

Course Objectives

No	Objective
CO1	Introduce students with fundamental programming concepts of Python
CO2	Enable students to solve data problems using Python
CO3	Enable students to understand the role of python in Data Science

Textbook(s)/Reference(s):

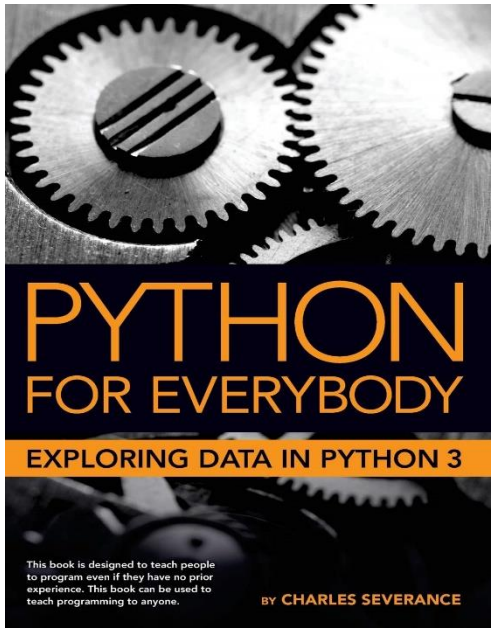
No	Author(s), Title, Edition, Publishing House
T1	Charles Severance: Python for Everybody, Exploring Data in Python 3, CreativeCommons, 2016
T2	Jake VanderPlas: Python Data Science Handbook, Essential Tools for Working withData, O'Reilly Media, 2016
R1	Edouard Duchesnay: Statistics and Machine Learning in Python Release 0.2, 2018
R2	Wes McKinney: Python for Data Analysis, Agile Tools for Real World Data, O'ReillyMedia, 2013



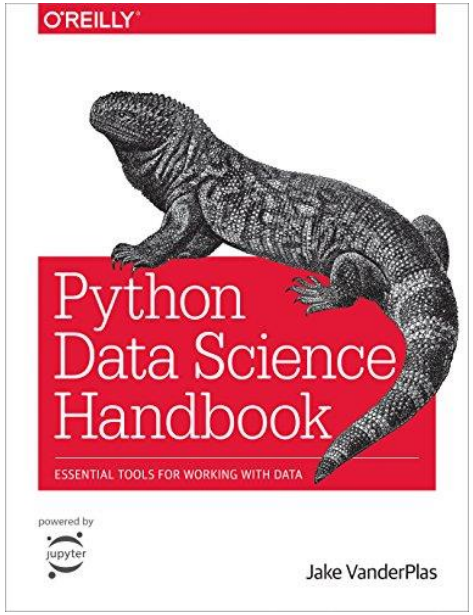
Part B: Modular Content Structure

Session	Topics	Reference
	Saturday, Oct 28, 2023 and Sunday, Oct 29, 2023	
	Introduction and Canvas walkthrough	
1	**Python Basics**	
1.1	**Setting up Python Environments**	**Python Documentation**
	Anaconda Distribution Spyder IDE Jupyter Notebooks Input / Output with Python	
1.2	**Getting familiarity with basic code constructs**	**T1 : Ch 2, Class Notes**
	Package imports Data Types & Type Casting Variables, Expressions & Statements	
2	**Python Data Structures**	
2.1	**Immutable Data Structures**	**T1 : Ch 6, 10, Class Notes**
	Immutable Data Structures Strings Operations on String Familiarity with Tuples	
2.2	**Mutable Data Structures**	**T1 : Ch 8, 9, Class Notes**
	List List operations Familiarity with Sets Dictionary operations	
3	**Python Programming Constructs**	
3.1	**Expressions, Operations, and Decision Structures**	**T1 : Ch 2, 3, Class Notes**
	Boolean Expressions and Logical Operators Conditional and Alternative execution Chained and Nested execution Catching Exceptions with try and except	
3.2	**Iterative Executions**	**T1 : Ch 5, Class Notes**
	While loops Infinite loops, break, continue For loops	
Self-Study	*Object Oriented Features supported by Python*	

Text Books



Charles Severance: Python for Everybody, Exploring Data in Python 3



Jake VanderPlas: Python Data Science Handbook

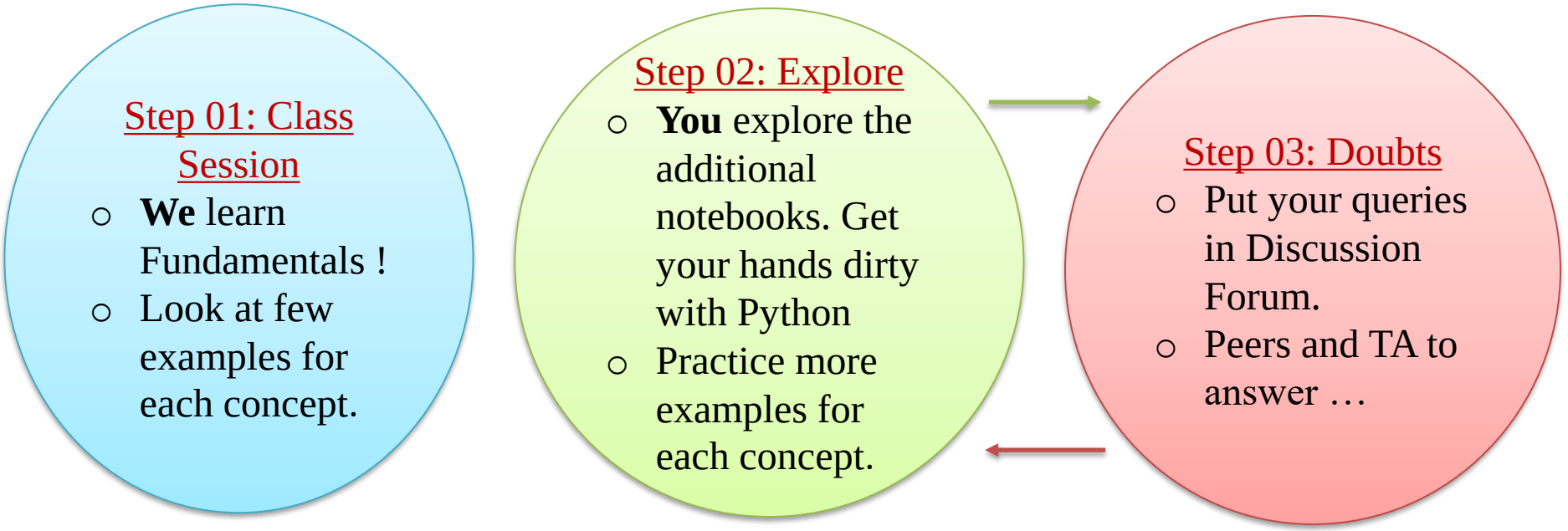
eBooks of both are made available in Canvas (Course → Files → Books).
Do not publish these on the Internet. For our use, we have got permission from the authors to use the eBook.

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: **[No need to submit].**
  - These will be on Canvas so you get accustomed to the Canvas LMS.
  - There will be “Assignments”
  - There will be “Quizzes”

Pedagogy for this Course



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 (2PM)	Today – S2 (After a short break)	29th Oct – S3 (PM)
○ Motivation & Agenda ○ Python Basics ○ Setting up Python Environment	○ Getting familiarity with basic code constructs. ○ Python Data Structures ○ Immutable Data Structures	○ Mutable Data Structures ○ Expressions, Operations & Decision Structures ○ Iterative Constructs
4th Nov (Sat) – S4 (2PM)	**4th Nov (Sat) – S5 (After a short break)**	**5th Nov (Sun)– S6 (9AM)**
○ Functions ○ Files ○ SciPy Ecosystem	○ NumPy ○ Pandas ○ Data Exploration with Pandas	○ Visualization with Matplotlib ○ Visualization with Seaborn ○ Way ahead

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

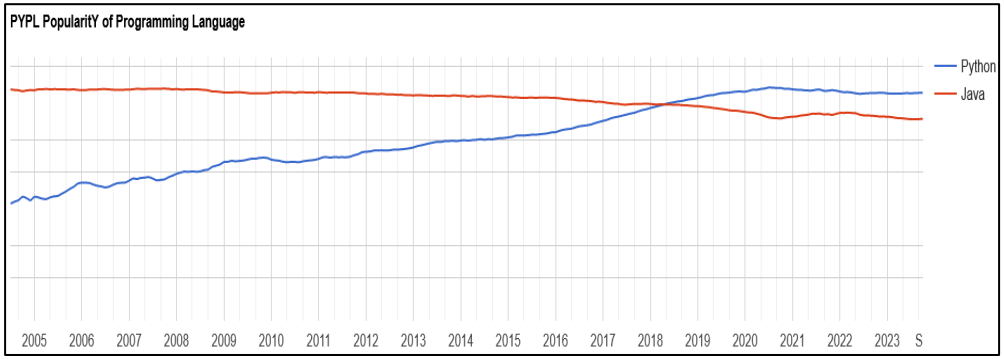
Python as a Programming Language

innovate achieve lead

Why Python ?

Worldwide, Oct 2023 :

Rank	Change	Language	Share	1-year trend
1		Python	28.05 %	+0.1 %
2		Java	15.88 %	-1.0 %
3		JavaScript	9.27 %	-0.3 %
4		C#	6.79 %	-0.2 %
5		C/C++	6.59 %	+0.3 %
6		PHP	4.86 %	-0.4 %
7		R	4.45 %	+0.4 %
8		TypeScript	2.93 %	+0.1 %



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 3.11.x
- Most fond of language for Data Scientists

Touchy Feel Properties

- Open Source
  - copyrighted but use not restricted
  - owned by independent non-profit, PSF
- Mature (30+ 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

innovate achieve lead

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://docs.anaconda.com/anaconda/install/windows/
- Use Cloud based services
  - The simplest of all but needs internet connectivity to use
  - Microsoft Azure Notebooks
  - Google Collab

Integrated Development Environments (IDE) for Python

Common IDE's:

Our Favourite (For MTech Programme):

- **.py** is a regular python file. It's plain text and contains just your code.
- **.ipynb** is a python notebook and it contains the notebook code, the execution results and other internal settings in a specific format.

Input / Output with Python

- print() can be used to output a message
- input() can be used to enter an input to the python program.
- # can be used to provide comments.
- “”” (triple quotes) can be used to write documentation.

Demo:

- Let’s see how to launch Jupyter Notebook
- See the basics of Notebook
- Practice some I/O statements and comments.

Basic Code Constructs

innovate achieve lead

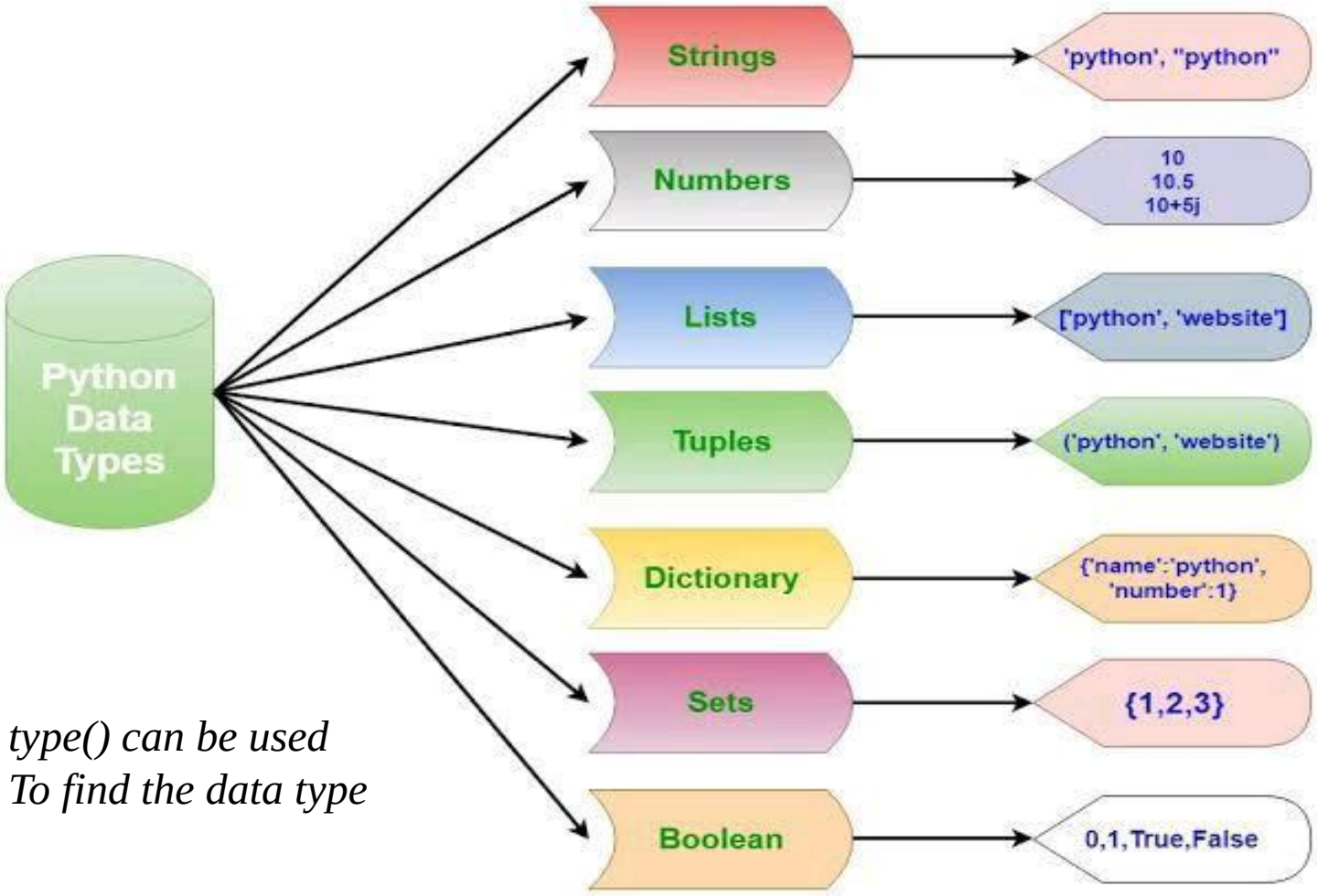
Imports:

- Import in Python is similar to #include in C/C++. Python modules can get access to code from another module by importing the file/function using import.
- Ex: import math
- print(math.pi)

Variable

- A Python variable is a reserved memory location to store values. In other words, variables are containers for storing data values.
- *Python has no command for declaring a variable.*
- A variable is created the moment you first assign a value to it.
- Ex: a = 100

Data Types in Python



type() can be used
To find the data type

Data Types in Python

Name	Type	Description
Integers	int	Whole numbers, such as: 3 300 200
Floating point	float	Numbers with a decimal point: 2.3 4.6 100.0
Strings	str	Ordered sequence of characters: "hello" 'Sammy' "2000" "楽しい"
Lists	list	Ordered sequence of objects: [10,"hello",200.3]
Dictionaries	dict	Unordered Key:Value pairs: {"mykey" : "value" , "name" : "Frankie"}
Tuples	tup	Ordered immutable sequence of objects: (10,"hello",200.3)
Sets	set	Unordered collection of unique objects: {"a","b"}
Booleans	bool	Logical value indicating **True** or **False**

type() can be used to find the data type

Post your queries in the Discussion Forum!!

Feedback

 : 5

 : 3

 : 1

Thank You for your time & attention !

Contact : parthasarathypd@wilp.bits-pilani.ac.in