**COURSE TITLE : TECHNICAL SKILLING (PFSDD)**

**[Python Full Stack Development with Django]**

**COURSE CODE** **: 21TS2101AA**

**Python Introduction :**

---------------------------------------

* **Python** is a general-purpose object-oriented, and high-level programming language.
* It is not intended to work in a particular area, it can be used with web, enterprise, 3D CAD, etc.
* Current version of python is 3.10.0
* Python File Format [.py]
* https://www.python.org/downloads/
* https://pypi.org/ (python external libraries or repositories)

**Features:**

---------------

-> Easy to use and Learn

-> Interpreted Language

-> Object-Oriented Language

-> Open-Source Language

-> Supports Wide Range of Libraries and Frameworks [You can Create your own libraries]

-> GUI Programming Support.

**Various Areas Where Python Used:**

---------------------------------------------------

Data Science

Data Mining

Machine Learning

Mobile Applications

Software Development

Desktop Applications

Web Applications

Enterprise Applications

Artificial Intelligence

3D CAD Applications

Computer Vision

Image Processing Applications

Speech Recognitions

**Python Basic Syntax:**

-----------------------------

-> There is no use of curly braces or semicolon in Python.

-> But Python uses the indentation to define a block of code.

-> Indentation is nothing but adding whitespace before the statement when it is needed.

**Applications Built Using Python**

-----------------------------------

-> Youtube

-> Instagram

-> Netflix

-> Linkedin

-> Mozilla Firefox

-> Dropbox

-> Quora

**Python Popular Frameworks and Libraries:**

-------------------------------------------------------------

It has wide range of libraries and frameworks.

Web development (Server-side) - Django, Flask, Pyramid, CherryPy

GUIs based applications - Tk, PyGTK, PyQt, PyJs, etc.

Machine Learning - TensorFlow, PyTorch, Scikit-learn, Matplotlib.

Mathematics - Numpy, Pandas, etc.

**Limitations Of Python:**

---------------------------------------

-> It is slower in execution than compiler-based languages.

-> Execution of C Program is faster than Python.

-> Not good for gaming.

-> Huge libraries and no dedicated support are available.

**Important Libraries In Our Syllabus:-**

-----------------------------------------------------------------

-> datetime

-> os

-> csv

-> re

-> json

-> sqlite3

-> requests, django, flask.

**Data Types In Python:-**

---------------------------------------------------------------

-> int

-> float

-> complex

-> bool

-> str

-> list

-> tuple

-> set

-> dict

-> class

**Important Basic "Functions":**

--------------------------------------------

-> len() returns length of the collection.

-> type() to know datatype of a variable.

-> dir() to know the functionalities of individual datatype.

-> print() to display a msg.

-> input() to read input from keyboard/input device.

-> help() to know the functionality/description of the function.

**Comment Line in Python:**

# - is used for single line command

'''text ''' - multiline comment/ doc string

**Basic Commands in Command Prompt**

**python --version (to get python version)**

**where python (to get where python installed in ur local machine)**

**py or python (to enter into Python shell)**

**Python print() Function**

--------------------------------------------

The print() function displays the given object to the standard output device (screen) or to the text stream file.

Syntax : print(variable name)

**Example:**

---------

a = 10

# Two objects are passed in print() function

print("a =", a)

b = a

# Three objects are passed in print function

print('a =', a, '= b')

**Taking Input to the User:**

----------------------------------------------

**input() function** ---> used to take input from the user.

Example:

-----------------------

name = input("Enter a name of student:")

print("The student name is: ", name)

Example:

-----------------------

a = int(input("Enter first number: "))

b = int(input("Enter second number: "))

print(a+b)

>>> a=10

>>> print(a)

10

>>> type(a)

<class 'int'>

>>> cost = 45.75

>>> type(cost)

<class 'float'>

>>> name ="KLUniv"

>>> type(name)

<class 'str'>

>>> status=True

>>> type(status)

<class 'bool'>

>>> status=False

>>> type(status)

<class 'bool'>

>>> pancard ="aiupc5685k"

>>> type(pancard)

<class 'str'>

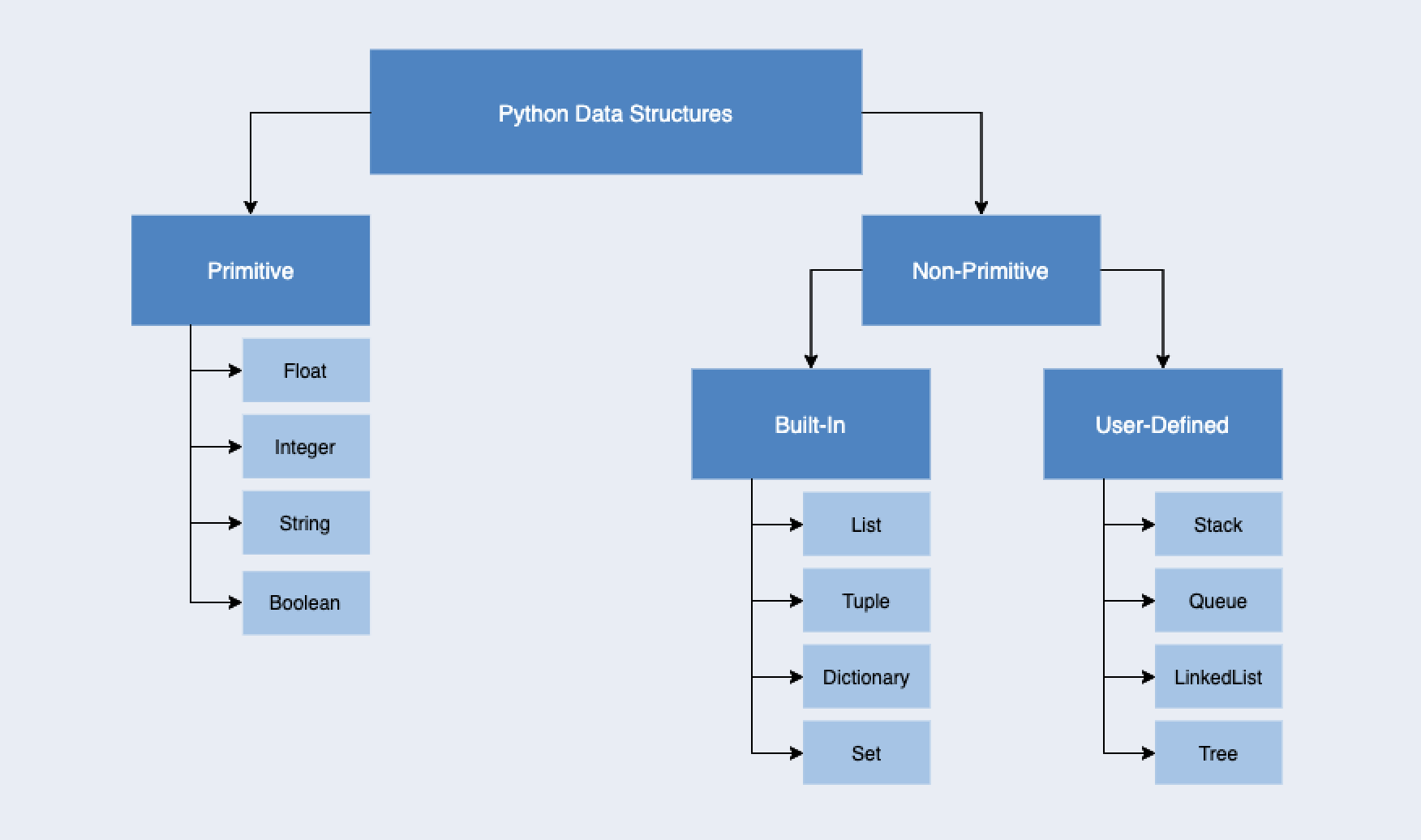
-->py is an interpreter based

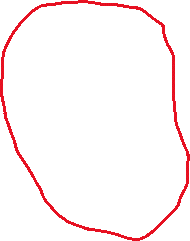
--> in py we don't have arrays concept unlike C, CPP, Java.

**Python Data Structures**

----------------------

-> Data structures are referred - they are used to store the data in organized way.





-> Python provides built-in data structures such as

list

tuple

set

dictionary



**Python List**

--------------------

-> Python list holds the ordered collection of items.

-> Python list is mutable -- i.e, we can modify after its creation.

-> The items of lists are enclosed within **the square bracket []** and separated by the comma.

Example:

>>> marks=[10,55,77,56,55,10] # Duplicates Allowed,

>>> type(marks)

<class 'list'>

>>> person =[123,456,"java","TN","AP",4500.78,"KL"] # Hetrogeneous type of values

>>> type(person)

<class 'list'>

**Python Tuple**

---------------------------

* Python Tuple is used to store the sequence of immutable Python objects.
* The items of lists are enclosed within the **bracket ( ) and** separated by the comma.
* The tuple is similar to lists since the value of the items stored in the list can be changed, whereas the tuple is immutable, and the value of the items stored in the tuple cannot be changed.

>>> course =("OS", "CN", "DS", "DBMS") #Duplicates Allowed

>>> type(course)

<class 'tuple'>

If we try to add new to the tuple, it will throw an error.

>>> cours[2] = "DAA"

Output: ERROR -TypeError: 'tuple' object does not support item assignment

**Dictionaries**

---------------------------

* It stores the data in the key-value pair format.
* In Dictionary each key is separated from its value by a colon (:)
* Keys must be a unique and value can be any type such as integer, list, tuple, etc.
* It is a mutable type; we can reassign after its creation.
* The empty curly braces {} are used to create empty dictionary

**Example:**

>>> states={'Andhra Pradesh':'Amaravati','Bihar':'Patna','Jharkhand':'Ranchi','Madhya Pradesh':'Bhopal','Telangana':'Hyderabad'}

**>>> print(states)**

{'Andhra Pradesh': 'Amaravati', 'Bihar': 'Patna', 'Jharkhand': 'Ranchi', 'Madhya Pradesh': 'Bhopal', 'Telangana': 'Hyderabad'}

>>>

**#upadate values**

**>>> states['Bihar']='chennai'**

>>> print(states)

{'Andhra Pradesh': 'Amaravati', 'Bihar': 'chennai', 'Jharkhand': 'Ranchi', 'Madhya Pradesh': 'Bhopal', 'Telangana': 'Hyderabad'}

**# remove a particular item**

>>>print(states.pop(‘Bhopal’)

>>>

**# remove an arbitrary item, return (key,value)**

print(states.popitem())

**# remove all items**

>>>states.clear()

**# delete the dictionary itself**

del states

**Python Sets**

------------

* A Python set is a collection of unordered elements.
* The elements in the set cannot be duplicates.
* The elements in the set are immutable (cannot be modified) but the set as a whole is mutable.
* There is no index attached to any element in a python set. So they do not support any indexing or slicing operation.

**Example:**

**#creating of set**

>>> Month = {"January", "February", "March", "April", "May", "June", "July"}

**>>> type(Month)**

<class 'set'>

>>> Subject={'ads','mp2','pfsd','May'}

>>> type(Subject)

<class 'set'>

**#adding of element**

>>> Subject.add('ccs')

>>> print(Subject)

{'ccs', 'pfsd', 'mp2', 'ads', 'May'}

**#Removing of elements**

>>> Month.discard('June')

**#Union of Sets**

>>> all=Month|Subject

>>> print(all)

{'February', 'ccs', 'March', 'pfsd', 'April', 'mp2', 'May', 'ads', 'January', 'July'}

>>>

**#Intersection of Sets**

>>> all=Month&Subject

>>> print(all)

{'May'}

>>>

**Python String**

---------------------

Python string is a sequence of characters.

is a collection of the characters surrounded by single quotes, double quotes, or triple quotes.

It can also define as collection of the Unicode characters.

Python doesn't support the character data-type.

A single character written as 'p' is treated as a string of length 1.

Stings are also immutable. We can't change after it is declared

**Example:**

----------------------------------------------------------------------------

# Creating string using double quotes

str1 = "Hi Python"

print(str1)

# Creating string using single quotes

str1 = 'Hi Python'

print(str1)

# Creating string using triple quotes

str1 = '''Hi Python'''

print(str1)

Output:

Hi Python

Hi Python

Hi Python