Python Workshop

By Deepak Rai

What you get from this session?

1. Python Introduction?
2. Python Project Creation
3. Python Classes, Interface & Packages
4. CRUD Operation
5. Flask using HTML, CSS, JS and Mysql
6. Tuple, Dictionary and List
7. What is Django?
8. Django Architecture
9. Django Project Creation
10. Django Site & Enviroment Setup

Trainer Intro

- Working Professional having 10+ Years of experience in Software Industry.

- One of the UiPath (Most Valuable Professional).
- Worked different region in world like UK, France, China, Poland etc.

Expertise: Software Testing, AWS, RPA Tools, Jira, Rally, Python, HTML, CSS, IPTV/STB

Expectations

What’s your expectation from this Training?

What is Python

- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.
- Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development.
- Python is relatively simple, so it’s easy to learn since it requires a unique syntax that focuses on readability.
- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python

Programming languages have been around for ages, and every decade sees the launch of a new language sweeping developers off their feet. Python is considered as one of the most popular and in-demand programming language. A recent Stack Overflow survey showed that Python has taken over languages such as Java, C, C++ and has made its way to the top:

- Easy to use.
- Platform independent
- Secured
- Easy to connect with DB
- Portable & Extensible
- Artificial Intelligence
- Computer Graphics
- Big Data
- Testing Framework

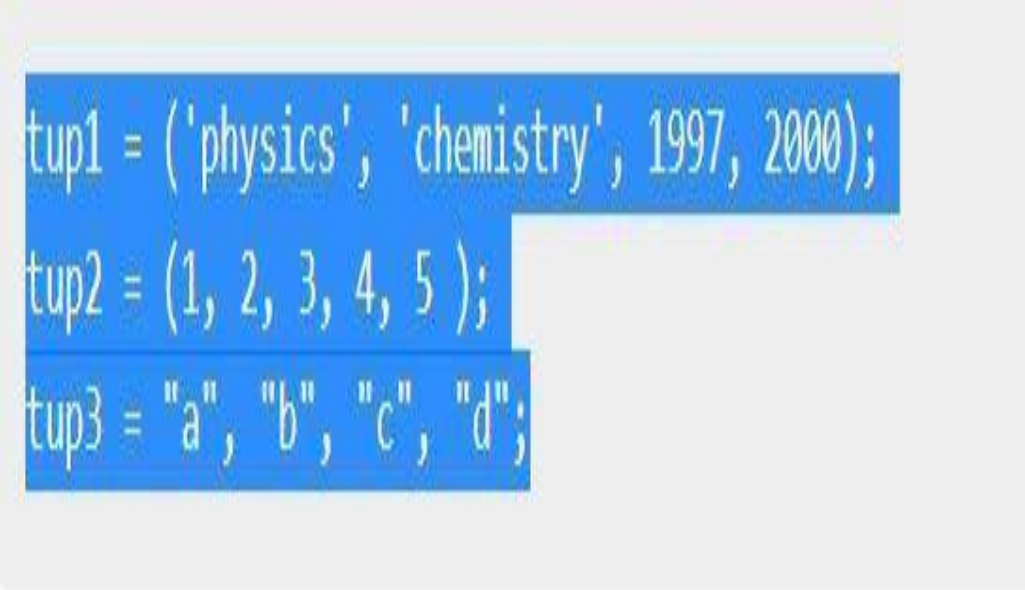
Python Data Types

Text Type:	`str`
Numeric Types:	`int`, `float`, `complex`
Sequence Types:	`list`, `tuple`, `range`
Mapping Type:	`dict`
Set Types:	`set`, `frozenset`
Boolean Type:	`bool`
Binary Types:	`bytes`, `bytearray`, `memoryview`

Tuple

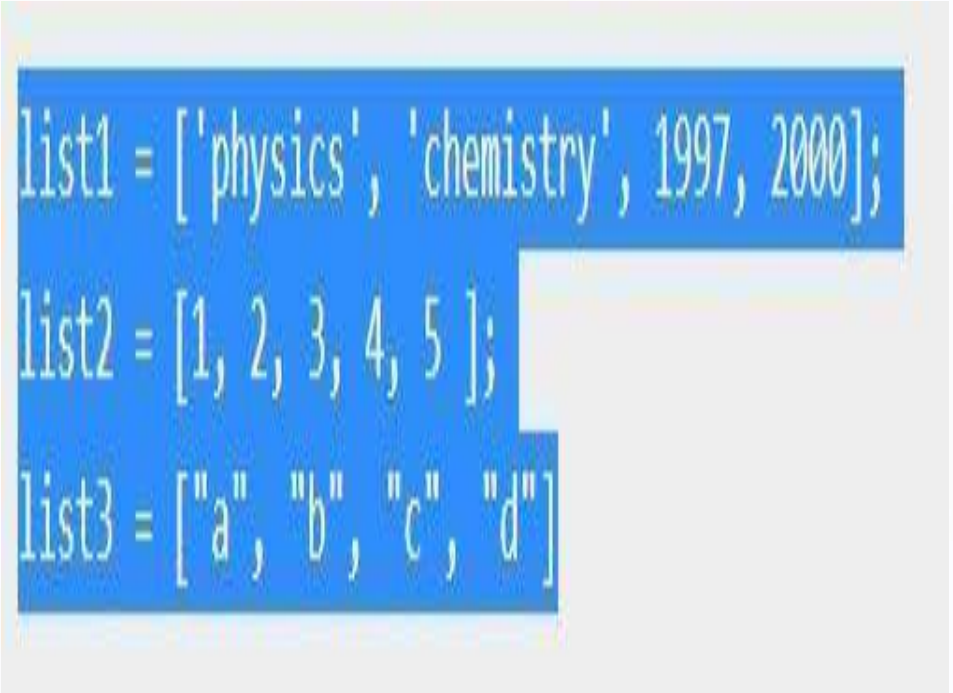
Tuple: Tuple is a collection of objects which ordered and immutable. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.

Creating a tuple is as simple as putting different comma-separated values. Optionally you can put these comma-separated values between parentheses also. For example –



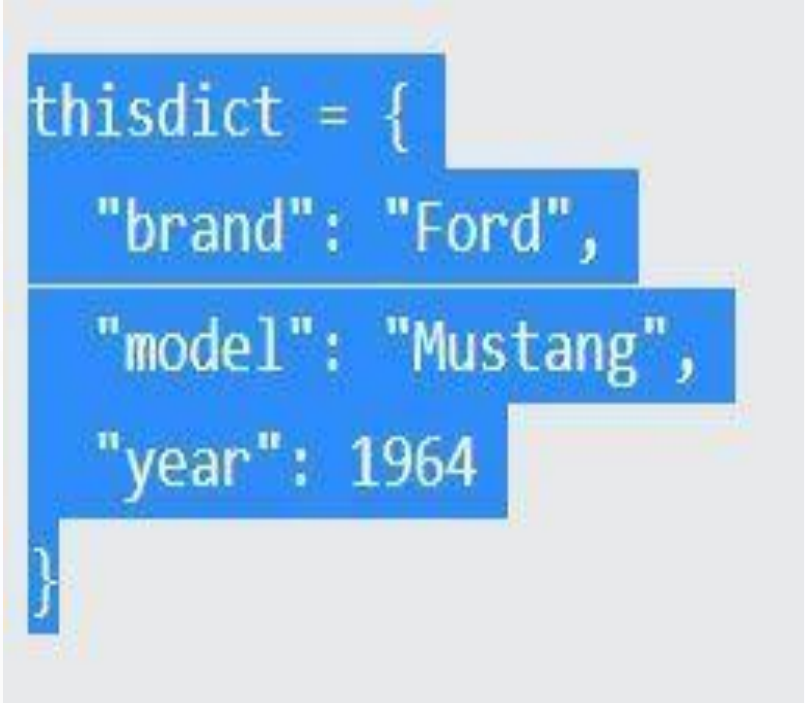
List

List: Lists are used to store multiple items in a single variable.
Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.
Lists are created using square brackets:



Dictionary

Dictionary: Each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this: {}. Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.



Classes and Objects

A class is a user-defined blueprint or prototype from which objects are created. Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by their class) for modifying their state.

An Object is an instance of a Class. A class is like a blueprint while an instance is a copy of the class with *actual values*. It's not an idea anymore, it's an actual dog, like a dog of breed pug who's seven years old. You can have many dogs to create many different instances, but without the class as a guide, you would be lost, not knowing what information is required.

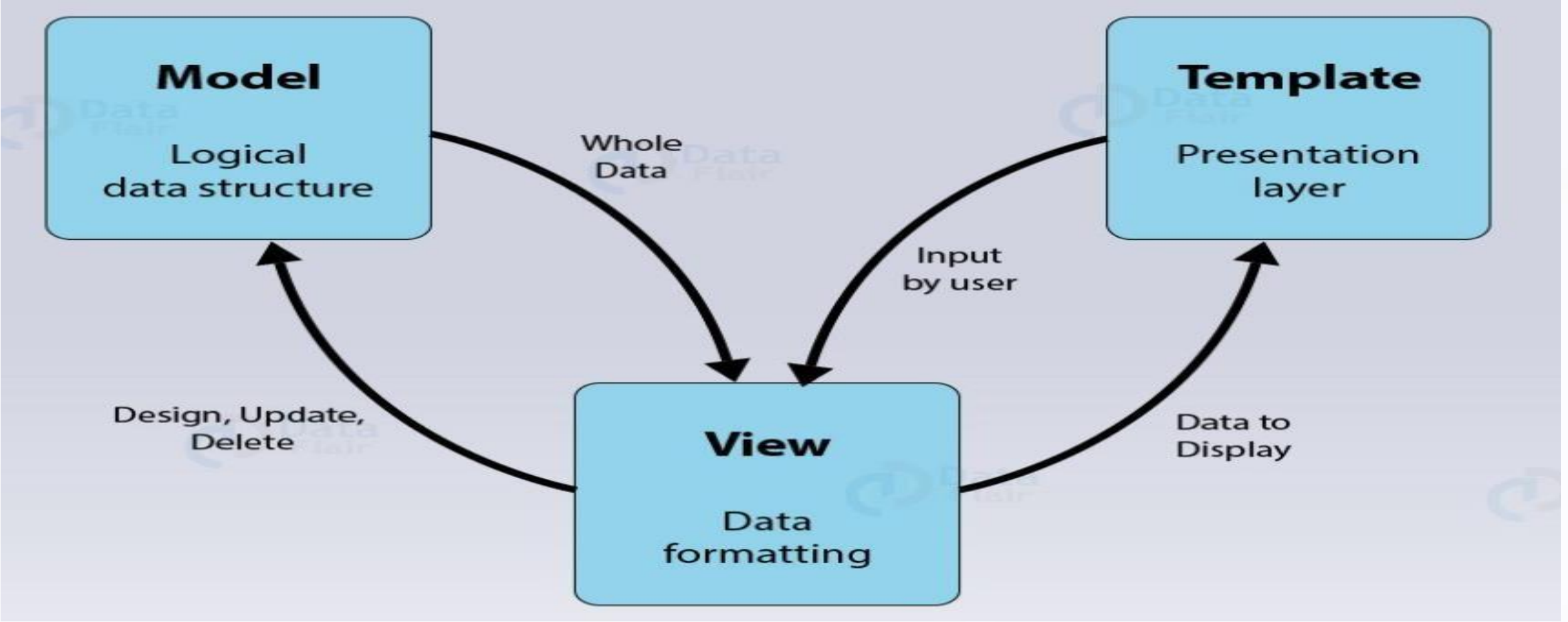
Django

- Django is a Python-based web framework which allows you to quickly create web application without all of the installation or dependency problems that you normally will find with other frameworks.

Why Django?

- Django is a rapid web development framework that can be used to develop fully fleshed web applications in a short period of time.
- It's very easy to switch database in Django framework.
- It has built-in admin interface which makes easy to work with it.
- Django is fully functional framework that requires nothing else.
- It has thousands of additional packages available.

Django MTVC


Model
Logical data structure
Template
Presentation layer
Whole Data
Input by user
Design, Update, Delete
View
Data formatting
Data to Display

DEMO

Deepak Rai

Thank You!