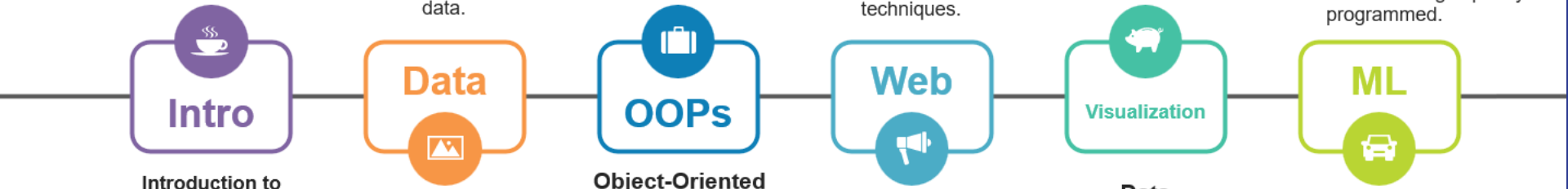
Python

Advance programming

Presented By
Jitendra Singh Tomar

Today's Agenda



Introduction to Python Programming

Python programming is a versatile and beginner-friendly language known for its simplicity, readability, and extensive libraries.

Data Manipulation

Performing various operations on datasets to extract, transform, clean, and analyze the data.

Object-Oriented Programming

OOP in Python allows you to structure your code by creating classes and objects to encapsulate data and behavior.

Web Scraping

Web scraping in Python is the process of extracting data from websites using programming techniques.

Data Visualization

Creating visual representations of data to gain insights and communicate findings effectively.

Machine Learning

is the process of developing and implementing algorithms that enable computers to learn from data and make predictions or decisions without being explicitly programmed.

Introduction to Python Programming

What we have to study?

- What is Python?
- Python syntax basics: variables, data types, operators
- Control flow statements: if-else, loops (for, while)
- Functions in Python
- Modules in Python

Introduction

- Python is a high-level programming language.
- Python emphasizes code readability and simplicity.
- Python supports multiple programming paradigms, including object-oriented and functional programming.
- Python is cross-platform, supporting Windows, macOS, and Linux.
- Python is widely used in various domains, such as web development, data analysis, machine learning, and scientific computing.
- Python uses indentation for code blocks.
- Python scripts use the ".py" file extension.

- Python is a high-level programming language.
- Python emphasizes code readability and simplicity.
- Python supports multiple programming paradigms, including object-oriented and functional programming.
- Python is cross-platform, supporting Windows, macOS, and Linux.
- Python is widely used in various domains, such as web development, data analysis, machine learning, and scientific computing.
- Python uses indentation for code blocks.
- Python scripts use the ".py" file extension.

Features

- Easy to learn and read
- Interpreted language
- Cross-platform compatibility
- Large standard library
- Third-party packages
- Object-oriented programming (OOP)
- Dynamic typing

Variables

- They are storage location with a name.
- Name=Value pair.
- Variable names are "case-sensitive".
- Underscores are allowed in variable names.
- Cannot use "+", "-".

```
>>> # Variables
>>>
>>> var1 = "Python is Awesome"
>>> print (var1)
Python is Awesome
>>>
>>>
```

Python basics

Data Types

- Numeric types
- Boolean type
- Strings
- Lists
- Tuples
- Dictionaries
- Sets

Operators

- Arithmetic Operators
- Comparison Operators
- Logical Operators
- Assignment Operators
- Membership Operators
- Identity Operators
- Bitwise Operators

Control flow statements

In Python, control flow statements allow you to control the execution flow of your code based on certain conditions or loops.

- Conditional Statements
  - if statement
  - if-elif-else statement
- Loops
  - for loop
  - while loop
  - break statement
  - continue statement
  - pass statement

Functions in Python

Functions are reusable blocks of code that perform a specific task.

Overview of functions in Python:

- Defining a Function
- Function Parameters
- Function Return Values
- Default Arguments
- Variable Number of Arguments
- Lambda Functions
- Scope

Modules in Python

- Modules are files containing Python code that define functions, classes, and variables that can be used in other Python programs.
- Modules provide a way to organize and reuse code, making it easier to manage and maintain large projects.
- Few more details regarding modules in Python:
  - Module Search Path
  - Module Namespace
  - Package
  - Importing from Packages
  - Importing All Items from a Module
  - Standard Library Modules
  - Third-Party Modules

Demo

Data Manipulation with Python

What we have to study?

- Working with lists, tuples, and dictionaries
- List comprehension and other iterable operations
- String manipulation and formatting
- File handling in Python
- Introduction to libraries like NumPy and Pandas

Lists

- Lists are versatile data structures that can hold different types of elements, such as integers, strings, or even other lists.
- Some common operations of list:
  - Creating a list
  - Accessing list elements
  - Modifying list elements
  - List length and membership
  - List concatenation and repetition
  - Iterating over a list
  - List sorting and reversing
  - List comprehension

Tuples

- Tuples are another type of data structure similar to lists, but with one key difference: tuples are immutable, meaning their elements cannot be modified once created.
- Tuples are defined using parentheses () or without any brackets.
- Some common operations of Tuples are:
    - Creating a tuple
    - Accessing tuple elements
    - Tuple unpacking
    - Tuple immutability
    - Tuple length and membership
    - Tuple concatenation and repetition
    - Iterating over a tuple
    - Tuple conversion

Dictionaries

- Dictionaries are another fundamental data structure that allow you to store and retrieve data using key-value pairs.
- Dictionaries are also known as associative arrays or hash maps in other programming languages.
- Some common operations of Dictionaries are:
    - Creating a dictionary
    - Accessing dictionary values
    - Modifying dictionary values
    - Dictionary length and membership
    - Removing key-value pairs
    - Iterating over a dictionary
    - Dictionary methods

String manipulation

- Some common string manipulation operations in Python:
  - Concatenating strings
  - Accessing individual characters
  - Slicing strings
  - String length
  - Changing case (upper to lower & vice-versa)
  - Finding substrings
  - Replacing substrings
  - Splitting and joining strings
  - Stripping whitespace
  - String formatting

File handling

- Opening a file
- Reading from a file
- Writing to a file
- Appending to a file
- Using with statement
- Reading file line by line
- Checking file existence

- File modes:
  - "r": Read mode
  - "w": Write mode
  - "a": Append mode
  - "x": Exclusive creation mode
  - "b": Binary mode
  - "t": Text mode

Introduction to libraries

- Libraries are collections of pre-written code that provide additional functionalities beyond what is available in the Python standard library.
- They can be installed and imported into your code, enabling you to leverage the functionalities they offer.
- Here is a list of some commonly used libraries in Python

  - NumPy
  - Pandas
  - Matplotlib
  - Scikit-learn
  - TensorFlow
  - PyTorch
  - Keras
  - SciPy
  - OpenCV
  - NLTK
  - Requests
  - Beautiful Soup
  - Flask
  - Django
  - SQLAlchemy
  - Celery
  - Pillow
  - Pygame
  - pytest
  - Seaborn

Libraries operations:

- Installing libraries
- Importing libraries
- Using library functions and classes
- Exploring library documentation
- Installing libraries with specific versions
- Virtual environments

Python NumPy

- NumPy (Numerical Python) is a powerful library in
- It is a fundamental library for scientific computing and data analysis in Python.
- NumPy is used for working with arrays.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- Arrays are very frequently used in data science, where speed and resources are very important.
- NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

Python NumPy operations:

- NumPy based operations:
  - Installation
  - Importing NumPy
  - NumPy Arrays
  - Array operations
  - Array Shape and Reshaping
  - Broadcasting
  - Random Number Generation

Python Pandas:

- Pandas is a popular open-source library for data manipulation and analysis in Python.
- Advantages:
  - Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
  - Fast and efficient for manipulating and analyzing data.
  - Data from different file objects can be loaded.
  - Flexible reshaping and pivoting of data sets
  - Provides time-series functionality.
  - Data set merging and joining.

Python Pandas:

Here are few key features of Pandas:

- DataFrame
- Series
- Data Import and Export
- Data Cleaning and Preparation
- Data Indexing and Selection
- Data Aggregation and Grouping
- Data Visualization
- Time Series Analysis
- Data Integration

Demo

Introduction to Object-Oriented Programming

What we have to study?

- Understanding the basics of OOP
- Classes and objects in Python
- Inheritance and polymorphism
- Encapsulation and data hiding
- Abstraction and interfaces

Object-oriented programming

- It is a programming paradigm that revolves around the concept of objects, which can contain both data (attributes) and functions (methods).
- Python is an object-oriented programming language that provides support for implementing OOP concepts.
- Here are some basic concepts of OOP in Python:
    - Classes and Objects
    - Encapsulation
    - Inheritance
    - Polymorphism
    - Abstraction

Classes and Objects

- A class is a blueprint or template for creating objects. It defines the attributes and methods that an object can have.
- An object is an instance of a class. It represents a specific entity based on the class definition.

```
# creating class
class MyClass:
    def method(self):
        print("This is a method.")

# Creating an instance of the class
my_instance = MyClass()

# Calling the method on the instance
my_instance.method()
```

Encapsulation

- Encapsulation is one of the fundamental principles of object-oriented programming (OOP) and refers to the bundling of data and methods together within a class, hiding the internal details of an object from the outside world.
- It allows for the control and protection of the data and behavior of an object.
- Encapsulation can be achieved through the use of access modifiers and property decorators:
    - Public members
    - Protected members
    - Private members

Abstraction

- Abstraction in Python is a fundamental concept of object-oriented programming (OOP) that involves simplifying complex systems by focusing on essential properties and behaviors while hiding unnecessary details.
- It allows you to create classes that represent real-world objects or concepts in a simplified manner.
- Abstraction helps in reducing programming efforts & reduce coding complexity.
- It is used to help unwanted details from the user.
- It allows focusing on the main concept.

Demo

Introduction to Web Scraping with Python

What we have to study?

- Understanding web scraping and its applications
- HTML structure and tags
- Introduction to libraries like Beautiful Soup and Requests
- Extracting data from web pages
- Handling pagination and navigating through websites

Understanding Web Scraping

- Web scraping refers to the process of extracting data from websites automatically.
- It is an automated method used to extract large amounts of data from websites. The data on the websites are unstructured.
- Web scraping helps collect these unstructured data and store it in a structured form.
- An overview of how web scraping works:
    1. Select the target website
    2. Choose a scraping library
    3. Inspect the website
    4. Write the scraping code
    5. Handle pagination and navigation
    6. Clean and process the data

Applications of web scraping in Python

- Data collection and analysis
- Content aggregation
- Lead generation
- Monitoring and tracking

Demo

Introduction to Data Visualization with Python

What we have to study?

- Why data visualization is important
- Introduction to Matplotlib and Seaborn libraries
- Creating basic plots: line, bar, scatter, and pie charts
- Customizing plots: labels, colors, and styles
- Visualizing data from Pandas DataFrame

Introduction

- Python is a powerful programming language widely used for data analysis and visualization due to its extensive libraries and tools.
- Data visualization is a crucial aspect of data analysis and interpretation.
- It involves representing data visually using charts, graphs, and other visual elements to uncover patterns, relationships, and insights that might not be immediately apparent in raw data.
- The most popular library for data visualization is Matplotlib, which provides a wide range of plotting functions and options.
- Additionally, there are other libraries such as Seaborn and Plotly that offer more advanced and interactive visualization capabilities.

Why data visualization is important

- Understanding and Exploring Data
- Communicating Insights
- Identifying Patterns and Trends
- Enhancing Data Analysis
- Improving Decision Making
- Storytelling and Presentations
- Data Quality and Anomaly Detection

Demo

Introduction to Machine Learning with Python

What we have to study?

- Overview of machine learning concepts
- Introduction to scikit-learn library
- Supervised learning algorithms: regression and classification
- Unsupervised learning algorithms: clustering and dimensionality reduction
- Model evaluation and validation techniques

Overview of Machine Learning concepts

- Machine learning (ML) is a field of study and practice that focuses on the development of algorithms and statistical models that enable computers to learn and make predictions or decisions without being explicitly programmed.
- It involves the analysis of large amounts of data to uncover patterns, relationships, and insights that can be used to make predictions or automate tasks.
- Machine Learning is continuously growing in the IT world and gaining strength in different business sectors.

Overview of Machine Learning concepts

- Machine Learning enables computers to behave like human beings by training them with the help of past experience and predicted data.
- There are three key aspects of Machine Learning, which are as follows:
    - **Task**: A task is defined as the main problem in which we are interested. This task/problem can be related to the predictions and recommendations and estimations, etc.
    - **Experience**: It is defined as learning from historical or past data and used to estimate and resolve future tasks.
    - **Performance**: It is defined as the capacity of any machine to resolve any machine learning task or problem and provide the best outcome for the same. However, performance is dependent on the type of machine learning problems.

M.L Concepts

- Data
- Feature Extraction
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Neural Networks
- Model Evaluation
- Overfitting and Underfitting
- Regularization
- Hyperparameters

M.L libraries for Python

- Python offers a rich ecosystem of libraries and frameworks for machine learning.
- Some of them are follows:
  - NumPy
  - Pandas
  - Scikit-learn
  - TensorFlow
  - PyTorch
  - Keras
  - XGBoost
  - LightGBM
  - SciPy

What is Scikit-Learn (Sklearn) ?

- Scikit-learn is a powerful and widely-used machine learning library in Python.
- It provides a comprehensive set of tools for various machine learning tasks, making it a go-to choice for many practitioners and researchers.
- An elaboration on Scikit-learn's key features and functionalities:
  - Consistent API
  - Supervised Learning
  - Unsupervised Learning
  - Preprocessing and Feature Extraction
  - Model Evaluation
  - Integration with Other Libraries
  - Extensibility
  - Robust Documentation and Community Support

Demo

Python Open House batches

Open-House Batches Advance Python

Duration – 5 Days

Special Discounted Price- 8999/- Per Pax

Private & Public Batches

Kindly reach for more details.

Pioneer Business Solutions

THANK YOU