Introduction to Python for AI & ML

Key Objectives

Session 1:

- Understand Why Python is Used for AI
- Learn Key Programming Concepts for AI Development
- Introduction to NumPy for Numerical Computation
- Introduction to Pandas for Data Manipulation

GALALA UNIVERSITY
Powered by
Arizona State University

Session 1: Introduction to Python for AI & ML

Why Python is Used for AI ?

Why Python is Used for AI ?

1. Simplicity and Readability

- Python has a clean and simple syntax, making it easy to learn and write.
- Readability allows AI researchers and engineers to focus on solving problems rather than syntax issues.

Why Python is Used for AI ?

2. Large Ecosystem of Libraries

- Python has a vast collection of AI and data science libraries:
  - NumPy (Numerical computations)
  - Pandas (Data manipulation)
  - Scikit-learn (Machine learning)
  - TensorFlow & PyTorch (Deep learning)
  - Matplotlib & Seaborn (Data visualization)

Why Python is Used for AI ?

3. Community and Support

- Large global community ensures quick bug fixes and vast documentation.
- Many AI and ML tutorials, research papers, and pre-trained models are available in Python.

Session 1: Introduction to Python for AI & ML

Key Programming Concepts for AI

Key Programming Concepts for AI

- Key Data Types

- Data types include integers, floats, Booleans and strings.

```
# --- Variables and Data Types ---
# Define variables of various types.

product_name = 'milk'        # String variable
quantity_sold = 10                # Integer variable
price_per_unit = 20.0            # Float variable
in_stock = True                    # Boolean variable
```

GALALA UNIVERSITY
Powered by
Arizona State University

Key Programming Concepts for AI

- Key Data Structures
- Lists and Dictionaries

```
fruits = ["Apple", "Banana", "Cherry"]
print("List of Fruits:", fruits)
# Accessing elements
print("First Fruit:", fruits[0])
# Modifying list elements
fruits.append("Orange")  # Add an element
print("Updated List:", fruits)

person = {
    "name": "Alice",
    "age": 25,
    "city": "Cairo"
}
print("\nPerson Dictionary:", person)
# Accessing values
print("Name:", person["name"])
print("Age:", person.get("age"))
# Modifying dictionary
person["age"] = 26  # Update age
person["job"] = "Engineer"  # Add new key-value pair
print("Updated Dictionary:", person)
```

Key Programming Concepts for AI

- ➢ Conditional Statements
- If statements

```python
# If statement with multiple conditions
score = 85

if score >= 90:
    grade = "A"
elif score >= 80:
    grade = "B"
elif score >= 70:
    grade = "C"
elif score >= 60:
    grade = "D"
else:
    grade = "F"

print(f"Score: {score}, Grade: {grade}")
```

GALALA UNIVERSITY
Powered by
Arizona State University

Key Programming Concepts for AI

- Loops
- For and while loops

```python
fruits = ["Apple", "Banana", "Cherry", "Orange"]
for fruit in fruits:
    print("Fruit:", fruit)

for num in range(1, 6):  # Loops from 1 to 5
    print("Number:", num)
```

```python
count = 5
while count > 0:
    print("Countdown:", count)
    count -= 1

password = ""
while password != "1234":
    password = input("Enter password: ")
print("Access Granted!")
```

GALALA UNIVERSITY
Powered by
Arizona State University

Key Programming Concepts for AI

- Functions

```python
# --- Functions ---
def calculate_discount(revenue, discount_rate=0.1):
  discount = revenue * (1 - discount_rate)
  return discount

def greet():
    print("Hello, welcome to AI programming!")

def add(a, b):
    return a + b

def rectangle_area_perimeter(length, width):
    area = length * width
    perimeter = 2 * (length + width)
    return area, perimeter
```

Session 1: Introduction to Python for AI & ML

Intro to numpy

What is NumPy?

- NumPy (Numerical Python) is a library for numerical computations.

- Provides fast and efficient array operations.

- Used in AI, ML, and data science for mathematical computations.

Why Use NumPy?

- Faster than Python lists (uses optimized C backend).
- Supports multi-dimensional arrays (arrays, matrices, tensors).
- Built-in mathematical functions for statistics, linear algebra, and random numbers.

Session 1: Introduction to Python for AI & ML

Intro to pandas

What is Pandas?

1. What is Pandas?

- Pandas is a library for data manipulation and analysis.

- Provides DataFrames, a tabular structure similar to Excel.

- Used in AI and ML for cleaning, transforming, and analyzing data.

Why Use Pandas?

2. Why Use Pandas?

- Efficient handling of large datasets.
- Supports data cleaning and transformation.
- Works well with NumPy, Matplotlib, and Scikit-learn.
- Fast operations on structured data (tables).

Key Features of Pandas

- Series: One-dimensional labeled data.
- DataFrame: Two-dimensional labeled table.
- Data Cleaning: Handling missing values, filtering, sorting.
- Data Aggregation: Grouping, summarization.

Session 1: Introduction to Python for AI & ML

Object Oriented Programing

Object Oriented Programing

What is Object-Oriented Programming (OOP)?

- A programming paradigm based on objects & classes
- Helps in structuring code for scalability, reusability, and maintainability
- Used in AI, software development, game development, and more

Object Oriented Programing

When to Use OOP in AI?

- When building large-scale AI applications.
- When building your own model from scratch
- When working on modular AI pipelines
- When developing AI models with reusable components.

Object Oriented Programing

Encapsulation & Abstraction (Data Hiding & Security)

- Bundles data & methods together in a class
- Prevents direct access to certain data (using private variables)
- Improves modularity and security

Object Oriented Programing

- Encapsulation & Abstraction (Data Hiding & Security)
  - Bundles data & methods together in a class
  - Prevents direct access to certain data (using private variables)
  - Improves modularity and security
- Abstraction (Hiding Complexity)
  - Shows only the necessary details, hiding the internal workings
  - Makes the system easier to use & maintain

Object Oriented Programing

Inheritance & Polymorphism

- Inheritance (Code Reusability)
  - A child class inherits attributes & methods from a parent class
  - Avoids code duplication
- Polymorphism (Multiple Forms, Flexibility)
  - Allows different classes to use the same method but with different behaviors
  - Makes code more flexible & scalable

Any question

Thank you