Bytewise Fellowship Program

DATA SCIENCE

Task #4

BWT- Data Science (Group1)

Submitted to: Mahrukh Khan
Submitted by: Usama Malik

Task: All about Python 🐍 Introduction, Syntax, Variables, conditions and operators.

Python

Python is a high-level, interpreted programming language known for its simplicity and readability. It supports multiple programming. Python is widely used for web development, data analysis, artificial intelligence, games, and more.

Use of Strings

Strings are used to handle text data enclosed in quotes.

Example: Program = "Hello, World!"

Concatenation (joining the two strings)

```
first_name = "John"
last_name = "wick"
full_name = first_name + " " + last_name
print(full_name)
```

Output: John wick

Use of Numbers

Python supports various types of numbers, integers, floating-point numbers, and complex numbers.

result = 5 + 3 print(result)

Output: 8

Variables and Data Types

Definition: Variables are used to store data, and data types define the type of data a variable can hold.

Example: age = 25 # integer

name = "Alice" # string

height = 5.7 # float

is_student = True # Boolean

Operators

Definition: Operators are symbols that perform operations on variables and values.

Example

```
sum = 5 + 3  # addition

difference = 10 - 2  # subtraction

product = 4 * 2  # multiplication

quotient = 8 / 2  # division
```

Loops

Definition: Loops are used to repeat a block of code multiple times.

Example:

```
# for loop
for i in range(5):
print(i)

# while loop
count = 0
while count < 5:
print(count)
count += 1
```

Functions

Definition: Functions are used to perform a specific task.

Example:

```
def greet(name):
return f"Hello, {name}!"
print(greet("Alice"))
```

Lists

Definition: A lists are ordered collections of items that can be modify

Example:

```
fruits = ["apple", "banana", "cherry"]
print(fruits[1])  # Output: banana
```

Tuples

Definition: A tuple is an ordered collection of items that cannot be modified after creation.

Example:

```
point = (10, 20)
print(point[0])  # Output: 10
```

Sets

Definition: Sets are unordered collections of unique items.

Example:

```
colors = {"red", "green", "blue"}
print("red" in colors)  # Output: True
```

File Handling

Definition: File handling involves reading from and writing to files.

Example:

```
with open("example.txt", "w") as file:
file.write("Hello, world!")
with open("example.txt", "r") as file:
content = file.read()
 print(content)
```

Exception Handling

Definition: Exception handling manages errors in a program.

Example:

```
try:
result = 10 / 0
except ZeroDivisionError:
 print("Cannot divide by zero!")
```

Classes and Objects

Definition: Classes define the structure and behavior of objects.

Example:

```
class Dog:
def __init__(self, name):
self.name = name
def bark(self):
```

```
return f"{self.name} says woof!"
my_dog = Dog("Buddy")
print(my_dog.bark())  # Output: Buddy says woof!
```

Data Analysis (with pandas)

Definition: pandas is a library for data manipulation and analysis.

Example:

```
import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]}

df = pd.DataFrame(data)
print(df)
```

Plotting (with matplotlib)

Definition: matplotlib is a library for creating plots.

Example:

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
plt.plot(x, y)
plt.show()
```

Web Scraping (with BeautifulSoup)

Definition: BeautifulSoup is a library for parsing HTML and XML documents.

Example:

```
from bs4 import BeautifulSoup
import requests
response = requests.get('http://geo news)
soup = BeautifulSoup(response.text, 'html.parser')
print(soup.title.text)
```