A Journey through the Basics of python programming

TIME REQUIRE: 30 MINUTES

Created By: Sazzad Hossain

Topic Cover:

1. Python introduction.
2. python use case
3. python installation
4. IDE & Editor for python
5. Variable and data types
6. Operators in python
7. print statements
8. Comment
9. Indentation
10. Conditional Statements
11. Functions
12. List, tuples , Dictionaries, sets
13. Input , Output
14. File handling
15. Error Handling
16. Modules and Libraries
17. List Comprehensions
18. Lambda Functions

Created By: Sazzad Hossain

Python Introduction:

Python is a high level programming language, created by Guido van Rossum in 1991.

Use cases of python:

1. Web development
2. Data analysis and visualization
3. Machine learning
4. Desktop Application
5. Game development
6. Natural language processing
7. Image processing and many more

Created By: Sazzad Hossain

Python installation:

1. Download Python from the official website (python.org).
2. Choose the appropriate version (Python 3.x recommended)
3. Install

Python IDEs and Editors:

1. PyCharm
2. Or Jupyter Notebook

Created By: Sazzad Hossain

V a r i a b l e :

1. Variable names must start with a letter (a-z, A-Z) or an underscore (_)
2. variable names can also include numbers (0-9)
3. Variable names are case-sensitive, meaning myVariable and myvariable are considered different variables.

C r e a t e d B y : S a z z a d H o s s a i n

Data types:

Python is dynamically typed, meaning we don't need to declare the data type of a variable.

1. int(for integer value)
2. float(floating-point numbers)
3. str(for string)
4. list(list of number, string)
5. tuple(tuple)
6. set(set)
7. dict(key value pair)

Operators:

1. arithmetic operators (+, -, *, /, //, %)
2. comparison operators (<, >, <=, >=, ==, !=)
3. logical operators (and, or, not)

Print:

For printing something use print() function

```
print("Hello, World!")
```

Created By: Sazzad Hossain

Comment:

This is a comment in python

Indentation:

Python uses indentation (whitespace)
to define code blocks instead of curly braces { }

Created By: Sazzad Hossain

Conditional Statement:

Python uses if, elif, and else for conditional branching

```
if condition:
    # this part will execute if the condition is True
elif another_condition:
    # this part will execute if the another_condition is True
else:
    # this part will execute if no conditions are True
```

Created By: Sazzad Hossain

L o o p :

Python supports for and while loops for iteration.

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

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

Function:

Functions are defined using the def keyword and can take parameters and return values.

```
#Function with parameters
def add(a, b):
    return a + b
```

Created By: Sazzad Hossain

Lists, Tuples, and Dictionaries:

Functions are defined using the def keyword and can take parameters and return values.

1. Lists are ordered collections of elements.
2. Tuples are similar with list but immutable.
3. Dictionaries store key-value pairs

```
#define list
my_list = [1, 2, 3]
#define tuple
my_tuple = (4, 5, 6)
#define dictionaries
my_dict = {"name": "Alice", "age": 30}
```

Created By: Sazzad Hossain

Input and output:

python use input function for take input from user and print function for the display of output

```
#input
name = input("Enter your name: ")
#output
print("Hello, " + name)
```

File Handling:

File handling allows you to read from and write to files. Here's an example of reading from a text file and then writing to another

```
# Reading from a file
with open("input.txt", "r") as file:
    content = file.read()
    print("Contents of input.txt:")
    print(content)

# Writing to a file
with open("output.txt", "w") as file:
    file.write("This is a sample line.\n")
    file.write("Another line for the output file.\n")
```

Error Handling:

Error handling is essential for dealing with exceptions

```
try:
    num = int(input("Enter a number: "))
    result = 10 / num
except ZeroDivisionError:
    print("Error: Cannot divide by zero.")
except ValueError:
    print("Error: Invalid input. Please enter a number.")
else:
    print(f"Result: {result}")
finally:
    print("Execution complete.")
```

Modules and Libraries:

Python's modules and libraries extend its functionality

#example
import math

num = 25
sqrt = math.sqrt(num)
print(f"The square root of {num} is {sqrt}")

Lambda functions:

Lambda functions are anonymous functions.

```
square = lambda x: x ** 2
result = square(4)
print(f"The square of 4 is {result}")
```

Created By: Sazzad Hossain

Thank you