Python Basics

1. What is Python, and why is it popular?

Python is a high level, general purpose programming language renowned for it's:

- Readable Syntax: It's syntax is clear, easy to understand, and write.
- Dynamic Typing: No need to declare variable types.
- Versatility: It excels in web development, data science, automation, machine learning, and more
- Extensive Libraries: It supports diverse functionalities.
- Community Support: It has a large and active community.

Use Cases:

- Web Development e.g. Django, Flask.
- Data Science and Machine Learning e.g. Tensorflow, Pandas, and NumPy.
- Scripting and Tasks Automation.

2. Installing Python and Verification.

Windows:

1. Download Python installer from Download Python | Python.org.
2. Run the installer and select 'Add Python to PATH' for system wide access.
3. Follow the installation prompts.

macOS:

1. Open Terminal.
2. Install Homebrew if not installed: ‘’’/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"’’’.
3. Install Python: “””brew install python”””.

Linux:

1. Open Terminal.
2. Update package lists: “””sudo apt update”””.
3. Install Python:””” sudo apt install python3”””.

Verification:

- Open terminal/ command prompt and type ””” python3 –version”””

Setting Up Virtual Environment:

1. Use venv module type in the terminal: “””python3 –m venv myEnv”””[myEnv is the name of the virtual environment you have created]
2. Activate:

- Windows: “””myEnv\Scripts\activate.bat”””
- macOS/Linux: “””source myEnv/bin/activate”””

3. Hello, World! Program

“””
print("Hello, World!")
“””

4. Basic Data Types

- Numbers(int, float): Integers e.g 60 and floating-point numbes e.g 1.234
- String(str): e.g Hello, Come, E.T.C.
- Booleans(bool): True or False values.

Script:

“””
age= 5 # int
height= 3.14 # float
name = "Alice" # str
is_valid = True # bool

print(f”Name:{name}, Age:{age}, Height:{height}”)

Print(is_valid)
“””

5. Conditional Statements and Loops

1. if-else Statement:

“””
x = 15
if x < 8:
    print("x is less than 8")
else:
    print("x is not less than 8")
“””

2. for Loop:

“””

no_s = [‘1’, ‘5’, ‘10’]
for no in range no_s:

```
    print(no)
“””
```

6. Functions

Definition and Uses:

- Reusable blocks of code that performs specific tasks.
- They improve code modularity and reusability

```
“””
def add(a, b):
   return a + b

result = add(2, 3)
print(result) #Output: 5
“””
```

7. Lists and Dictionaries

Script:

```
“””
numbers = [1, 2, 3, 4, 5]  # List: ordered, can have duplicates
data = {"name": "Alice", "age": 25}  # Dictionary: unordered, unique keys

# List operations
numbers.append(6) #Add element to list
print(numbers) # Output: [1, 2, 3, 4, 5, 6]

# Dictionary operations
print(data["name"]) #Access dict value by key(name)
data["age"] = 26 #Modify dict value
print(data) # Output: {"name": "Alice", "age": 26}

“””
```

8. Exception Handling

- try-except-finally: Manages errors gracefully.
- try: Block where code might raise an exception.
- except: Block that handles the exception if it occurs.
- Finally: Block that always executes, regardless of exceptions.

Example:

“””

```
try:
    x = 1 / 0
except ZeroDivisionError:
    print("Cannot divide by zero!")
finally:
    print("Execution completed.")
```

“””

9. Modules and Packages

- Modules are python files containing reusable code.
- Packages are hierarchies of modules organized in directories.

Example:

“””

```
import math # math is a module

print(math.sqrt(16)) # is a function in the math module
```

“””

10. File Operations

- “””open(filename, mode”””: Opens a file for reading (r), writing (w), appeading (a).

1. Reading from a File:

“””

```
with open("file.txt", "r") as file:
    content = file.read()
    print(content)
```

“””

2. Writing to a File:

“””

```
data = ["Line 1", "Line 2"]
with open("output.txt", "w") as file:
file.writelines(data) # Write multiple lines
```

“””

References:

- The Python Tutorial — Python 3.12.4 documentation

- Learn Python the Hard Way (A more challenging approach)
- Python If Elif (w3schools.com)
- Data Types — Python 3.12.4 documentation
- Python Try Except (w3schools.com)
- File and Directory Access — Python 3.12.4 documentation