



Python Programming Fundamentals using Jupyter & Google Colab

**Class 2
28/1/2025**

Acknowledgement

**The series of the IT & Japanese language course is
Supported by AOTS and OEC.**



Ministry of Economy, Trade and Industry

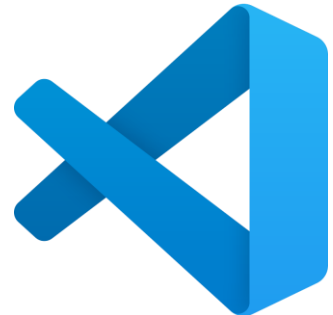


Overseas Employment Corporation

Software Used

Following software are frequently use.

1. PyCharm
2. Jupyter Notebook
3. Google Colab
4. GitHub
5. AWS
6. VS code



What you will Learn Today

We will focus on following points.

- Introduction of Python, Variables & Operators
- Data Types & String Manipulation
- Introduction of Google colab and Jupyternotebook
- Basic Python Syntax and Data Types
- Lower, upper, length (len), random and split function
- Random module
- Upload code on github
- Quiz
- Q&A Session

Introduction of Python

Python is a powerful, high-level, and versatile programming language

[Features of Python]

- **Interpreted:** Code is executed line by line without prior compilation.
- **High-level:** Abstracts complex details of computer operations, making it user-friendly.
- **Dynamic Typing:** No need to declare variable types explicitly.
- **Versatile:** Compatible with multiple platforms and environments.
- **Extensive Libraries:** Rich standard library and third-party packages for varied tasks.

Python: A language for everyone, from beginners to experts!"

Applications of Python

Python is designed to be easy to learn and use, emphasizing readability and simplicity

[Applications of Python]

- **Web Development:** Frameworks like Django, Flask, and FastAPI.
- **Data Science:** Libraries like Pandas, NumPy, and Matplotlib.
- **AI and Machine Learning:** Tools like TensorFlow, PyTorch, and Scikit-learn.
- **Automation:** Scripts for task automation.
- **Game Development:** Libraries like Pygame.
- **Scientific Computing:** Tools like SciPy and SymPy.

Python: A language for everyone, from beginners to experts!"

Introduction of Google colab and Jupyternotebook

Various software for setting up python environment

[Tools for using Python]

- VS Code Editor
- PyCharm
- Jupyter Notebook
- Google Colab

Python: A language for everyone, from beginners to experts!"

Introduction of Google colab

Setting Up Google Colab

Step 1: Open Google Colab

1. Visit Google Colab using your browser.
2. Log in with your Google account.

Step 2: Start a New Notebook

1. On the Colab homepage, click **“New Notebook”**.
2. You’ll see a code cell where you can start writing Python code.

Python: A language for everyone, from beginners to experts!"



Learn Python with Projects

Basic Python Syntax and Data Types

Python has a clean and readable syntax, making it beginner-friendly

[Python Syntax Basics]

- Indentation and Comments
- Variables and Constants
- Input and Output Functions

Python: A language for everyone, from beginners to experts!"

Indentation and Comments

Python uses indentation (spaces or tabs) to define blocks of code instead of braces {} like in other languages

```
# This is a single-line comment in Python

# Example of indentation
if True:
    print("This is indented correctly.") # Inside the block

# Incorrect indentation will raise an error:
# if True:
# print("This will cause an IndentationError.")
```

Python: A language for everyone, from beginners to experts!"

Variables and Constants

Variables are containers for storing data, while constants are values that don't change

```
# Variable
name = "Umar"
age = 26

# Constant (by convention, Python doesn't enforce immutability)
PI = 3.14159

print("Name:", name)
print("Age:", age)
print("PI:", PI)
```

Python: A language for everyone, from beginners to experts!"

Input and Output Functions

Python provides input() for user input and print() for displaying output.

```
# Getting user input
user_name = input("Enter your name: ")
user_age = int(input("Enter your age: ")) # Convert input to integer

# Displaying output
print("Hello,", user_name + "!")
print("You are", user_age, "years old.")
```

Python: A language for everyone, from beginners to experts!"

Primitive Data Types

Integers, Floats, and Complex Numbers

```
# Integer
x = 10
print("Integer:", x)

# Float
y = 3.14
print("Float:", y)

# Complex number
z = 2 + 3j
print("Complex Number:", z)
print("Real part:", z.real)
print("Imaginary part:", z.imag)
```

Python: A language for everyone, from beginners to experts!"

Primitive Data Types

Strings and Their Operations

```
# String
text = "Hello, Python!"

# Accessing characters
print("First character:", text[0])
print("Last character:", text[-1])

# String concatenation
greeting = "Hello" + " World!"
print("Concatenated String:", greeting)

# String repetition
repeat = "Python! " * 3
print("Repeated String:", repeat)
```

```
# String slicing
print("Sliced String:", text[0:5]) # Output: Hello
# text[0:5] extracts characters from the string text starting from
index 0 (the first character, H) up to (but not including) index 5
(the comma ,)

# String methods
print("Uppercase:", text.upper())
print("Lowercase:", text.lower())
print("Replace:", text.replace("Python", "World"))
```

Collection Data Types

Lists

```
# List
numbers = [1, 2, 3, 4, 5]
print("List:", numbers)

# Accessing elements
print("First element:", numbers[0])

# Modifying list
numbers.append(6)
print("After appending:", numbers)

# List slicing
print("First 3 elements:", numbers[:3]) #numbers[:3] extracts
elements at indices 0, 1, and 2
```


Collection Data Types

Tuples

```
# Tuple
coordinates = (10, 20, 30)
print("Tuple:", coordinates)

# Accessing elements
print("Second element:", coordinates[1])

# Tuples are immutable, so no modification is allowed
```

Collection Data Types

Sets

```
# Set
unique_numbers = {1, 2, 3, 4, 5}
print("Set:", unique_numbers)

# Adding elements
unique_numbers.add(6)
print("After adding:", unique_numbers)

# Sets are unordered and do not allow duplicates
unique_numbers.add(3)
print("After trying to add duplicate:", unique_numbers)
```

Collection Data Types

Dictionaries

```
# Dictionary
student = {"name": "Alice", "age": 22, "grade": "A"}
print("Dictionary:", student)

# Accessing values
print("Name:", student["name"])

# Modifying values
student["grade"] = "A+"
print("Updated Dictionary:", student)

# Adding new key-value pairs
student["subject"] = "Math"
print("After adding subject:", student)
```

Collection Data Types

List vs Tuple vs Set vs Dictionary

Property	List	Tuple	Set	Dictionary
Mutable	Yes	No	Yes	Yes
Ordered	Yes	Yes	No	Yes (Python 3.7+)
Duplicates	Yes	Yes	No	Keys: No, Values: Yes
Access	Index-based	Index-based	Value-based	Key-based
Syntax	[1, 2, 3]	(1, 2, 3)	{1, 2, 3}	{"key": "value"}

Type Conversion

Python automatically converts a smaller data type to a larger data type to prevent data loss

```
# Implicit conversion
num_int = 10
num_float = 3.5

result = num_int + num_float # int is converted to float
print("Result:", result)
print("Type of result:", type(result))
```

Type Conversion

You manually convert data types using typecasting functions

```
# Explicit conversion
num_str = "25"
num_int = int(num_str) # Convert string to integer
print("Converted Integer:", num_int)

# Convert integer to float
num_float = float(num_int)
print("Converted Float:", num_float)

# Convert number to string
num_str_again = str(num_float)
print("Converted String:", num_str_again)
```

Operators

Arithmetic Operators

```
a = 15  
b = 4  
  
print("Addition:", a + b)  
print("Subtraction:", a - b)  
print("Multiplication:", a * b)  
print("Division:", a / b)  
print("Floor Division:", a // b)  
print("Modulus:", a % b)  
print("Exponentiation:", a ** b) #e.g. a^b => 15^4
```

Operators

Comparison Operators

```
x = 10  
y = 20  
  
print("Equal:", x == y)  
print("Not Equal:", x != y)  
print("Greater Than:", x > y)  
print("Less Than:", x < y)  
print("Greater or Equal:", x >= y)  
print("Less or Equal:", x <= y)
```


Operators

Logical Operators

```
p = True  
q = False  
  
print("AND:", p and q) # Both must be True  
print("OR:", p or q)  # At least one must be True  
print("NOT:", not p)  # Negates the value
```

Operators

Bitwise Operators

```
x = 6 # Binary: 110
```

```
y = 3 # Binary: 011
```

```
#If both bits are 1, the result is 1; otherwise, it is 0
```

```
print("Bitwise AND:", x & y) # Binary: 010 -> 2
```

```
#If at least one bit is 1, the result is 1; otherwise, it is 0.
```

```
print("Bitwise OR:", x | y) # Binary: 111 -> 7
```

```
#If the bits are different, the result is 1; if they are the same, the result is 0
```

```
print("Bitwise XOR:", x ^ y) # Binary: 101 -> 5
```

```
# Flips all the bits of the number, changing 0 to 1 and 1 to 0
```

```
print("Bitwise NOT:", ~x) # Inverts all bits -> -7
```

```
#Shifts all bits of the number to the left by the specified number of positions, and  
fills in 0s from the right
```

```
print("Left Shift:", x << 1) # Binary: 1100 -> 12
```

```
#Shifts all bits of the number to the right by the specified number of positions,  
discarding the bits that are shifted out, and fills in 0s from the left
```

```
print("Right Shift:", x >> 1) # Binary: 011 -> 3
```

Operators

Assignment Operators

```
a = 10
```

```
a += 5 # Equivalent to a = a + 5  
print("Addition Assignment:", a)
```

```
a -= 3 # Equivalent to a = a - 3  
print("Subtraction Assignment:", a)
```

```
a *= 2 # Equivalent to a = a * 2  
print("Multiplication Assignment:", a)
```

```
a /= 4 # Equivalent to a = a / 4  
print("Division Assignment:", a)
```

```
a %= 3 # Equivalent to a = a % 3  
print("Modulus Assignment:", a)
```

```
a **= 2 # Equivalent to a = a ** 2  
print("Exponentiation Assignment:", a)
```

```
a //= 2 # Equivalent to a = a // 2  
print("Floor Division Assignment:", a)
```

Operators

Assignment Operators

```
a = 10
```

```
a += 5 # Equivalent to a = a + 5  
print("Addition Assignment:", a)
```

```
a -= 3 # Equivalent to a = a - 3  
print("Subtraction Assignment:", a)
```

```
a *= 2 # Equivalent to a = a * 2  
print("Multiplication Assignment:", a)
```

```
#gives a float result (5.0), even though  
it's a whole number.
```

```
a /= 4 # Equivalent to a = a / 4  
print("Division Assignment:", a)
```

```
a %= 3 # Equivalent to a = a % 3  
print("Modulus Assignment:", a)
```

```
a **= 2 # Equivalent to a = a ** 2  
print("Exponentiation Assignment:", a)
```

```
#gives an integer result (5), truncating any decimal part  
a //= 2 # Equivalent to a = a // 2  
print("Floor Division Assignment:", a)
```

Length Function (len())

The len() function returns the number of items in an object (e.g., a string, list, tuple, dictionary)

```
string = "Python"  
numbers = [1, 2, 3, 4, 5]  
  
# Length of a string  
print("Length of string:", len(string))  
  
# Length of a list  
print("Length of list:", len(numbers))
```

Lowercase (lower()) and Uppercase (upper())

These methods are used to convert strings to lowercase or uppercase

```
text = "Hello, Python World!"  
  
# Convert to lowercase  
print("Lowercase:", text.lower())  
  
# Convert to uppercase  
print("Uppercase:", text.upper())
```

Splitting a String (split())

The split() method breaks a string into a list of substrings based on a specified delimiter (default is a space)

```
text = "Python is fun"

# Split using default space
words = text.split()
print("Split by space:", words)

# Split using a specific delimiter
csv_line = "Name,Age,Country"
data = csv_line.split(",")
print("Split by comma:", data)
```

Random Module

The random module is used to generate random numbers or select random items.

```
import random

# Random integer between 1 and 10
print("Random Integer:",
      random.randint(1, 10))

# Generates a random float between 0
and 1
print("Random Float:",
      random.random())

# Generates a random float between a
and b
print("Random Float (1 to 5):",
      random.uniform(1, 5))
```

```
items = ['apple', 'banana', 'cherry', 'date']

# Random choice from a list
print("Random Choice:", random.choice(items))

# Shuffle a list
random.shuffle(items)
print("Shuffled List:", items)

# Random sample of multiple items
print("Random Sample:", random.sample(items, 2))
```


Quiz Section

Quiz

Everyone student should click on submit button otherwise MCQs will not be submitted after time will finish

[Guidelines of MCQs]

1. There are 20 MCQs
2. Time duration will be 10 minutes
3. This link will be share on 6:10pm (Pakistan time)
4. MCQs will start from 6:15pm (Pakistan time)
5. This is exact time and this will not change
6. Everyone student should click on submit button otherwise MCQs will not be submitted after time will finish
7. Every student should synchronize there laptop clock with actual time otherwise they cannot solve the MCQs

<https://quiz.everestwebdeals.co/?form=4b60dc16b3c6f1e4c3e4aa9258dab582>

Q&A Session

ありがとうございます。

Thank you.

شكريا



For the World with Diverse Individualities