

Presented by Sahil Chawla

Data Analytics with Python



Course Timeline



Week 1	●	16 October: In-person Session 1
Week 2	●	23 October: In-person Session 2
Week 3	●	30 October: In-person Session 3 Assessment available
Week 4	●	06 November: In-person Session 4
Week 5	●	13 November: Assessment Due (No class)

Certificate Eligibility

Attendance:

Minimum of 75%
(present in 3/4 sessions)

Assessment:

Must score >80%

Course Overview



Programming in Python (Basics)

Week 1 and 2

- Variables and Data Types
- Input/Output operations
- Conditional Statements
- Loops
- Functions
- Data Structures (Lists, Dictionaries)

Data Analytics w/Python Libraries

Week 3 and 4

- Introduction to Pandas for data manipulation
- Data visualization using Matplotlib
- Working with NumPy for numerical computations

What is python?



High-level programming language - Focuses on simplicity

Versatile and Widely Used - Web development, data science, automation, AI & more.

Large Standard Library - Built-in modules for diverse tasks.

Most importantly - Beginner Friendly

How to use python? (Online/Cloud) +++

Google Colab

- Go to <https://colab.research.google.com/>
- Click on File => New notebook
- Rename “untitled1.ipynb” to name of your choice

How to use python? (Offline/Local) +++

Step 1: Install python (windows)

- Go to <https://www.python.org/downloads/>
- Click on download python 3.10.7
- Once python-3.10.7-amd64.exe is downloaded, click on install now
- Go on cmd (command prompt) to check if python is downloaded or not
 - Write python3 -version
- This opens python in interpreter mode (basic calculations and other stuff)

How to use python? (Offline/Local) +++

Step 2: Open any IDE (VS Code, PyCharm, etc.).

- Click on open folder
- Create a folder on desktop
- Create a .py file by clicking on the + icon
- Run scripts using run button or use command `python <name.py>`

Variables in python

+++

- A variable stores information that can be used and manipulated in a program.
- It acts as a container for data.



```
1 # Variables are assigned using the = sign
2 # Syntax: variable_name = variable_value
3
4 name = 'Sahil'
5 age = 22
6 email = 'sahil.chawla@dal.ca'
7 is_student = False
```


Variables in python



Variable Naming Rules

- Case-sensitive
- Variable names in Python can be any length and can consist of uppercase and lowercase letters (A-Z, a-z), digits (0-9), and the underscore character (_).
- Although a variable name can contain digits, the first character cannot be a digit
 - e.g.: 3xcs = 23 (is wrong)
- Cannot use reserved keywords

False	def	if	raise
None	del	import	return
True	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	from	or	
continue	global	pass	

Python Data Types



- Variables can store different types of data, and each type behaves differently
- Data types:
 - **int**: Integer numbers (e.g., 5, -3)
 - **float**: Decimal numbers (e.g., 4.5, 3.14)
 - **str**: Text or characters (e.g., "Hello", "email@email.com", "claSs#1")
 - **bool**: Boolean values (True, False)
- **Python assumes the type of variable based on the assigned value**

Input Operation

+++

- input() function allows users to input values (case sensitive)
- Input taken by a user is saved in a variable
- **Everything that input receives is stored as string data type**



```
1 # Syntax: variable_name = input("prompt to user, if any")
2 name = input("Enter your name: ")
3 >> Enter your name:
4
5 # Need to define a data type in case we expect any other value
6 ph_num = (int)(input("Enter your phone number:"))
7 >> Enter your phone number:
```

Output Operation

+++

Using print() function (case-sensitive)



```
1 # Syntax: print("information") or print(variable_name)
2
3 print ("hello world")
4 >> hello world
5
6 x = 3.5
7 print(x)
8 >> 3.5
```

Conditional Statements

+++

- Conditional statements are used to perform different actions based on different conditions
- In Python, we primarily use the if, elif, and else statements in combination with comparison operators and logical operators

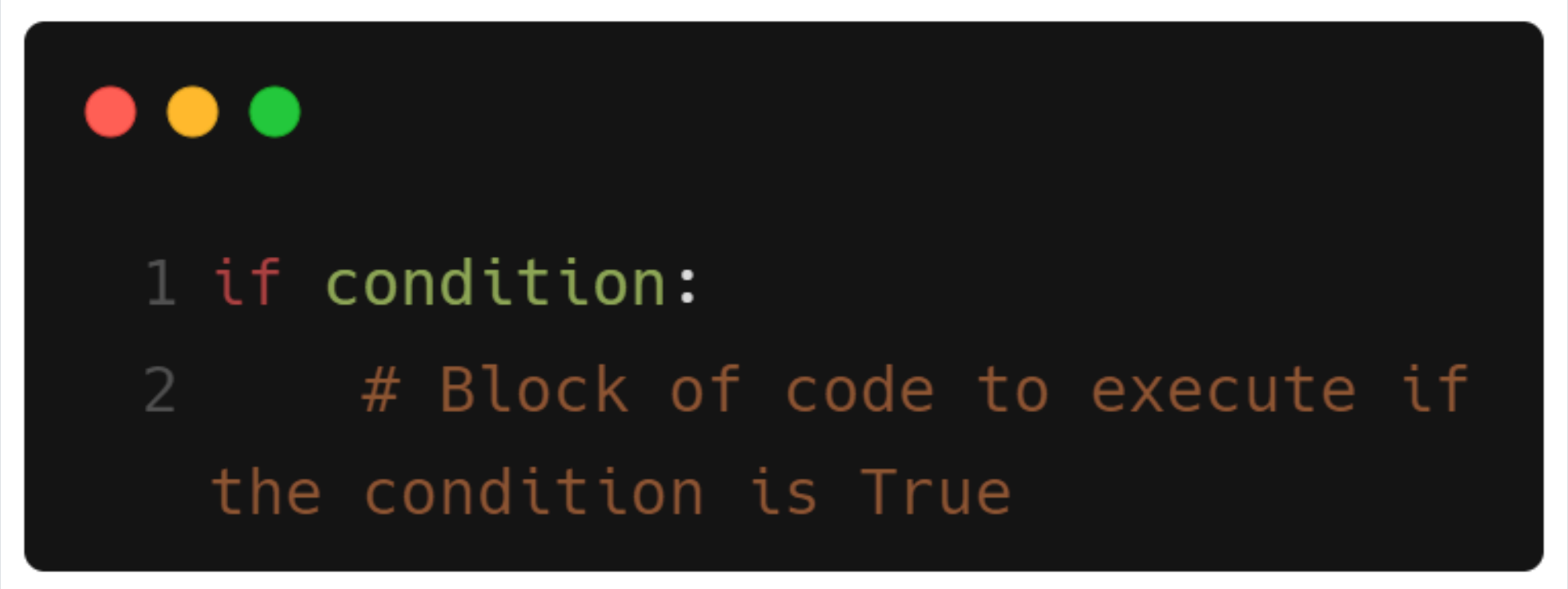
Operator	Denotes	Type
==	Equal To	Comparision
!=	Not Equal To	Comparision
>	Greater Than	Comparision
<	Less Than	Comparision
>=	Greater Than or Equal To	Comparision
<=	Less Than or Equal To	Comparision
and	Returns True if both conditions are true	Logical
or	Returns True if at least one condition is true	Logical
not	Inverts the Boolean value (True becomes False, and vice versa)	Logical

Conditional Statements

+++

If Statement checks a condition.

- If the condition evaluates to True, the block of code under the if statement is executed.
- If the condition evaluates to False, Python skips the if block.



```
1 if condition:
2     # Block of code to execute if
    the condition is True
```

Conditional Statements

+++

elif statement (short for "else if") allows you to check multiple conditions after the if statement.

- If the condition in the if statement is False, Python will evaluate the condition in the elif statement.
- The code inside the elif block runs only if the elif condition is True.



```
1 if condition1:
2     # Block of code to execute if condition1 is True
3 elif condition2:
4     # Block of code to execute if condition2 is True
```


Conditional Statements



else statement provides a block of code that runs if none of the conditions in the if or elif statements are True.

It serves as a fallback option when all the previous conditions fail.



```
1 if condition1:
2     # Block of code to execute if condition1 is True
3 elif condition2:
4     # Block of code to execute if condition2 is True
5 else:
6     # Block of code to execute if none of the above conditions are True
```