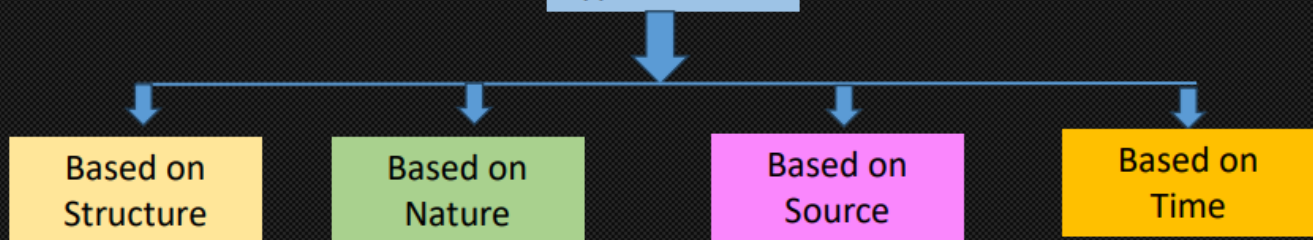


Data

Data is the fuel for decision-making in Data Science, Machine Learning, and Artificial Intelligence. Data refers to raw facts, figures, or information that can be collected, stored, and processed to gain meaningful insights. It can be in the form of numbers, text, images, videos, or any other digital format.

Types of Data



Based on Structure

Day 2 – Variables & Data Types

Structured Data

Data that follows a fixed format and is stored in databases (tables, rows, and columns)

Real-life Examples:

- Customer details in an Excel sheet (Name, Age, Email)
- Bank transactions data (Date, Amount, Account Number)
- Student records in a school database

Mostly stored in relational databases like MySQL, PostgreSQL, etc.

Unstructured Data

Data that does not have a predefined format

Real-life Examples:

- Social media posts (Tweets, Instagram captions)
- Images and videos on YouTube
- Emails with different formats
- Scanned handwritten documents

requires techniques like NLP (Natural Language Processing) and Image Processing to analyze.

Semi-structured Data

Partially structured with tags and metadata.

Real-life Examples:

- JSON data from APIs
- HTML web pages
- Log files generated by web servers

Examples include XML, JSON, and NoSQL databases.

Based on Nature

Day 2 – Variables & Data Types

Quantitative Data (Numerical Data)

Data that represents measurable quantities.

Further divided into:

Discrete Data → Countable numbers (e.g., Number of students in a class).

Continuous Data → Measurable numbers with decimals (e.g., Temperature, Weight).

Real-life Examples:

- Age of a person (Discrete: 25, 30, 40)
- Height of students (Continuous: 5.4ft, 6.1ft)
- Salary of employees (Continuous: ₹50,000, ₹1,00,000)

Qualitative Data (Categorical Data)

Descriptive data that represents categories or labels.

Real-life Examples:

- Gender (Male, Female, Other)
- Blood Group (A, B, AB, O)
- Customer feedback (Good, Average, Bad)

Based on Source

Primary Data

Data collected firsthand by an individual or organization for a specific purpose.

Real-life Examples:

- Surveys conducted by companies for customer feedback
- Experiments in a laboratory
- Sensor readings from IoT devices

Secondary Data

Data collected by someone else and used for a different purpose.

Real-life Examples:

- Census data published by the government
- Market research reports
- Online datasets from Kaggle, UCI, etc.

Based on Time

Static Data

Data that does not change over time. Used in historical analysis.

Real-life Examples:

- Books in a library
- Population data of 2001
- Past temperature records

Dynamic Data

Data that changes over time. Used for real-time monitoring and predictions.

Real-life Examples:

- Stock market prices
- Live weather updates
- Online streaming analytics (YouTube views)

Variables & Data Types

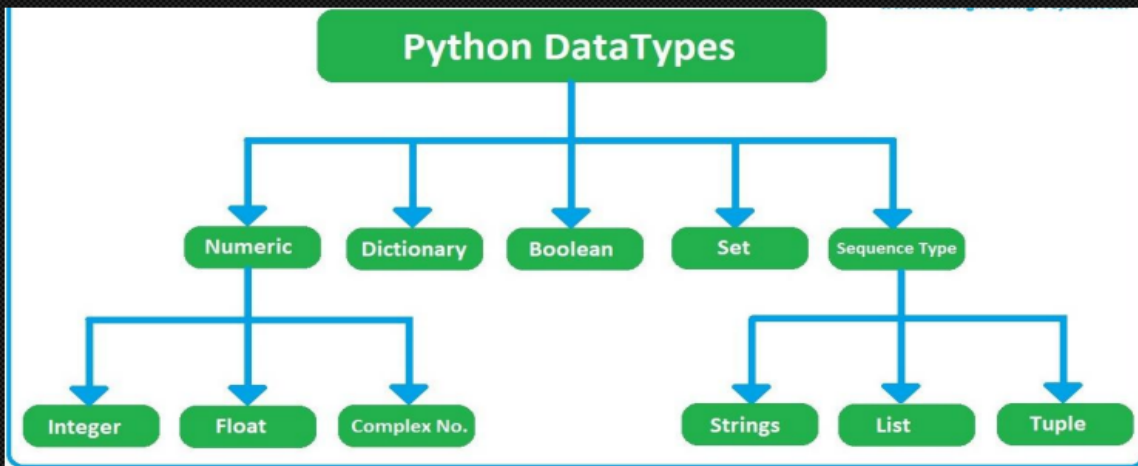
Data Types in Python

Example - e-commerce company Customer Data

NAME	NUMBER OF PRODUCT	PHONE NO VERIFICATION	COST RS	ORDER DETAILS
Anita baghel	5	TRUE	1652.35	Stationary
Ranu Maheshwari	12	FALSE	2564.356	Clothes, Hair care
Ram Kapoor	23	TRUE	100.00	Book
Sunny Sharma	4	FALSE	6583.698	Electronics,
Ayushi gupta	8	FALSE	2156.45	Clothes
Preeti Verma	1	TRUE	3568.124	Electronics, Cosmetics

Variables & Data Types

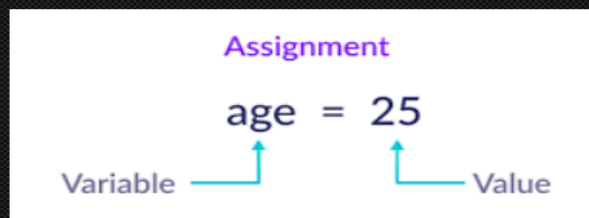
Data Types in Python



Variables & Data Types

Variables in Python

Variables in Python are like containers that hold data values. When you assign a value to a variable, you're essentially putting that value into the container. In Python, variables are used to store data values. They are created when you assign a value to them using the assignment operator (=)



Variables & Data Types

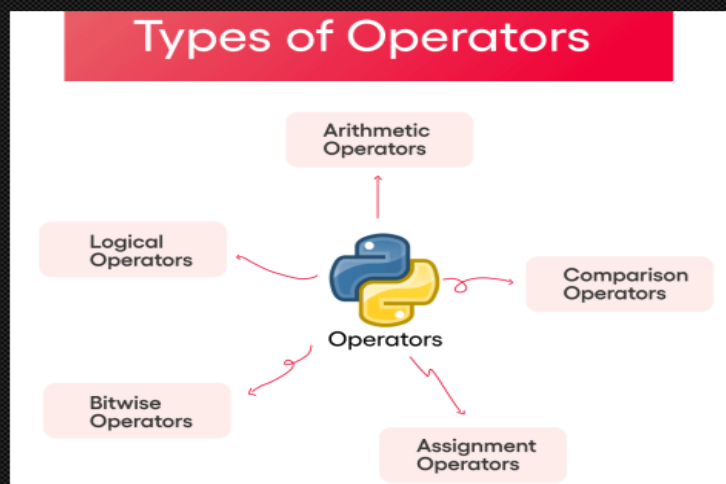
Variables in Python

Rules for Python variables:

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)
- A variable name cannot be any of the Python keywords.

Operators in Python are special symbols that perform specific actions on variables or values. They allow you to carry out tasks like addition, comparison, and logical operations in your code. Operators make it easier to write concise and readable code by providing shorthand notations for common tasks.

Types of Operators



Arithmetic Operators

Operator	Meaning	Example
+	Addition	$4 + 7 \longrightarrow 11$
-	Subtraction	$12 - 5 \longrightarrow 7$
*	Multiplication	$6 * 6 \longrightarrow 36$
/	Division	$30 / 5 \longrightarrow 6$
%	Modulus	$10 \% 4 \longrightarrow 2$
//	Quotient	$18 // 5 \longrightarrow 3$
**	Exponent	$3 ** 5 \longrightarrow 243$

Operators	Meaning	Example
>	Greater than	8>3=True, 4>9=False
<	Less than	8<3=False, 4<9=True
==	Equal to	(3==4)=False, (4==4)=True
!=	Not equal to	3!=4=True, (4!=4)=False
>=	Greater than or equal to	3>=2=True, 3>=3=True
<=	Less than or equal to	2<=3=True, 2<=2=True

Practice Question 1

Which of the following is NOT a valid variable name in Python?

- A) _my_var
- B) 2nd_value
- C) myVar
- D) var_name

Practice Question 2

Which operator is used for floor division in Python?

- A) /
- B) //
- C) %
- D) **

Practice Question 3

What will be the output of the following code?

```
x = 10  
y = 20  
x, y = y, x  
print(x, y)
```

- A) 10 20
- B) 20 10
- C) 10 10
- D) 20 20