Content Covered

- Variables and its naming convention
- Data Type
- Numeric data type
- Numeric data operations
- String
- String Operators
- Type Casting and Checking

Variables

- Containers to hold value in memory
- Prior Declaration not required
- Can be reassigned
- Constants are also stored like variables
- Example:
  - course_name = "Data Science“
  - course_duration = 3
  - COURSE_BATCH = 2025 # constant

Name Convention

- Can contain only Alphabet, Number & Underscore
- Cannot contain Number at initial
- Can't be inbuilt keywords
- **Should pose proper meaning**
- Case-Sensitive
- Constant named in all-caps

Data Types

- Python Data Types:
  - Numeric (+ve, -ve, float or complex)
  - Boolean
  - String
  - Collection (List, Tuple, Set)
  - Dictionary
- Each has own purpose & operation
- 3 / 2 has meaning but not 'cow' / 'dog'

Numeric

- For storing Number and perform numeric calculations
- We can store +ve, -ve number, float & complex
- We can also store number in other number system.

  E.g.: 0b10, 0xF, 0o10

Operators

- Perform operation on Operands
- Numeric Operators:
  - + => Addition
  - - => Subtraction
  - * => Multiplication
  - / => Division
  - // => Floor Division
  - % => Modulus (Remainder)
  - ** => Exponent (Power)

String

- For storing Texts
- Stored with single or double quotes
- Multiline string stored in triple quote
- We use \ to escape special character
- Example:
  - course = 'Python’
  - place = "Broadway's Classroom"

Lab

Store below texts in variable & display them

1. Your Name
2. This is Ram's Phone
3. Ram said, "I'll pass this exam"
4. Dear Students,

   Welcome to Python Class.
5. Hari said, "I love Python"
6. \t and \n are special characters
7. \ is special character in Python.

String Methods

Indexing

- Extract letter from specific position
- Starts from 0, –ve index is also allowed
- Example:
  - sample_text = "hello“
  - print(sample_text[1])
  - print(sample_text[-1])
  - var_1 = sample_text[-2]

String Methods

Slicing

- Extract substring from string
- Uses by bracket notation [::]
- First value inclusive, second exclusive
- Example:
  - sample_text = "hello“
  - print(sample_text[1:3])
  - print(sample_text[1:])
  - var_1 = sample_text[:2]
  - var_2 = sample_text[::-1]

String Methods

Repetition

- Used to repeat string value given number of times
- Uses multiplication operator
- Example:
  - sample_text = "hello“
  - print(sample_text * 2)

String Methods

Concatenation

- Used to concatenate two string together
- Uses addition operator
- Example:
  - hello_txt = "hello“
  - world_txt = “world”
  - combined_txt = hello_txt + world_txt
  - print(combined_txt)

String Methods

Len

- Calculates length of string
- Example:
  - hello_txt = "hello world“
  - string_length = len(hello_text)
  - print(string_length)

String Methods

Stripping

- Removes characters from initial and end of a string
- Default character to remove is space
- Example:
  - txt = “ This is to be removed “
  - print(txt.strip())
  - print(txt.lstrip())
  - print(txt.rstrip())
  - print(txt.strip(‘ de’))

String Methods

Change Case of string

- Convert string into different cases
- Functions:
  - .upper()
  - .lower()
  - .swapcase()
  - .capitalize()
  - .title()
- Example: 'I am in tinkune'.upper()

String Methods

Check Case of string

- Check if string is given cases
- Functions:
  - .isupper()
  - .islower()
  - .istitle()
- Example:
  - txt = ‘sample’
  - print(txt.isupper())
  - print(txt.islower())

String Methods

Check type of value string holds

- Functions:
  - .isalpha()
  - .isdigit()
  - .isspace()
  - .isalnum()
- Example:
  - txt = ‘sample’
  - print(txt.isupper())
  - print(txt.islower())

String Methods

Find

- Finds substring within a string
- Returns start position of substring
- Returns -1 if substring is not found
- Example:
  - txt = 'pineapple'
  - print(txt.find('apple'))
  - print(txt.find('orange'))
  - print(txt.find('apple', 1, 8))

String Methods

Index

- Finds substring within a string
- Returns start position of substring
- Returns error if substring is not found
- Example:
  - txt = ‘pineapple’
  - print(txt.index(’apple’))
  - print(txt.index(’orange’))
  - print(txt.index(’apple’, 1, 8))

String Methods

Replace

- Replace substring in a string
- By default, replaces all occurrence
- Example:
  - txt = ‘pineapple’
  - txt.replace(‘p’, ‘o’)
  - txt.replace(‘p’, ‘o’, 2) # replace 2 p’s
  - txt.replace(‘apple’, ‘orange’)

String Methods

Count

- Count number of occurrence of substring in a string
- Start & end position can be specified
- Example:
  - txt = ‘pineapple’
  - txt.count(‘a’)
  - txt.count(‘e’)

String Methods

Split

- Separates character of string with given delimiter
- Default delimiter is space
- Example:
  - txt = “data science”
  - txt.split(“a”)
  - txt.split(“”)

String Methods

Join

- Puts specified substring between characters of string
- Example:
  - txt = “hello”
  - ‘.’.join(txt)
  - ‘tst’.join(txt)

String Methods

.startswith & .endswith

- Checks if string start or end with given substring
- Start and end position can be given
- Example:
  - txt = ‘python’
  - txt.startswith(‘p’)
  - txt.endswith(‘o’)

String Methods

Ord and Chr

- ord gives ascii value of a character
- chr returns character of given ascii
- Example:
  - ord('a')
  - chr(65)

Type Casting

- Data type of can be known with type
- Data type can be converted
- Functions: int, str, bool, float, list, set, tuple
- Example:
  - num = 6
  - type(num)
  - print("The number is" + str(num))
  - type(num) == int

Exercise