Day 4

Date 10 June 2024

Daily Report

Today's traning based on immutable bult in data type- string and operators in Python.

Topic covered in todays lesson

String in python

strings are built in data type in python.

syntax:-

```
str_name = " element "
str_name = ' element '
str_name = '''element'''
```

Given above any method can be used to write a string.

String Method

There are number of String method in python. but here are some important one.

- upper() - to convert string into uppercase

  ```
  # str.upper()
  ```

- lower() - to convert string into lowercase

  ```
  # str.lower()
  ```

- capitalize() - to convert first letter of string into uppercase letter

  ```
  # str.capitalize(0)
  ```

- title() - to convert all first letters of string into uppercase

  ```
  # str.title()
  ```

- swapcase() - to swap lowercase and uppercase letters

```
# str.swapcase()
```

- endswith() - to tell whether its end with specific string or not.

```
# endswith()
```

- center() - to give equal space front front and end of the string

```
# str.center()
```

- expandtabs() - to give tab spaces

```
# str.expandtabs()
```

- index() - used to find element at given index

```
# str.index(index)
```

- find() - used to find index of specific element

```
# str.find(element)
```

operators in python

1. Arithmatics - already done
2. Comparison Operators - already done first day
3. Logical Operators-
    - and - Return True if both operators are True

    ```
    # a>1 and a<9
    ```

    - OR - Return True if one of the operator is True

    ```
    # a>1 or a<9
    ```

    - NOT - Return True if given operator is False

```
# not a>1
```

4. Bitwise operator-
   - Bitwise and operator (&) - Return True if both operand are 1.

     ```
     # a & b
     ```

   - Bitwise or operator (|) - Return True if one of the operand is 1.

     ```
     # a | b
     ```

   - Bitwise not operator (^) - Return True if operand is one of the operand is 0.

     ```
     # a^b
     ```

5. Assignment Operator -
   - Assignment Operator (=) - Assign the value of the right side of the expression to the left side operand

     ```
     # c = a + b
     ```

   - Addition Assignment Operator (+=) - Add right side operand with left side operand and then assign the result to left operand

     ```
     # a += b
     ```

   - Subtraction Assignment Operator (-=) - Subtract right side operand from left side operand and then assign the result to left operand

     ```
     # a -= b
     ```

   - Multiplication Assignment Operator (*=) - Multiply right operand with left operand and then assign the result to the left operand

     ```
     # a *= b
     ```

- Division Assignment Operator (/=) - Divide left operand with right operand and then assign the result to the left operand

```
# a /= b
```

- Modulus Assignment Operator (%=) - Divides the left operand with the right operand and then assign the remainder to the left operand

```
# a %= b
```

- Floor Division Assignment Operator (//=) - Divide left operand with right operand and then assign the value(floor) to left operand

```
# a //= b
```

- Exponentiation Assignment Operator (**=) - Calculate exponent(raise power) value using operands and then assign the result to left operand

```
# a**= b
```

- Bitwise AND Assignment Operator (&=) - Performs Bitwise AND on operands and assign the result to left operand

```
# a &= b
```

- Bitwise OR Assignment Operator (|=) - Performs Bitwise OR on operands and assign the value to left operand

```
# a |= b
```

- Bitwise XOR Assignment Operator (^=) - Performs Bitwise XOR on operands and assign the value to left operand

```
# a ^= b
```

- Bitwise Right Shift Assignment Operator (>>=) - Performs Bitwise right shift on operands and assign the result to left operand

```
# a >>= b
```

- Bitwise Left Shift Assignment Operator (<<=) - Performs Bitwise left shift on operands and assign the result to left operand

```
# a <<= b
```