

## # Variables:

→ Python variables are simply containers for storing data values. Unlike other languages, such as Java, Python has no command for declaring variable.

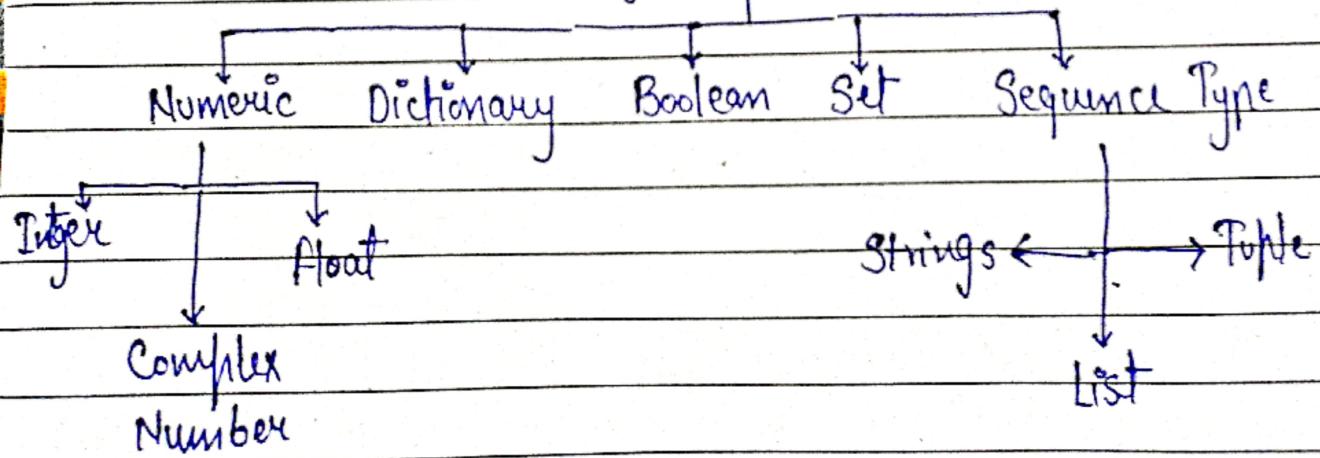
## # Rules for declaring Variables:

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

## # Data Types:

→ Data Types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data.

## Python Data Types



Subject Python

# Integer DataType:- This value is represented by int class. It contains +ve or -ve whole numbers. In Python, there is no limit to how long an integer value can be.

a = 5

print(type(a)) // Int

# Float DataType:- This value is represented by the float class. It is a real number with a floating-point representation. It is specified by a decimal point.

a = 6.0

print(type(a)) // float

# Complex DataType:- A complex number is represented by a complex class. It is specified as (real part) + (imaginary part)j. for ex:- 2+3j

a = 2+4j

print(type(a)) // complex

# String Datatype:- A string is a collection of one or more characters put in a single quote, double-quote, or triple quote. In Python, there is no character data type. Python, a character is a string of length one. It is represented by str class.

```
str = "Hello"
```

```
print(type(str)) // String
```

⇒ Methods in String Data Type :-

str = "Hello how are you"

print(type(str))

→ Type is a built-in function that returns the type of data stored in the program's objects/variables.

## [] // Indexing

- ① Positive (forward) [0...x)
- ② Negative (Backward) [-1...-x)

print(len(str))

→ The len() function returns the no. of characters in the string.

x = str.count("how")

→ The count() method returns the number of times a specified value appears in the string.

print(str.upper())

→ The upper method returns a string where all characters are in uppercase.

print(str.lower())

- The lower() method returns a string where all characters are lower case.

print(str.title())

- The title() method returns a string where the first character in every word is upper case.

print(str.capitalize())

- The capitalize() method returns a string where the first character is upper case, and rest is lower case.

x = str.replace("Hello", "Nice")

- The replace() method replaces a specified phrase with another specified phrase.

x = str.find("how")

- The find() method finds the first occurrence of the specified value. The find() method returns -1 if the value is not found.

\* The find() and index() are same but the index() raises exception if phrase not found.

x = str.startswith(value, start, end)

- The startswith() method returns True if the string starts with the specified value, otherwise False.

Day 2

Date: 04/07/24

Subject: Python

MON TUE WED THR FRI SAT SUN

x = str.endswith(value, start, end)

→ The `endswith()` method returns True if the string ends with the specified value, otherwise False.

x = str.split(separator, maxsplit) / str.split()

→ The `split()` method splits a string into a list. You can specify the separator, default separator is any whitespace.

x = "banana"

y = x.strip()

print(y) // banana

→ The `strip()` method removes any leading, and trailing whitespace.

x = str.center(20, "0")

→ The `center()` method will center align the string, using a specified character (space is default) as fill character.

x = str.rfind("how")

→ The `rfind()` method finds the last occurrence of the specified value. The `rfind()` method returns -1 if the value is not found.

# Boolean Data Type : In any variable only True or False is stored then it is defined as Boolean Datatype.

var1 = True

var2 = False

# List Data Type : []

- List is an alternative of array, collection of variable but not similar datatype.
- List is of Referential nature { size is not fixed & can store multiple datatypes }

ls = [1, 2, 'upHairs', 3.6, True]

- Indexing is same as String.

- List is Mutable { changeable, can be modified }

=> Methods in List :

ls.append('ritu')

- The append() method is used to add the item in list & inserts item at last.

ls.pop()

- The pop() method is used to remove item from the last in list.

ls.insert(1, 'value')

index

↑  
item to insert at given index

→ The `insert()` method is used to insert item at given index and next element is shifted.

`st.remove('value')` value to remove from list

→ The `remove()` method is used to remove the item from the list.

`del ls[2]` Index at which to delete item

→ The `del` method is used to remove item from list by passing index.

`print(ls.count('value'))`

→ The `count()` method is used to count number of times the item is present

`ls.reverse()`

`print(ls)`

→ The `reverse()` method is used to reverse the list.

`ls.sort() # Ascending`

`ls.sort(reverse=True) # Descending`

→ The `sort()` method is used to sort the items of the list

`print(min(ls))`

→ The `min()` is an aggregate method which finds minimum value from list.

`print(max(l))`

- The max() method is used to find maximum value from list.

`print(sum(l))`

- The sum() method is used to find sum of all items in list.

`ls1.extend(ls2)`

- The extend() method is used to concatenate two lists.

`ls.clear()`

- The clear() method is used to clear the list.

`print(ls.index('value'))`

- The index() method is used to find the index of specific item in list.

## # Tuple Data Type :- ()

- Tuple datatype are same as list but they are immutable {unchangeable}

`tp = (2, 3, 4, 'UpHairs', True, 7, 6)`

⇒ Methods in Tuple :-

`print(tp.count(3))`

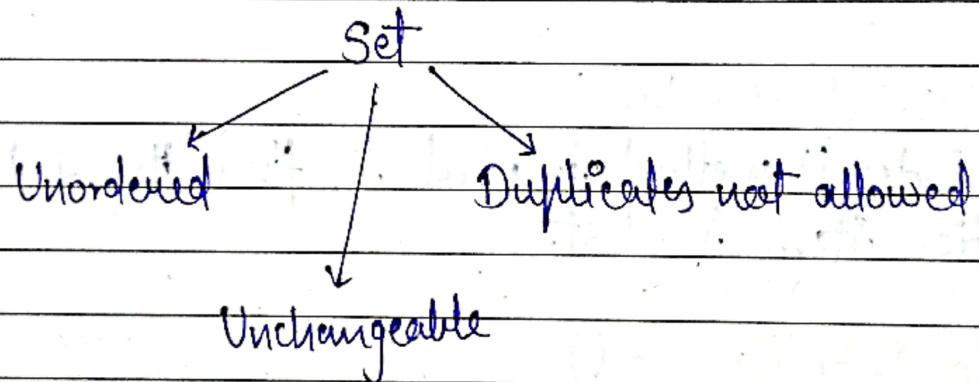
→ The count() method returns the no. of times specified value appears in tuple.

`print(tp.index(8))`

→ The index() method find the first occurrence of specified value and raises exception if the value is not found.

# Set Data Type :- {}

→ Sets are used to store multiple items in a single variable.



⇒ Methods in Set :-

`st.add(500)`

→ The add method is used to add item in the set

`st.remove(54)`

→ The remove method is used to remove item from set &

Subject Python

gives error if item is not present in set.

### st1.discard(st2)

- The discard() method same as remove but don't give error if item is not present.

### st1.update(st2) # st1 + st2

- The update() method is used to add items in set.

### st1.intersection(st2)

- The intersection() method is used to find common items in two sets

### # Dictionary Data Type: {}

- Dictionary datatype contain

pairs = (Key:value)

for ex:

marks = {'mohit':90, 'rohit':96, 'Keshav':69}

- Duplicate Key is not allowed but duplicate value is allowed.

- The key is passed in [] brackets to get the value.

print(marks['mohit'])

Day 4

Date: 08/07/24

Subject Python

MON TUE WED THR FRI SAT SUN

marks['mohit'] = 65

- Updation of the value in Dictionary
- If the key is not present then it will be inserted in the Dictionary

marks['rahul'] = 40 # Insertion

⇒ Methods in Dictionary Data Type :-

marks.clear()

- The clear method removes all the elements from the Dictionary.

x = marks.copy()

print(x)

- The copy method returns a copy of the specified dictionary.

print(marks.get('Keshav'))

The get() method returns the value of the item with the specified key.

print(marks.keys())

The keys() method returns a view object. The view object contains the keys of the dictionary, as a list.

marks.pop('mohit')

The pop() method is used to remove the pair by passing the key of that pair.

Subject Python

`marks.update {"manvik": 99}`

The update() method inserts the specified items to the dictionary.

`print(marks.values())`

The values() method contains the values of the dictionary as a list.

# Type casting :- It is the method to convert the Python variable datatype into a certain datatype in order to perform required operation by users.

There can be two types of Type Casting in Python:

1) Implicit Type Conversion      2) Explicit Type Conversion

In this, Python converts the datatype into another datatype automatically. Users don't have to involve in this process.

`a = 7 // Int`

`b = 3.0 // Float`

`c = a + b`

`print(type(c)) // Float`

In this, Python needs user involvement to convert the variable datatype into the required datatype.

`a = 5 // Integer`

`n = float(a)`

`print(n) // 5.0`

`print(type(n)) // Float`