

Python Notes

- Vaishnavi abbugari

Variables: Variables are entities of programme that hold a value.

Eg: $x = 100$

Here, x is a variable and the number the variable is holding is a value.

Data Types:

Integers (int) : Any number without a decimal point.

FLOATS: Number with decimal point.

Strings : Collection of characters ("str")

Data Types Specific to Python:

List : collection of values in one variable

Eg: [14, 23, 68, 90]

Tuple: Collection of values in one variable

Eg: (23, 48, 68, 72)

inspuddi iorvior -

Sets: A Set is a collection which contains only unique values in unordered, unchangeable and unindexed way.

Eg: $\{10, 20, 20, 10, 30, 40, 30\}$

Set = $\{10, 20, 30, 40\}$ (or) $\{40, 30, 10, 20\}$

Dictionary: Dictionary is used to store the data values in key : Value pairs

Eg: $D = \{"Name": "Vaishnavi", "Age": 22\}$

Rules for naming variables

1. Variable name must begin with an alphabet (or) an underscore
2. The first character can be followed by alphabets, numbers or underscore
3. Variable names are case sensitive
4. Reserved words cannot be used as a variable names

Arithmetic Operations :

(+) = Addition , (-) = Subtraction

(%) = mod , (*) = multiply

(/) = Division and values in float.

(//) = Division and values in Int

String Operations : (Slicing)

s = ("Vaishnavi Abbugari")

s[0] = "V"

s[0:10] = "Vaishnavi"

s[5:] = "navi Abbugari"

s[3:7] = "shnav"

s[::-1] = "iragubba ivanhisia"

s[-3] = "a"

s[-5:-1] = "ubbA ivanhisia"

s[2:9:0] = "iaishnavi"

String (Methods) :

• Capitalize() - It will capitalize the 1st letter.

• Lower() - Lowers the values

• Upper() - Uppers the values

• Center(10, "*") - Returns centered string.

- Count ("L") - count of character
- index ("O") - Index of character
- find ("O") - Index of word
- replace ("H", "W") - replacing
- Split ("/") - split the characters
- isalnum () - all numbers & Alpha
- isnumeric () - all numbers ?
- islower () - lower caps ?
- isupper () - upper caps ?
- None of these methods change the original value, to change the value permanently you have to save in a new variable.

String Concatenations :

str1 = ("Good")

str2 = ("Morning")

str3 = str1 + str2

str3 = Good Morning

str = {{}, {}, {} . format ("str1", "str2", "str3")}

str = "Good Morning".

Lists : Collection of multiple data types

L = [1, "A", 10.2]

Operations on lists :

- if you want to print 100 zero's you can't just keep on typing them instead you can use this operation.

Z = [0] * 100

Z = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

- concatenating 2 variables

x = [1, 2, 3]

y = [4, 5, 6]

z = x + y

z = [1, 2, 3, 4, 5, 6]

- unpacking element

list ("hey there")

list = ["h", "e", "y", " ", "t", "h", "e", "r", "e"]

Methods on lists :

- append

- extend

- insert

- count

- clear
- index
- remove
- pop
- reverse
- sort
- copy

Built in functions in list :

min()

max()

len()

sum()

Tuples: collection of immutable heterogeneous python objects

t = (1, 10, "Game", 10.2)

t = 1, 10, "Game", 10.2

- tuple may use or may not use brackets and parenthesis
- we can change list into tuple if we don't want to change the values in list, however we cannot

convert tuples into list

Nesting tuples in a list :

list = [(1, 2, 3), (4, 5, 6)]

list - [(1, 2, 3), (4, 5, 6)]

- As tuples are immutable we can use nested tuples for operations we can modify the tuples in a list

Dictionaries : collection of data stored as a pair of key & value

D = {"name": "Vaishnavi"}

- Key's should be unique.

Methods in dict :

- copy()
- clear()
- fromkeys()
- items()
- get()
- keys()
- pop()
- values()

- update()
- setdefault()
- popitem()

Sets: It is an unordered collection of unique elements

$$S = \{10, 10, 20, 30, 40, 40\}$$

$$S = \{10, 20, 30, 40\}$$

Methods in Set:

- | | |
|--------------|-------------------|
| • add() | • Intersection() |
| • discard() | • difference() |
| • remove() | • issubset() |
| • pop() | • issuperset() |
| • union() | • isdisjoint() |
| • update() | |

• add() - add the values to the set

• discard() - removes the given character

• remove() - it also removes the given character

But the difference between `discard()` and `remove()` is `discard` doesn't give any error if the given argument is not in the set. whereas `remove()` throws an `KeyError`.

- `union()` - Gives all values of 2 sets
- `update()` - updates the set with union
- `intersection()` - Gives the common values within 2 sets
- `difference()` - Gives uncommon values
- `issubset()` - another set contains this set?
- `issuperset()` - another set contains set
- `isdisjoint()` - null intersections.

Mutable & Immutable

Mutable : We can change the values

Immutable : We cannot change the values.

Mutable data types : list , Dictionary
Set

Immutable data types : string , tuples
Integer .