Learn Coding | Python

(core python)

Topic we will cover in core python

- Introduction to python
- Installation of python and vscode
- Reserved words and Identifiers
- Data Types
- Type casting
- Operators
- Input and output statements
- Command line arguments
- control flow
- pass, del and None keyword
- functions
- Modules

Introduction to Python

Chapter 01

What is python?

-general purpose high level language
-developed by Guido van Rossam in 1989
national research institute neatherland
-publically available : feb 20 1991

Why the name python?

"monty flying python circus" broadcast by BBC in 1969-74.
Comedy show, it brings smile on the face.
Name taken from this show

What Features taken from which languages ?

- functional progammng - c
- object oriented - c++
- modular programming - modula-3
- scripting - perl, bash scripting

Applications of python

-web development
-game development
-iot programming
-machine learning
-Artificial Intelligence
-Network programming
-db programming

Features of python

-Simple and easy to learn
-freeware and open source
-platform independent
-potable
-rich library
-extensible
-embedded
-dynamically types
-interpreted

Limitations of python

-mobile Application
-performence is not upto the mark

Flavors of python

- jython
- pypy
- anaconda python
- Cpython (default version)

Installation of Python

Chapter 02

To install python :

- Visit the below site:
  https://www.python.org/downloads/
- Click on download then install the python

To install vscode :

- Visit the below site:
  https://code.visualstudio.com/download
- Choose your OS download will start automatically

Reserved Word & Identifiers

Chapter 03

Reserved Words:

- words which convey special meaning to the Identifiers
- Only 33+ keyword in python
- Reserved words also known as keyword
- Only True, False, None starts with uppercase
- List of keywords are:-

'False', 'None', 'True', 'and', 'as', 'assert', 'async',
'await', 'break', 'class', 'continue', 'def', 'del',
'elif', 'else', 'except', 'finally', 'for', 'from', global',
'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not',
'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield'

Identifiers:

- A name in python program is called Identifier
- It can be class name, function name, variable name, module name.

Rules to define Identifiers :

- ❑ Allowed character:
  - alphabet (uppercase or lowercase)
  - digits (0-9)
  - underscore (_)
- ❑ should not be start with digit
- ❑ we cant use Reserved word (keyword)
- ✓ Identifiers are case sensitive

Data Types

Chapter 04

Data Types:

- data types represent the type of data present inside a variable
- we do not need to specify the data type explicitly, based on values types allocated automatically.
- Python is a dynamically typed language.

Primary Data Types:

1. int
2. float
3. complex
4. bool
5. string

Note: All primary data types are immutable in nature.

Inbuilt Data Types:

1. int
2. float
3. complex
4. bool
5. string
6. bytes
7. bytesarray
8. range
9. list
10. tupple
11. set
12. frozenset
13. dict
14. None

SOME IMPORTANT FUNCTIONS

type():-

it returns the data type of the given values

id():-

it return the id of the given values

int

- To represent the integeric values.
- We can store integeric value in various base.
  decimal, binary, hexadecimal, octal
- Values with base 10 is default.
- Python can receive value in any base but store value in decial form only

```
binary  = '0B'   '0b'
hex     = '0x'   '0X'
octal   = '0o'   '0O'
```

float

- to represent the floating point values
- we can also store numbers in exponential forms
- For e.g.
  10e2 (means 10 to power 2 i.e 1000)

complex

- numbers of form a+bj is called complex number.
  a => real part (integer/float) (hex, bin, dec, oct)
  b => imaginary part (integer/float) (dec)

- To important attribute of complex object:
  real ==> real part of numbers
  imag ==> imaginary part

bool

- It represent Boolean values
- Supports only two values i.e. True, False
- Python internally store :
  - True as 1
  - False as 0

string

- String is sequence of characters.
- It is immutable in nature.
- The most used data type in any programming language.
- we can write single quote inside double quote
- double quote inside single quote
- triple quote for multi-line string

❑ Escape sequence characters has specific meaning inside string.

\n -> new line
\t -> horizontal tab
\c -> carriage return
\v -> vertical tab
\\ \" \v etc.

String Indexing

- Indexing is used to get the value at specific index.
- Forward indexing:
  left to right - start with 0
  right to left - start with -1

String Slicing

- Slicing is used slice the string (get the part of the string).
- [starting_position : ending_position : step]

Formating of String

- {} -> Replacement operator (it substitute the variable with the value)

- f -> formatting string
- r -> raw string (treat the string as it is written)

Some Important Methods of String

- **split("pattern")** - split into several part based on pattern and return a list.
- **("pattern").join(sequence)** - join the string based on the pattern
- **find("pattern","start", "end")** - find the position of our pattern in the string. if pattern not found then it will return -1
- **index("pattern","start","end")** - same as find but it will generate error if pattern not found
- count("pattern", "start", "end") :- return no. of occurence of specified pattern

Some Important Methods of String

- strip() - remove leading and trailing speaces
- replace("old patter", "new pattern") - replace older pattern with new one.

Changing case of the string

upper() -> convert to uppercase
title()
lower()
capitalize()

Some Important Methods of String

- strip() - remove leading and trailing speaces
- replace("old patter", "new pattern") - replace older pattern with new one.

Changing case of the string

upper() -> convert to uppercase
title()
lower()
capitalize()

Checking the case of the string

- isupper()
- islower()
- isalpha()
- isalnum()
- isspace()

range

- It is use to store the sequence of integeric numbers.
- range(begining, ending, step)

Type Casting

Chapter 05

Type casting

- we can convert one type into another type . this conversion is known as typecasting.

- int()
- float()
- complex()
- bool()
- str()
- list()
- tuple()
- set()
- dict()

Operator

Chapter 06

Operator

operator are symbols that perform certain task

- Arithmetic Operators:
  +, -, *, /, //, %, **
  helps to perform arithmetic operations

- Relational or comparison operator:
  <, <=, >, >=, ==, !=
  helps to perform comparison between two object

- Logical operator:
  and, or, not
  helps to perform logical operations

- Bitwise operator (applicable for int and bool value only)

  & -> bitwise and (if both 1 then 1 else 0)

  | -> bitwise or (if any 1 then 1 else 0)

  ^ -> if both same then output 0 else 1

  ~ -> complement

  >> -> right shift

  << -> left shift

  helps to perform bitwise operations

- assignment operator

  =

  helps to assing value to a variable

- special operator:
  a) identity operator
     is, is not
     helps to verify whether two object is same or not

  b) membership operator
     in, not in
     helps to verify whether an object is the part of any sequence

Input & Output Statement

Chapter 07

input()

It is used to take input from the user.
It always return string.
Syntax: variable_name = input(“your message here”)

print()

It is used print the output on the console
Syntax: print(“your message here”, sep=“”, end=“”)
where everything is optional

Flow Control

Chapter 08

Flow Control

flow control describes the order in which statements will be executed at runtime

- ❑ Conditional statements
- if - else

if the condition evaluate to true then if block will execute otherwise else part.

- ❑ Iterative statements
- for

If you know in advance how many times block will execute then go with the for loop

- while

Use it If you want to execute the block until condition remains true

- Transfer statements
  - continue
    - it skip the iteration
  - break
    - it break the flow and get out of loop
  - pass
    - if we do not want to define define the body then we can use pass

Note: we can use “else” also with the loop in python.

Else part will execute if the block within the loop completes its execution without break.

len(), del(), None

Chapter 09

del()

It is used to delete the object.

len()

It returns the length of the object.

None

It is similar to Null.
It means nothing

Inbuilt data types

Chapter 10

List

-duplicates are Allowed
-heterogeneous object are Allowed
-slicing and indexing are Allowed
-mutable in nature
-order are maintained
-values are stored with in square brackets []

Tuple

- values stored within parenthesis ().
- parenthesis are optional
- it is exactly same as tuple but it is immutable.

set

-if we want to represent a group of unique value as a single entity then we should go for set
-duplicates are not Allowed
-insertion order is not preserved
-indexing and slicing not work
-heterogeneous elements are Allowed
-mutable in nature
-values are stored within curly braces { }.

Dictionary

we can use list, tuple, and set to represent group object as A single entity.

if we want to represent a group of object as key-value pairs then we should go for Dictionary.

- do not support indexing and slicing
- insertion order is preserved
- heterogeneous
- mutable
- keys must be unique, but duplicates value are Allowed
- values are stored withing curly braces { key:value }

Command Line Arguments

Chapter 11

Command Line Argument

With the help of command line argument we can pass the input at the time of executing the program.

For this we have to import the sys module.

```
from sys import argv
```

Here argv is a list of values that are pases at the time of program execution

Function

Chapter 12

Function

If a group of statements is repeatedly required then it is not recommended to write these statements every time separately.

We have to define these statements as a single unit and we can call that unit any number of times based on our requirement without rewriting. This unit is nothing but function.

The main advantage of functions is code Reusability

Types of Function

- Built-in Function :
  The pre-defined function which comes with python.
  e.g. len, type, print, del etc..

- User defined function:
  we create to fulfil our business requirement.

Syntax:

```
def function_name(parameter):
    """ doc strintg """
    body
    return value
```

Types of arguments

- Positional:
  - order should be maintained
  - no. of parameter = no. of arguments
- Keyword:
  - order is not important
  - no. of parameter = no. of arguments
  - keyword arguments should follow positional arguments
- Default arguments:
  - order is important
  - no. of parameter may or may not be equal no. of arguments
- Variable length arguments:
  - *args, **kwargs

Modules

Chapter 13

Modules

A group of functions, variables and classes saved to a file, which is nothing but module.
Every Python file (.py) acts as a module.

main objective : code Reusability

- Module aliasing:

---

to improve reality
to prevent name conflicting

#Bonus | Best Practices

Chapter 14

Best Practices:

PEP8 ===> guidelines

snake case naming conversion

firstname X
first_name ----> snake

modules, functions, variable ===> should startwith lowercase

class ===> should startwith uppercase

_ => dummy variable for one time use.

```
for _ in range(10):
        print(_)
```

constant variable should be in upper case.

meaningful of variables

Comment

doc string.....

Thank You

Like Share Subscribe | Learn Coding