

Python 1. Interpreter

- 2. case sensitive so write in lower letter.
- 3. Python 2 types of comments
- (1) single line comment (#)
- (2) multi line

```
'''     '''
```

Python was developed by Guido Van Rossum in 1991.

It is a scripting language.

It is an interpreter based language.

The name of the Python interpreter is PyPy Python.

The extension is .py.

It is a case sensitive language (attA. attA)

The statement delimiter is ENTER key.

Comments:- It is an additional information to the programmer.

Python interpreter does not check any syntax error and does not convert into machine language.

Comments are 2 types:-

1. Single line comment (#)

2. Multiline to ''' or'''

2

in Python, implicitly declare variable & type.

Variable:

It is a data name and memory location or name.

It is a temporary storage location.
Every Variable, occupies a little memory at
Python point of view.

Its value can change during execution of the
programme.

Python allows implicit variable declaration.
Every variable occupies memory at run time.

known as dynamic memory management.

If datatype can't change during execution time
known as dynamic typed language.

variable starts with

5 symbols

1. alph. Alphabate OR -

2. keywords (36) x

3. any length

4. space is not allowed.

It should not be a key word.

It can be any length

It should not contain "space character".

The value characters are a-z, A-Z, 0-9, -

Types of variable

1. scalar variable

2. List

3. Tuple

4. Dictionary

5. Set

Date structures

or
collections

or
Sequenced

Find keyword

→ >> import keyword → called module

>> Point (keyword.kwlist) → property means variable

>> Point (len(keyword.kwlist)) ..

Scalar variable

keeps 1 value i.e. single value

⇒ It holds only one value. The value may be integer, float, string, complex, boolean and none type.

c = 2+3j or 2+3J → called complex variable

x = 'Techno'
y = "Techno"
z = '''Techno'''

} string

x = True { # boolean T ~~and~~ and F must be capital
y = False

x = None # None type N → must be capital.

But x = "2+3j" → This is string.

Escape character \) ⇒

''' OR """ " " " " → Takes multiple lines during variable declaration.

Note → Triple quoted strings mainly designed to store multiple lines of data.

x = "I said, \"Don't talk"

y = 'I said, ~~Don't~~ "don't talk'

4

$x = "Hello python \n It is simple \n It is easy!"$

OR

$y = """Hello python \n It is simple \n It is easy!"""$

✓ How to clear interactive screen :-

`import os
os.system('cls')`

for UNIX: `os.system('clear')`

String concatenation

$x = "Hello"$

$y = "Python"$

$k = x + y$ $\text{Print}(k) = \text{HelloPython}$

$k = x + y$

$\text{Print}(k) = \text{Hello Python}$

Multiplication

$a = "Python"$

$k = a * 3$

$\text{Print}(k) = \text{PythonPythonPython}$

$z = "Python\n"$ $k = z * 3$

$\text{Print}(k) = \text{Python}$

Python

Python

⇒ Output function - `Print()`:

It is a predefine function. It is used to write data on screen.

If you want to print something. Put it inside " " in `Print` function.

Ex:- `Print`

`Print("my name is Haris")`

⇒ my name is Haris

⇒ Input function `Input()`

(i) It is a predefine function. It is used to read from user.

(ii) Every input is read in string format.

• `variables = Input("Prompt...")`

`a = 100`

`type(a)`

`id(a) → address of variable`

`Type()`

It returns given variable datatype.

Syntax : `type(variablename)`

`a = "Technosoft"`

`Print(a type(a)) <Class 'str'>`

`Id()`

It returns given variable address in integer format.

Syntax : `id(variablename)`

`print(id(a))`

6

⇒ `del:`

It is a keyword to delete one or more variables.

`del var1`

`del var1, var2`

`x = "Technosoft"`

`a = 10`

`b = 20`

`del a`

`del a, b`

29/12/20

Type casting / Data conversion functions

① `int(a)` It converts 'a' into ~~string~~ integer

`int()`, `str()`, `float()`, `complex()`, `bool()`.

These functions are used for to convert from one type to another type.

`str(a) ⇒` It converts 'a' into string

`float(a) ⇒` It converts 'a' into float

`complex(a) ⇒` It converts 'a' into complex

`bool(a) ⇒` It converts 'a' into boolean.

`a = 100` } int

`b = 200`

`x = "Python"`

`y = "World"` } String

`k = 300`

`m = 43.786` } float

Print (a+b) # 300
Print (x+y) # PythonWorld
Print (a+a) # Type Error X
Print (a+str(a)) # Python00
Print (a+k) # type Error X
Print (a+int(k)) # 400
Print (a+int(x)) # Value Error X
Print (float(a)) # 100.0
Print (float(w)) # 300.0
Print (float(w)) # Value Error X
Print (cne(m)) # 43
Print (str(m)) # '43.786'
a = 18 / -18 / 1.8 / 'techo' / 2+3j / True
print (bool(a)) # True
a = 0 / False / None
Print (bool(a)) # False
a = 2
Print (Complex(a)) # 2+0j

* We can convert integer to string. But can not convert string to number.

String Manipulation

[] → Indexing

[:] → Slicing

[start : stop : step]

no upper limit means till last

no lower || from beginning

{ Start : stop : step }

no step means 1

no start means 0

no stop means till last

If step negative step means reverse order

`a = "Hello Python"`

`a = "Hello Python"`

0 1 2 3 4 5 6 7 8 9 10 11

Print(a) # complete sentence

Print(type(a)) # <class 'str'>

(2) Print(a[2]) # 3rd char

Print(a[8]) # 9th char

Print(a[0]) # 1st char

Print(a[-1]) # n . Last char

Print(a[2:9]) # 110 Pyt

including 3rd to 9th char
excluding

Print(a[3:10]) # lo Python

Print(a[1:7]) # ello Po 4th to 10th char

Print(a[4:8]) # llo Python 2nd to 7th char

Print(a[1:5]) # Hello 1st to 5th char

Print(a[-6:-1]) # python Last 6 char

Print(a[:7]) # Hello Python

Print(`a[::2]`) # Hello Hlopo, odd char
 Print(`a[1::2]`) # el YHn even char
 Print(`a[:::-1]`) # rohtyT olleH [print Reverse order]

Date - 30/12/20

Note 1:-

strings are immutable (can't be changed)

`a = "Technosoft"`

~~del a[3]~~ X

But `a = a[:3] + a[4:]` h is missing
~~Tech~~ ~~nosoft~~ Print(a) # Technosoft

`a = "Technosoft"`

How to add 'n' in a string

`a[3] = 'n'` X

`a = a[:3] + 'n' + a[4:]`

~~Tec~~ ~~osoft~~

Print(a) # Technosoft

Note 2:-

'are'

Number objects also immutable objects

~~and~~ Number objects does not support indexing and slicing.

`a = 91996642225`

Print(type(a)) # <class 'int'>

10

Print(a[:2])

Print(str(a)[2:]) # '91'

but want only phone #

Print(str(a)[2:]) # '9168422225'

print(int(str(a)[2:])) # gives integer

* Every arguments
are separated by $,$ (comma).

like Print("Hello", name, "nice to meet you,
and you are", age, "years old.")

name = kara

age = 7

* Every value/input is treated as string variable

name = input("What is your name?")

age = input("What is your age?")

Print("Hello", name, "nice to meet you
and you are", age, "years old.")

Q Write a program, accept the ~~int~~ 2 integer

numbers and find sum?

11

```
a = input("Enter number 1: ") #10  
b = input("Enter number 2: ") #20
```

$$c = a + b$$

```
print("Sym:", c) # 1020
```

```
print("a type:", type(a), "b type:", type(b)) # str str
```

101

```
a = input("Enter number 1: ") # 10
```

b=c input(" " " " 2") #20

$$c = \text{cnt}(a) + \text{cnt}(b)$$

```
print("Sum:", c) #30
```

```
print("Sum:", c) #30  
Print ("a type:", type(a), "b type:", type(b))  
# Str Str
```

```
a = int(input("Enter number 1:")) # 10
```

```
b = int(input(" " " " ".2:")) #20
```

Cath

```
C = "ab"  
Print ("Item: ", C) #3.
```

```
Print ("Item: ", c) #3.  
Print ("a type", type(a), " b type:", type(b))  
# ind, stat;
```