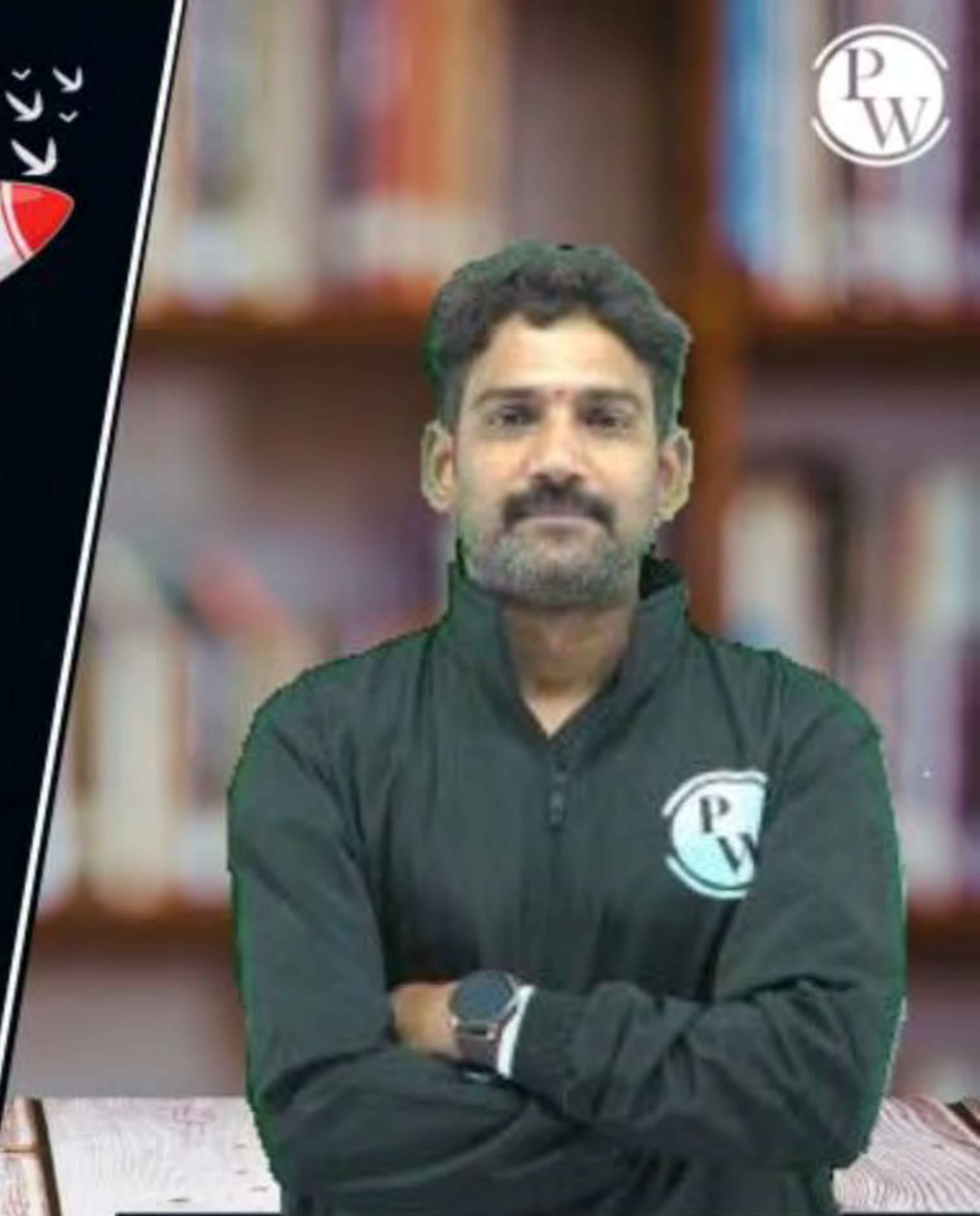
Data Science & Artificial Intelligence

Python For Data Science

Basics Of Python

Lecture No.- 03



By- Satya sir

Recap of Previous Lecture

- input ( ) method
- Print ( ) method
- Data Type
  - Primary : int, float, bool, complex, str, None
  - Secondary : list, Set, tuple, dict

binary : 0b Prefix
octal : 0o Prefix
Hexa Decimal : 0x Prefix

Topics to be Covered

- integers
- float values
- boolean, Complex
- Python Tokens
  - Identifiers
  - keywords
  - Operators.

Topic : Python Fundamental Data Types

Examples

i = 73

j = hex(i) #0x49

k = oct(j) #0o111

l = bin(k) #0b001001001

Print(i, j, k, l)

73 0x49 0o111 0b001001001

$$16\,|\underline{73}$$
4 - 9

$$8\,|\underline{73}$$
$$8\,|\underline{9} - 1$$
1 - 1

Print(47 + 0o37 + 0x6A + 0b10001001)

$$\underline{47}_{\text{Decimal}} + \underline{0o37}_{\text{octal}} + \underline{0x6A}_{\text{Hexa}} + \underline{0b10001001}_{\text{Binary}}$$

o/p = 321

NOTE : operations are Performed on Decimal values only.

$$(37)_8 = 3\times8^1 + 7\times8^0 = 24+7 = (31)_{10}$$

$$(6A)_{16} = 6\times16^1 + A\times16^0 = 96+A = 96+10 = (106)_{10}$$

$$(\overset{7}{1}\overset{6}{0}\overset{5}{0}\overset{4}{0}\overset{3}{1}\overset{2}{0}\overset{1}{0}\overset{0}{1})_2 = 1\times2^0 + 1\times2^3 + 1\times2^7 = 1+8+128 = (137)_{10}$$

$$47+31+106+137 = 321$$

Topic : Python Fundamental Data Types

format specifiers : integer (%d, %o, %x, %b), Strings|char (%s), float (%f (or) %e)

Syntax:

Print(Object [Width] . [Precision width] [format specifier] % (Variables or values))

Example : 1   a = 47

```
Print(' a value in decimal is %d '%(a))   # a value in decimal is 47
Print(' a dec= %d, a octal= %o, a Hexa= %x' %(a, a, a))   # a dec= 47, a oct= 57,
                                                           a Hex = 2f
```

Example 2 :   a = 23
b = 14.47
c = "Student"

```
Print('a= %d  b= %f  c= %s' %(a,b,c))   # a = 23  b= 14.470000
                                          c = Student
```

Topic : Python Fundamental Data Types

float type

- A number with Precision (fractional Part) (or) decimal Point

- Ex: 3.14, 12.37, 0.00023, 147.23103, 72.0, 43.

- Ex:

```
x = 72
y = 72.0
z = 72.
Print ( type(x)) # <class 'int'>
Print ( type (y) ) # <class 'float'>
Print ( type (z)) # <class 'float'>
```

integers operation float => Result in float type

Ex:

```
i = 12
j = 14.
K = i+j
Print(K, type(K))
# 26.0, <class 'float'>
```

Topic : Python Tokens

bool type : boolean Values : Two Pre-defined Constants : True, False → capital letters

Ex:

```
a = 'True'
b = False
c = True
d = "False"
                    e = false  # Error, Invalid assignment

Print (type(a), type(b), type(c), type(d))

<class 'str'> <class 'bool'> <class 'bool'> <class 'str'>
```

Complex Type

Complex Numbers :

$$\begin{tikzcd} i \arrow[d] & + & c*j \arrow[d] \\ \text{Real Part} & & \text{imaginary Part} \end{tikzcd}$$

Ex: 2+3i

```
a = 42
b = 23+10j
Print(complex(a), int(b))
# 42+0j  23
```

Topic : Python Tokens

Token ? The smallest (or) basic individual Element of a Program.

⇒ Program is set of Tokens

Example :

```
def f( ):
    i = int(input())
    j=9
    k= i*j
    Print('k value is', k)
```

Sample.py

Tokens		Type
f int i input j k Print		Identifiers
= ( ) *		Operators
def		Keyword
9	constants	Numeric Constant
'k value is'		String Constant
, :		Seperator (or) special Symbol

Python Tokens

1) Identifiers

2) Keywords

3) Operators

4) Constants

5) Seperators / Special symbols

Identifiers

$\underset{\text{Identifier}}{\underline{\text{vignesh}}} = \underset{\text{string}}{\underline{\text{'vignesh'}}}$

- Name of any Programming Element [Variable | $\overset{\text{list/}}{\text{array}}$ | set | function | dictionary | tuple | module | class | Package | File ---]

Rules for Identifiers

(1) Should start with either alphabet (or) Underscore Symbol

Ex: a, _a, apple, abc, _7b valid

7a, &a, @xyz, +a Invalid

(2) No Space is allowed

(3) No Symbol in the Name Except Underscore Symbol.

(4) Not a keyword.

Topic : Python Tokens

If = 7 ✓

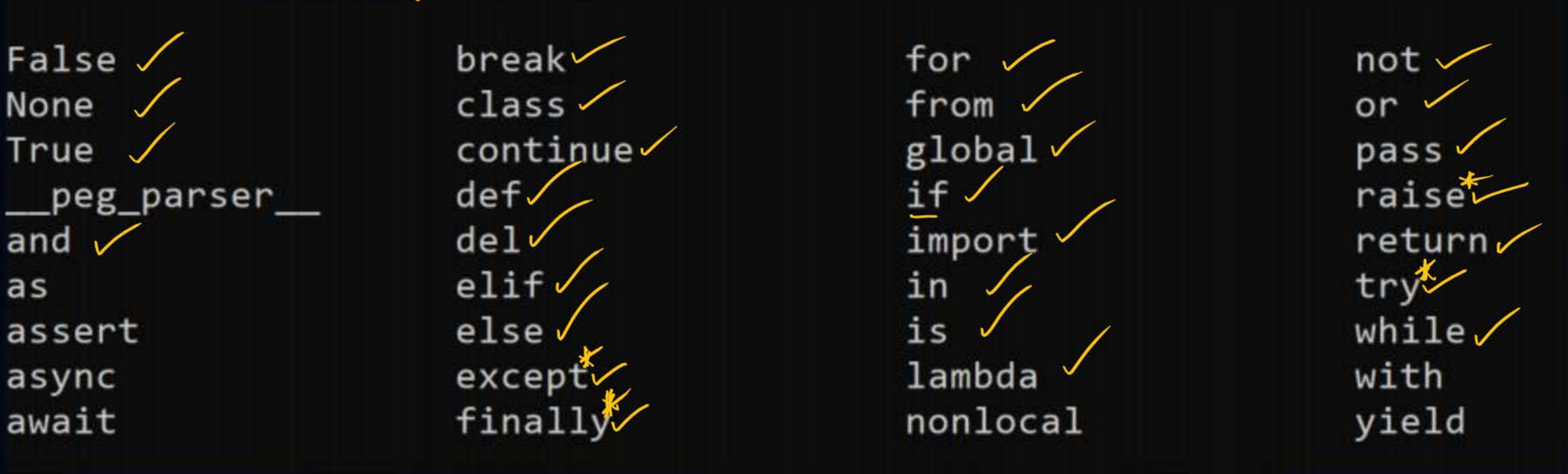
Keywords

⇒ The words, whose meaning | Purpose is Predefined.

⇒ Also known as Reserved words.

⇒ 36 keywords

⇒ All keywords will be Expressed in lower-Case [Except ✓True, ✓False, ✓None]



False ✓
None ✓
True ✓
__peg_parser__
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

Topic : Operators in Python - 1

Operators in Python

- Operator Performs Operation on operands

Ex: a * b
- a,b operands
- * operator
- multiplication operation

- Operators in Python classified into 7 Categories:

1) Arithmetic operators
2) Logical
3) Bitwise & shift
4) Relational (or) Comparison
5) Assignment
6) Identity
7) Membership

Topic : Operators in Python - 1

Operator Type	operators
Arithmetic	+, −, *, /, //, %, **
Logical	and, or, not
Bitwise and shift	&, |, ^, ~, <<, >>
Relational	<, <=, >, >=, ==, !=
Assignment	=, +=, −=, *=, &=, |= - - -
Identity	is , is not
Membership	in, not in

2 mins Summary

- Print ( ) with format specifiers
- Tokens
- Identifiers
- keywords
- Operators*

To be Contd . . .

THANK - YOU