

PYTHON INTRODUCTION

Why should you learn python?

1. Python is rapidly growing and has been used in many fields like Data Analysis, Machine Learning, Deep Learning, AI, Desktop Application Development and Web Development.

Python 2 Vs Python 3:

1. In 2008, 'Guido van Rossum' author of python has decided to full fill demerits of Python 2 and introduced Python 3.
2. But Python-3 not backward compatible by that time(i.e, we cannot run Python2's code in Python3)
3. This has a huge problem in migration and the modules of python 2 are being re-written in Python-3 again.
4. So Rossum decided to maintain both the versions.
5. By 2020 most of the companies have successfully migrated to Python-3.

Comments in Python:

1. To enhance the readability of the code.
2. Application level logics are easily understandable.
3. These are non-executable code.
4. Helps in maintaining the project.
5. Two types of comments are useful a) single line comment and b)multi line comments
6. Multiple line comments: `'''comments'''` (or)
7. Multiline comments: `""" comments """`
8. Single line comments: `#comments`

Escape sequence characters in Python:

1. `print('He said, "iam good"')`
2. `print("He said, 'iam good'")`

Formatting data using format specifiers(like in C):

1. int → %d or %i
2. float → %f, %g
3. String → %s

Example:

```
eid, ename, sal = 123, "jonathan", 25000
```

```
#formatting using format specifiers
```

```

print("Emp id is %d and name is %s and salary is %f" %(eid,ename,sal))

#formatting using format specifiers
print("Emp id is %d and name is %s and salary is %g" %(eid,ename,sal))

#formatting using braces and format function
print("Emp id is {} and name is {} and salary is {}".format(eid,ename,sal))

#formatting using format specifiers with indexes
print("Emp id is {0} and name is {1} and salary is {2}" .format(eid,ename,sal))

print("Emp id is {eid} and name is {ename} and salary is {sal}"
.format(eid=111,ename='hello',sal=35000))

```

DataTypes:

1. String can be mentioned in single quotes or in Double quotes
 - a. Ex: String str ='hello' or String str="hello"
2. There is no char data type, instead we use String for this type too.
3. int , float, boolean and double are common.
4. The short form representation of String in python is 'str' and boolean is 'bool'.
 - a. True → 1 and False → 0
5. Unlike other programming languages , in python the boolean values are represented as capital 'True' and 'False'.
6. Deleting a variable is possible in python by using 'del' keyword
7. Addition of number + number and string+string is possible but, number+string is not possible.
8. Multiple initialization of variable name is valid but new one will get overridden.

```

Ex:  name = 'hello'
      print(name)
      del name
      print(name)

```

9. Swapping is simply done : x,y=2,3 and then to swap x,y = y,x and swapping is done.

Identifiers in Python:

1. The naming conventions of variables, functions, methods , classes and modules of python are called identifiers.
2. A-Z, a-z, 0-9 are allowed
3. Underscore is allowed as identifier.
4. No identifier should start with a digit.
5. Case sensitive and should not use reserved words as identifiers.

6. No name length limits for python identifiers.
7. If any identifier starts with a single under-score '_' then it is a private identifier.
 - a. `_name`
8. If any identifier starts with a double under-score '__' then it is a strictly private identifier.
 - a. `__name`
9. If any identifier starts and ends with double under-score then it is 'language specific identifier'.
 - a. `__init__`
 - b. `__demo__`

Reserved Words in Python:

1. As of now 33 keywords are present in Python.
 - True, False, None
 - and, or, not, is
 - if, else, elif
 - while, for, break, continue, return, in, yield
 - try, except, finally, raise, assert
 - import, from, as, class, def, pass, global, nonlocal, lambda, del, with

Example: Returns all available keywords in python

```
import keyword
```

```
for k in keyword.kwlist:  
    print(k)
```

Pre-defined DataTypes of Python:-

int

float

complex

bool

str

bytes

bytearray

range

list

tuple

set → *list of values without duplicates*

frozenset → *just like set but not modifiable*

dict → *just like map concept of java*

NOTE: In python datatype 'complex' is available best suitable for scientific computations like $10+2j$.

NOTE: In the above list int, float, complex and bool are non-objects and remaining all are objects.

NOTE: In python most of the things are pre-defined objects even some functions act as objects.