python Introduction

In [ ]:
```
# what is python?
- python is simple,general purpose,interpreted,scripting,highlevel,oop language.
- python is developed by Guido Van Rossum.
- started implementing python in  1989 and released in 1991
```

In [ ]:
```
# why python named as python
- this name came from BBC's comedy televeision show -Monty Python's Flying circus(from 1970s)
```

In [ ]:
```
# python features
- easy to learn and use
- expressive language
- interpreted language(line by line)
- cross platform language/portable language
- free and opensource
- object oriented
- extensible
- large standard library
- gui programing support
```

In [1]:
```
print("python")
gdsjhgsjhfkjhkjkj
jhsghjgvjhvhkjv
```

```
python

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-8f0de7907d8e> in <module>
      1 print("python")
----> 2 gdsjhgsjhfkjhkjkj
      3 jhsghjgvjhvhkjv

NameError: name 'gdsjhgsjhfkjhkjkj' is not defined
```

In [ ]:
```
# applications of python?
- web applications
- ai
- datascience
- gaming
- gui based applications
- multimedia'
- machine learning
- scientific computing
```

In [ ]:
```
# python versions:
- python 1.0,january 1994
- python 2.0,octobar 2000
- python 3.0,december 2008
- python 2.7,july 2010
- python 3.8.3, may,2020
```

In [ ]:
```

```

python basics

In [ ]:
```
# python comments:
- comment is not a part of the program,it makes the program readble
# 2 types of comments:
- single line comments
- multiline comments
```

In [3]:
```
#single line comment
"""
multi
line
comment
"""
print("hello")
```

```
hello
```

In [8]:
```
print('python"s workshop')
```

```
python"s workshop
```

In [ ]:
```
# python basics
- variables
- keywords
- operators
```

In [ ]:
```
# variable:
- variable name=VALUE
- which is used to refer memory location.it used  to hold values.
```

In [9]:
```
n=10
print(n)
```

```
10
```

In [ ]:
```
- smart enough to get variable type.
- it is grup of letters and digits,but it starts with only letters and unserscore.
- keywords cannot be used as variablenames.
```

In [16]:
```
pyth1on=1000
print(pyth1on)
```

```
1000
```

In [ ]:
```
#keyword:
- keywords ar the reserved words in python.
```

In [20]:
```
import keyword
print(keyword.kwlist)
```

```
['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', 'p
ass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

In [18]:
```
False=100
print(False)
```

```
  File "<ipython-input-18-5c4f214c7594>", line 1
    False=100
          ^
SyntaxError: can't assign to keyword
```

In [ ]:
```
#datatype
- int
- float
- string(str)
```

In [24]:
```
a="python"
b=45
c=98.7
print(a)
print(type(a))
print(b)
print(type(b))
print(c)
print(type(c))
```

```
python
<class 'str'>
45
<class 'int'>
98.7
<class 'float'>
```

In [25]:
```
a="python"
b=45
c=98.7
print(a,b,c)
```

```
python 45 98.7
```

In [26]:
```
a=10,b=78.90,c="python"
print(a,b,c)
```

```
  File "<ipython-input-26-8d755ef92774>", line 1
    a=10,b=78.90,c="python"
     ^
SyntaxError: can't assign to literal
```

In [27]:
```
a,b,c=10,78.90,"python"
print(a,b,c)
```

```
10 78.9 python
```

In [29]:
```
a=b=c=100
print(a,b,c)
```

```
100 100 100
```

In [30]:
```
a=a+1
print(a)
```

```
101
```

In [33]:
```
n="67"
print(type(n))
n=n+"1"
print(n)
```

```
<class 'str'>
671
```

In [34]:
```
n=67
print(type(n))
n=n+1
print(n)
```

```
<class 'int'>
68
```

In [ ]:
```
# Type conversions
integer -int(variablename)
string  -str(variablename)
float   -float(variablename)
```

In [36]:
```
#str to int
n="67"
print(type(n))
a=int(n)
print(a)
print(type(a))
```

```
<class 'str'>
67
<class 'int'>
```

In [38]:
```
#float to str
f=90.67
print(type(f))
a=str(f)
print(a)
print(type(a))
```

```
<class 'float'>
90.67
<class 'str'>
```

In [39]:
```
n="10"
print(n+str(10))
```

```
1010
```

In [40]:
```
n="10"
print(int(n)+10)
```

```
20
```

In [ ]:
```
#userinputs
str---->input("enter")
int---->int(input("enter"))
float---->float(input("enter"))
```

In [50]:
```
#string userinput
s=input("enter a string")
print(s)
print(type(s))
```

```
enter a string78
78
<class 'str'>
```

In [48]:
```
#int userinput
s=int(input("enter a number"))
print(s)
print(type(s))
```

```
enter a number23
23
<class 'int'>
```

In [51]:
```
#float userinput
s=float(input("enter a number"))
print(s)
print(type(s))
```

```
enter a number90
90.0
<class 'float'>
```

In [54]:
```
s=input("enter a name")
print("hello",s)
print("hello"+s)
print("hello "+s)
```

```
enter a namepavani
hello pavani
hellopavani
hello pavani
```

In [ ]:
```
welcome to clgname urname
ex: welcome to aditya pavani
```

In [55]:
```
a=input("enter clg name")
b=input("enter your name")
print("welcome to",a,b)
```

```
enter clg nameaditya
enter your namepavani
welcome to aditya pavani
```

In [ ]:
```
# python operators
1.arithmatic operators(+,-,*,%,/,//(floor),**(power))
2.assignment operators(=,+=,-=,/=,*=,%=,//=,**=)
3.logical operators(and,or,not)
4.comparision operators(==,!=,<,>,<=,>=)
5.membership operators(in,not in)
6.identity operators(is,is not)
7.bitwise oprators(&,|,^,<<,>>)
```

In [64]:
```
#arithmatic operators
a=110
b=70
print(a+b)
print(a/b)
print(a//b)#floor
print(a*b)
print(a**b)#power
```

```
180
1.5714285714285714
1
7700
789746956799439217432898878450480984754072988193502405966258189471033220100000000000000000000000000000000000000000000000
00000000000000000000000
```

In [67]:
```
#assignment operators
a=10
b=20
a+=b#a=a+b
print(a)
a-=b#a=a-b
print(a)
```

```
30
10
```

In [70]:
```
#Logical operators
a=7
print(a<5 and a<10)
print(a<5 or a<10)
print(not(a<5 or a<10))
```

```
False
True
False
```

In [74]:
```
#comparision operators
a=10
b=22
print(a==b)
print(a<b)
print(a>b)
print(a!=b)
```

```
False
True
False
True
```

In [76]:
```
#membership operators
a="python"
print('p' in a)
print("p" not in a)
```

```
True
False
```

In [79]:
```
#identity operators
a="python"
b="pjhagjfgjh"
print(a is b)
print(a is not b)
```

```
False
True
```

In [84]:
```
#bitwise operators   #128 64 32 16 8 4 2 1
a=2 #0000 0010
b=3 #0000 0011
print(a&b) #0010
print(a|b) #0011
print(a^b) #0001
print(a<<b) #0001 0000
print(a>>b) #0000 0000
```

```
2
3
1
16
0
```

In [ ]:
```
0010
0011

_____
0010--2
0011--3
0001--1
```