In [1]:
```
#python is a programming language
#This is platform independent
#This is a highlevel / procedural as well as oops language
# Guido von rossum made it , took inspiration from a circus
```

In [3]:
```
#This is rich of libraries, launched in 1991.
```

In [4]:
```
#These are suitable for standalone , data science , web application , Artifici
```

In [5]:
```
#This is open source language
```

In [6]:
```
#python has its own IDE as the name of Python IDLE
#This will support other ide such as pycharm , vs code etc.
```

In [7]:
```
#Pycharm is used in the world wide.
```

In [8]:
```
#python is the interpreter based, this is a fast interpretor
```

In [9]:
```
#Python has the component PVM that is like the jvm (Python virtual machine)
```

In [10]:
```
#java : compile then interpret, Python is interpret and then interpret.
```

In [11]:
```
#The source code is high level language , it will enteract with its library.
```

In [12]:
```
#here is the package / module where usually its called module , java calls it
```

In [ ]:
```
#Tensor flow , Pytorch are the modules of python used in AI
```

In [2]:
```
a=4
print(a)
```

```
4
```

In [ ]:
```
 #int x = 20; is example of the explicit declaration
#python will support the implicit declaration like
'''>>> x=10
>>> type(x)
<class 'int'>
>>> y=10.22
>>> type(y)
<class 'float'>
>>> x='Ravi' #or "Ravi" or Ravi(triple quotes) gives you type string
>>> type(x)
<class 'str'>'''

# is single line comment and then there the ''' used for the multiline comment
```

In [ ]:
```
x=True
type(x)
type boolean
```

In [ ]:

```
'''most of the programming language , there is the same memory for the value 1
x=10 and y=10 . Python treats the x and y the same
>>> id(x)
140711885649096
>>> id(y)
140711885649096

if the value is changed then the id will be also be changed.
>>> id(x)
140711885649128
>>> id(y)
140711885649096'''
```

In [ ]:

```
Mathematical operators:
>>> a=10
>>> b=5
>>> a+b
15
>>> a-b
5
>>> a*b
50
>>> a/b
2.0
>>> b=3
>>> a/b
3.3333333333333335
>>> a//b
3
>>> a**b
1000
```

In [ ]:

```
logical operators

>>> a>b
True
>>> a<b
False
>>> a>=b
True
>>> a<=b
False
>>> 2>3 and 3>2
False
>>> 2>3 or 3>2
True

There is no concept of the increment and the decrement operators , there is th
eg.

'''
>>> x=10
>>> x+=1
>>> x
11
'''

There is no direct use of the ternary condition
like (condition )? True : False
```