basic Python (variable,data types,operation)

In [6]:
```
a = 9
b = 10
c = 20
```

In [7]:
```
print(a)
print(b)
print(c)
```

```
9
10
20
```

In [8]:
```
type(a,b,c)    #pass only one argument
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[8], line 1
----> 1 type(a,b,c)

TypeError: type.__new__() argument 1 must be str, not int
```

In [9]:
```
type(a)
```

Out[9]:
```
int
```

In [10]:
```
d = 13.9
e = 'nit'
print(d,e)
```

```
13.9 nit
```

In [11]:
```
str(d)       #convert int to string
```

Out[11]:
```
'13.9'
```

In [12]:
```
str('nit')
```

Out[12]:
```
'nit'
```

In [13]:
```
int('nit')      #sholud not convert string to int
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[13], line 1
----> 1 int('nit')

ValueError: invalid literal for int() with base 10: 'nit'
```

In [14]:
```
int(e)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[14], line 1
----> 1 int(e)

ValueError: invalid literal for int() with base 10: 'nit'
```

In [18]:
```
a = 9
b = 10
c = 20
a
b
c
```

Out[18]: 20

In [20]:
```
f1 = a + b
f2 = c + d
f1
```

Out[20]: 19

In [22]:
```
f2
```

Out[22]: 33.9

In [23]:
```
f3 = a - b
f3
```

Out[23]: -1

In [24]:
```
f4 = d - e      # can not do subtraction with string and float
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[24], line 1
----> 1 f4 = d - e

TypeError: unsupported operand type(s) for -: 'float' and 'str'
```

In [26]:
```
f5 = a + e
f5
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[26], line 1
----> 1 f5 = a + e
      2 f5

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

In [27]:
```
f4 = a /b
f4
```

Out[27]: 0.9

In [29]:
```
f4 = a // b           #// gives exact value means withoout decimals
f4
```

Out[29]: 0

In [30]:
```
f5 = a*b
f5
```

Out[30]: 90

In [31]:
```
f6 = a**3        #** is used to find squares,cubes etc
f6
```

Out[31]: 729

In [32]:
```
f7 = c % b     # % gives remainder value
f7
```

Out[32]: 0

In [35]:
```
5 *  ' hi'
```

Out[35]: ' hi hi hi hi hi'

In [36]:
```
help()
```

```
Welcome to Python 3.12's help utility! If this is your first time using
Python, you should definitely check out the tutorial at
https://docs.python.org/3.12/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To get a list of available
modules, keywords, symbols, or topics, enter "modules", "keywords",
"symbols", or "topics".

Each module also comes with a one-line summary of what it does; to list
the modules whose name or summary contain a given string such as "spam",
enter "modules spam".

To quit this help utility and return to the interpreter,
enter "q" or "quit".

No Python documentation found for 'integers'.
Use help() to get the interactive help utility.
Use help(str) for help on the str class.


You are now leaving help and returning to the Python interpreter.
If you want to ask for help on a particular object directly from the
interpreter, you can type "help(object)".  Executing "help('string')"
has the same effect as typing a particular string at the help> prompt.
```

In [37]:
```
range(10)
```

Out[37]: range(0, 10)

In [40]:
```
i = list[range(10,30)]
i
```

Out[40]: list[range(10, 30)]

In [43]:
```
i = list[range(10,20)]
```

In [44]:
```
for i in range(10,20):
    print(i)
```

```
10
11
12
13
14
15
16
17
18
19
```

In [45]:
```
for i in list[range(0,10)]:

    print(i)
```

```
*list[range(0, 10)]
```

In [46]:
```
range(10,20,5)
```

Out[46]:
```
range(10, 20, 5)
```

In [48]:
```
for i in range(10,50,5):
    print(i)
```

```
10
15
20
25
30
35
40
45
```

In [ ]:
```

```