In [3]:

```
1 print(1)
```

```
1
```

In [4]:

```
1 print(1.2)
```

```
1.2
```

In [5]:

```
1 print(1+1.2)
```

```
2.2
```

In [6]:

```
1 1+3
```

Out[6]:

```
4
```

In [7]:

```
1 print("Hello World!")
```

```
Hello World!
```

In [8]:

```
1 print("1+3")
```

```
1+3
```

In [9]:

```
1 True
```

Out[9]:

```
True
```

In [10]:

```
1 False
```

Out[10]:

```
False
```

In [11]:

```
1 x = 5
```

In [12]:

```
print(x)
```

```
5
```

In [13]:

```
x
```

Out[13]:

```
5
```

In [14]:

```
type(x)
```

Out[14]:

```
int
```

In [16]:

```
y = 1.5
type(y)
```

Out[16]:

```
float
```

In [18]:

```
print(type(y)) # after oops
```

```
<class 'float'>
```

In [19]:

```
s = "hello"
```

In [20]:

```
type(s)
```

Out[20]:

```
str
```

In [21]:

```
x
```

Out[21]:

```
5
```

In [22]:

```
x = 1.5
```

In [23]:

```
1 x
```

Out[23]:

```
1.5
```

In [24]:

```
1 type(x)
```

Out[24]:

```
float
```

In [25]:

```
1 8 / 3
```

Out[25]:

```
2.6666666666666665
```

In [26]:

```
1 - 8 / 3
```

Out[26]:

```
-2.6666666666666665
```

In [27]:

```
1 8 // 3
```

Out[27]:

```
2
```

In [ ]:

```
1 - 8 // 3
```

In [28]:

```
1 - 8 // 5
```

Out[28]:

```
-2
```

In [29]:

```
1 - 3 // 2
```

Out[29]:

```
-2
```

In [30]:

```
5 % 2
```

Out[30]:

```
1
```

In [31]:

```
2 ** 3
```

Out[31]:

```
8
```

In [40]:

```
age = 17

if age >= 18:
    print("Eligible")
elif age >= 16 and age < 18:
    if age == 17:
        print("Mohammed Emran loves you")
else:
    print("Not Eligible")
```

```
Mohammed Emran loves you
```

In [ ]:

```
# age = input()

# if age >= 18
# {

# }
```

In [49]:

```
age = 23
if age >= 18:
    print(age)
```

```
23
```

In [50]:

```
signal = "green"
if signal == "red":
    print("stop")
else:
    print("go")
```

```
go
```

In [51]:

```
signal = "green"
print("stop") if signal == "red" else print("go")
```

```
go
```

In [53]:

```
# loops - for, while
```

In [55]:

```
list(range(1, 5))
```

Out[55]:

```
[1, 2, 3, 4]
```

In [56]:

```
for num in range(1, 5):
    print(num)
```

```
1
2
3
4
```

In [60]:

```
range(1, 5) # generator - to be discussed later
```

Out[60]:

```
range(1, 5)
```

In [62]:

```
# print first 5 even numbers
```

In [65]:

```
cnt = 0
for num in range(1, 1000):
    if num % 2 == 0:
        cnt += 1
        if cnt <= 5:
            print(num)
```

```
2
4
6
8
10
```

In [67]:

```
cnt = 0
for num in range(1, 1000):
    if num % 2 == 0:
        cnt += 1
        if cnt <= 5:
            print(num)
        else:
            break
```

```
2
4
6
8
10
```

In [70]:

```
for num in range(2, 11, 2):
    print(num)
```

```
2
4
6
8
10
```

In [71]:

```
list(range(5))
```

Out[71]:

```
[0, 1, 2, 3, 4]
```

In [72]:

```
# x - [0, x)
# x, y --> [x, y)
# x, y, z --> [x, y), +z
```

In [74]:

```
for num in range(10, 0, -1):
    print(num)
```

```
10
9
8
7
6
5
4
3
2
1
```

In [75]:

```
1 cmd = input("Enter command:")
```

```
Enter command:ls
```

In [76]:

```
1 cmd
```

Out[76]:

```
'ls'
```

In [77]:

```
1 cmd = input("Enter command:")
```

```
Enter command:123
```

In [78]:

```
1 cmd
```

Out[78]:

```
'123'
```

In [79]:

```
1 int('123')
```

Out[79]:

```
123
```

In [80]:

```
1 int(input("Enter command:"))
```

```
Enter command:123
```

Out[80]:

```
123
```

In [81]:

```
1 float('123')
```

Out[81]:

```
123.0
```

In [83]:

```
cmd = input("Enter command:")
while cmd != "exit":
    cmd = input("Enter command:")
    print(cmd)
```

```
Enter command:ls
Enter command:something
something
Enter command:something else
something else
Enter command:else else
else else
Enter command:exit
exit
```

In [84]:

```
def is_even(num):
    return num % 2 == 0
```

In [85]:

```
# lists
```

In [86]:

```
marks = [90, 100, 30, 85, 60, "absent"]
```

In [87]:

```
type(marks)
```

Out[87]:

```
list
```

In [88]:

```
[90, 100, 30, 85, 60, 1.5]
```

Out[88]:

```
[90, 100, 30, 85, 60, 1.5]
```

In [ ]:

```
[90, 100, 30, 85, 60, [1.5]]
```

In [89]:

```
# indexing
```

In [91]:

```
1 marks[1]
```

Out[91]:

```
100
```

In [92]:

```
1 marks[3]
```

Out[92]:

```
85
```

In [93]:

```
1 marks[-3]
```

Out[93]:

```
85
```

In [94]:

```
1 # slicing
```

In [95]:

```
1 marks[1:4]
```

Out[95]:

```
[100, 30, 85]
```

In [96]:

```
1 marks[:4]
```

Out[96]:

```
[90, 100, 30, 85]
```

In [97]:

```
1 marks[::2]
```

Out[97]:

```
[90, 30, 60]
```

In [98]:

```
1 marks[:]
```

Out[98]:

```
[90, 100, 30, 85, 60, 'absent']
```

In [100]:

```
1 marks[::-1]
```

Out[100]:

```
['absent', 60, 85, 30, 100, 90]
```

In [101]:

```
1 len(marks)
```

Out[101]:

```
6
```

In [102]:

```
1 max(marks)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call
 last)
<ipython-input-102-e9b62b81b60b> in <module>
----> 1 max(marks)

TypeError: '>' not supported between instances of 'str' and 'int'
```

In [103]:

```
1 marks = [90, 100, 30, 85, 60]
```

In [104]:

```
1 max(marks)
```

Out[104]:

```
100
```

In [105]:

```
1 sum(marks)
```

Out[105]:

```
365
```

In [106]:

```
sum(['absent', 60, 85, 30, 100, 90])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call
 last)
<ipython-input-106-eadb1e30b23a> in <module>
----> 1 sum(['absent', 60, 85, 30, 100, 90])

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

In [107]:

```
[1, 2, 3] + [4, 5, 6] # concatenate
```

Out[107]:

```
[1, 2, 3, 4, 5, 6]
```

In [108]:

```
[1, 2, 3] * 3
```

Out[108]:

```
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

In [109]:

```
[1, 2, 3] * [4, 5, 6]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call
 last)
<ipython-input-109-fde107a94d11> in <module>
----> 1 [1, 2, 3] * [4, 5, 6]

TypeError: can't multiply sequence by non-int of type 'list'
```

In [110]:

```
for element in marks:
    print(element)
```

```
90
100
30
85
60
```

In [112]:

```
for idx in range(0, len(marks)):
    print(idx, marks[idx])
```

```
0 90
1 100
2 30
3 85
4 60
```

In [167]:

```
for element in enumerate(marks):
    print(element)
```

```
(0, 1)
(1, 2)
(2, 3)
(3, 4)
```

In [122]:

```
a = [10, 20, 30]
b = a
c = [10, 20, 30]
d = list(a) # list() creates a sep copy
e = a[:] # slicing also creates a sep copy
f = a.copy()
```

In [145]:

```
print(a == a)
print(a == b)
print(a == c)
print(a == d)
print(a == e)
print(a == f)
```

```
True
True
True
True
True
True
```

In [146]:

```
a[0] = 100
```

In [147]:

```
a
```

Out[147]:

```
[100, 20, 30]
```

In [148]:

```
print(a == a)
print(a == b)
print(a == c)
print(a == d)
print(a == e)
print(a == f)
```

```
True
True
False
False
False
False
```

In [127]:

```
id(a)
```

Out[127]:

```
140334840161088
```

In [128]:

```
id(b)
```

Out[128]:

```
140334840161088
```

In [129]:

```
id(c)
```

Out[129]:

```
140334840198400
```

In [130]:

```
id(d)
```

Out[130]:

```
140334048892800
```

In [131]:

```
id(e)
```

Out[131]:

```
140334840090432
```

In [137]:

```
a = [10, 20, 30]
b = a
c = [10, 20, 30]
d = list(a) # list() creates a sep copy
e = a[:] # slicing also creates a sep copy
```

In [138]:

```
print(a == a)
print(a == b)
print(a == c)
print(a == d)
print(a == e)
```

```
True
True
True
True
True
```

In [139]:

```
print(a is a)
```

```
True
```

In [140]:

```
print(a is b)
```

```
True
```

In [141]:

```
print(a is c)
```

```
False
```

In [142]:

```
print(a is d)
```

```
False
```

In [143]:

```
print(a is e)
```

```
False
```

In [149]:

```
# nested lists
```

In [150]:

```
mat = [[1, 2], [3, 4]]
```

In [151]:

```
1 mat
```

Out[151]:

```
[[1, 2], [3, 4]]
```

In [ ]:

```
1 mat[0]
```

In [ ]:

```
1 # 3
```

In [153]:

```
1 mat[1][0]
```

Out[153]:

```
3
```

In [154]:

```
1 mat * 2
```

Out[154]:

```
[[1, 2], [3, 4], [1, 2], [3, 4]]
```

In [156]:

```
1 marks = [1, 2, 3, 4]
```

In [ ]:

```
1 # [1, 2, 3, 4] * 2
```

In [155]:

```
1 [1, 2, 3, 4] * 2
```

Out[155]:

```
[1, 2, 3, 4, 1, 2, 3, 4]
```

In [159]:

```
1 result = []
2 for num in marks:
3     result.append(num * 2)
```

In [160]:

```
result
```

Out[160]:

```
[2, 4, 6, 8]
```

In [161]:

```
[num*2 for num in marks] # list comprehension
```

Out[161]:

```
[2, 4, 6, 8]
```

In [162]:

```
# first 10 sqaures - 1, 4, 9, 16, 25...
[num**2 for num in marks]
```

Out[162]:

```
[1, 4, 9, 16]
```

In [163]:

```
# take the power of 3, if its even or odd, same number, +1
```

In [165]:

```
def fn(num):
    num_2 = num**3
    if num_2 % 2 == 0:
        return num_2
    else:
        return num_2 + 1
```

In [166]:

```
[fn(num) for num in marks]
```

Out[166]:

```
[2, 8, 28, 64]
```

In [173]:

```
-8 % 3
```

Out[173]:

```
1
```

In [169]:

```
type(print(5))
```

```
5
```

Out[169]:

```
NoneType
```

In [170]:

```
x = print(5)
```

```
5
```

In [172]:

```
print(x)
```

```
None
```

In [174]:

```
3.3 - 3.2
```

Out[174]:

```
0.09999999999999964
```

In [175]:

```
-8 % 3
```

Out[175]:

```
1
```

In [176]:

```
-8 == -3*3 + 1
```

Out[176]:

```
True
```

In [177]:

```
-8 == -2*3 - 2
```

Out[177]:

```
True
```

In [1]:

```
marks = [1, 2, 3, 4]
marks.append(10)
```

In [2]:

```
1 marks
```

Out[2]:

```
[1, 2, 3, 4, 10]
```

In [3]:

```
1 marks.pop()
```

Out[3]:

```
10
```

In [4]:

```
1 marks
```

Out[4]:

```
[1, 2, 3, 4]
```

In [5]:

```
1 marks.insert(0, 100)
```

In [6]:

```
1 marks
```

Out[6]:

```
[100, 1, 2, 3, 4]
```

In [7]:

```
1 marks
```

Out[7]:

```
[100, 1, 2, 3, 4]
```

In [8]:

```
1 marks.remove(2)
```

In [9]:

```
1 marks
```

Out[9]:

```
[100, 1, 3, 4]
```

In [11]:

```
marks.pop(0)
```

Out[11]:

```
100
```

In [12]:

```
marks
```

Out[12]:

```
[1, 3, 4]
```

In [13]:

```
### Strings
```

In [14]:

```
s = "Alexa, Switch off the lights."
```

In [15]:

```
print(s)
```

```
Alexa, Switch off the lights.
```

In [16]:

```
s[7:17]
```

Out[16]:

```
'Switch off'
```

In [17]:

```
s[7:17:2]
```

Out[17]:

```
'Sic f'
```

In [18]:

```
s[:5]
```

Out[18]:

```
'Alexa'
```

In [19]:

```
s[-2]
```

Out[19]:

```
's'
```

In [20]:

```
s[::-1]
```

Out[20]:

```
'.sthgil eht ffo hctiwS ,axelA'
```

In [21]:

```
s = "Hello World"
```

In [22]:

```
id(s)
```

Out[22]:

```
140454020862448
```

In [23]:

```
s[0] = "Y"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call
 last)
<ipython-input-23-191ee95fd5d4> in <module>
----> 1 s[0] = "Y"

TypeError: 'str' object does not support item assignment
```

In [24]:

```
l = [1, 2, 3, 4]
l[0] = "Y"
```

In [25]:

```
# Immutability
# Strings are immutable - all basic data types mutable
# Lists are mutable
```

In [26]:

```
a = 1
id(a)
```

Out[26]:

```
4365732192
```

In [27]:

```
a = 2
id(a)
```

Out[27]:

```
4365732224
```

In [28]:

```
l = [1, 2, 3, 4]
id(l)
```

Out[28]:

```
140455099351616
```

In [29]:

```
l[0] = "Y"
id(l)
```

Out[29]:

```
140455099351616
```

In [30]:

```
a = "Hello World!"
b = a
a = "Yello World!"
```

In [31]:

```
print(b)
```

```
Hello World!
```

In [32]:

```
a is b
```

Out[32]:

```
False
```

In [33]:

```
a = "hello"
b = a
c = str(a) #list()
d = a[:]
e = "hello"
f = "hel" + "" + "lo"
print(a, b, c, d, e, f)
```

```
hello hello hello hello hello hello
```

In [34]:

```
print(a==b, a==c, a==d, a==e, a==f)
```

```
True True True True True
```

In [35]:

```
print(a is a)
print(b is a)
print(c is a)
print(d is a)
print(e is a)
print(f is a)
```

```
True
True
True
True
True
True
```

In [36]:

```
len(s)
```

Out[36]:

```
11
```

In [37]:

```
s = "Hello World"
s.index("e")
```

Out[37]:

```
1
```

In [38]:

```
s[1]
```

Out[38]:

```
'e'
```

In [39]:

```
s.index("o")
```

Out[39]:

```
4
```

In [40]:

```
s.index("o", 5)
```

Out[40]:

```
7
```

In [41]:

```
1 s.index("Wor")
```

Out[41]:

```
6
```

In [42]:

```
1 s.upper()
```

Out[42]:

```
'HELLO WORLD'
```

In [43]:

```
1 s.lower()
```

Out[43]:

```
'hello world'
```

In [45]:

```
1 "anant mittal".title()
```

Out[45]:

```
'Anant Mittal'
```

In [46]:

```
1 s.startswith("hello")
```

Out[46]:

```
False
```

In [47]:

```
1 s.startswith("Hello")
```

Out[47]:

```
True
```

In [51]:

```
1 web = input()
```

```
HttPs://www.scaler.com (HttPs://www.scaler.com)
```

In [49]:

```
1 web
```

Out[49]:

```
'https://www.scaler.com'
```

In [50]:

```
# http(s), HTTP(S), HTtp(s), hTTp(s)
```

In [52]:

```
web.lower().startswith("http")
```

Out[52]:

```
True
```

In [53]:

```
s="Hello sebastian"
s.index("e",5)
```

Out[53]:

```
7
```

In [59]:

```
email = '''Hi,
My name is Anant'''
```

In [61]:

```
print(email)
```

```
Hi,
My name is Anant
```

In [62]:

```
length = 5
breadth = 3
```

In [65]:

```
"The area of the rectangle with length " + str(length) + " and " + str(breadth)
```

Out[65]:

```
'The area of the rectangle with length 5 and 3 is 15'
```

In [67]:

```
f"The area of the rectangle with length {length} and {breadth} is {length*breadt
```

Out[67]:

```
'The area of the rectangle with length 5 and 3 is 15'
```

In [68]:

```
### Dictionary, Sets, Tuples
```

In [69]:

```
### Tuples
```

In [71]:

```
t1 = (1, 2, 3, 4)
```

In [72]:

```
type(t1)
```

Out[72]:

```
tuple
```

In [73]:

```
t1[0]
```

Out[73]:

```
1
```

In [75]:

```
t1[-1]
```

Out[75]:

```
4
```

In [77]:

```
# iteratable
for element in t1:
    print(element)
```

```
1
2
3
4
```

In [78]:

```
t1[0] = 100
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call
 last)
<ipython-input-78-3c8e12ad4afd> in <module>
----> 1 t1[0] = 100

TypeError: 'tuple' object does not support item assignment
```

In [79]:

```
(1, 2) + (3, 4)
```

Out[79]:

```
(1, 2, 3, 4)
```

In [80]:

```
(1, 2) * 3
```

Out[80]:

```
(1, 2, 1, 2, 1, 2)
```

In [81]:

```
list()
```

Out[81]:

```
[]
```

In [82]:

```
[]
```

Out[82]:

```
[]
```

In [83]:

```
""
```

Out[83]:

```
''
```

In [84]:

```
tuple()
```

Out[84]:

```
()
```

In [94]:

```
()
```

Out[94]:

```
()
```

In [96]:

```
type([1])
```

Out[96]:

```
list
```

In [97]:

```
type("a")
```

Out[97]:

```
str
```

In [98]:

```
1 type((100))
```

Out[98]:

```
int
```

In [101]:

```
1 type((100,))
```

Out[101]:

```
tuple
```

In [102]:

```
1 (100)
```

Out[102]:

```
100
```

In [104]:

```
1 # packing and unpacking
```

In [105]:

```
1 t1 = (100, 200, 300)
```

In [106]:

```
1 t1
```

Out[106]:

```
(100, 200, 300)
```

In [107]:

```
1 t1, t2, t3 = (100, 200, 300)
```

In [108]:

```
1 t1
```

Out[108]:

```
100
```

In [109]:

```
1 t2
```

Out[109]:

```
200
```

In [110]:

```
t3
```

Out[110]:

```
300
```

In [111]:

```
def foo(num):
    return num+1, num+2, num+3
```

In [112]:

```
foo(4)
```

Out[112]:

```
(5, 6, 7)
```

In [114]:

```
for idx, value in enumerate([4, 5, 6, 7]):
    print(idx, value)
```

```
0 4
1 5
2 6
3 7
```

In [ ]:

```
#Tuples are immutable
```

In [ ]:

```
#Tuples are more memory effiecient than Lists
```

In [115]:

```
#Strings are used for packing and unpacking
```

In [116]:

```
# Dictionary
```

In [117]:

```
d = {"Anne": 123456,
    "Bill": 23456,
    "Cathy": 34567}
```

In [118]:

```
d
```

Out[118]:

```
{'Anne': 123456, 'Bill': 23456, 'Cathy': 34567}
```

In [119]:

```
d["Anne"]
```

Out[119]:

```
123456
```

In [120]:

```
d["Bill"]
```

Out[120]:

```
23456
```

In [121]:

```
# insert
d["Don"] = 57588
```

In [122]:

```
d
```

Out[122]:

```
{'Anne': 123456, 'Bill': 23456, 'Cathy': 34567, 'Don': 57588}
```

In [ ]:

```
# delete
```

In [123]:

```
d.pop("Anne")
```

Out[123]:

```
123456
```

In [124]:

```
d
```

Out[124]:

```
{'Bill': 23456, 'Cathy': 34567, 'Don': 57588}
```

In [125]:

```
dict(a=1, b=2, c=3)
```

Out[125]:

```
{'a': 1, 'b': 2, 'c': 3}
```

In [126]:

```
dict([("a", 1), ("b", 2), ("c", 3)])
```

Out[126]:

```
{'a': 1, 'b': 2, 'c': 3}
```

In [127]:

```
for element in d:
    print(element)
```

```
Bill
Cathy
Don
```

In [128]:

```
for element in d:
    print(element, d[element])
```

```
Bill 23456
Cathy 34567
Don 57588
```

In [129]:

```
d.keys()
```

Out[129]:

```
dict_keys(['Bill', 'Cathy', 'Don'])
```

In [130]:

```
d.values()
```

Out[130]:

```
dict_values([23456, 34567, 57588])
```

In [131]:

```
d.items()
```

Out[131]:

```
dict_items([('Bill', 23456), ('Cathy', 34567), ('Don', 57588)])
```

In [134]:

```
# Nested Dictionary
EMPLOYEE_DB = {
    'HR'   : {            # In HR Department
        '967'   : 51000, # Employee number '967' has salary of 51,000
        '650'   : 60000
    },
    'TECH' : {
        '516'   : 950000,
        '1001' : 750000,
        '918'   : 800000
    },
    'SALES' : {
        '887' : 45000,
        '490' : 63000
    }
}
```

In [137]:

```
EMPLOYEE_DB["HR"]["650"]
```

Out[137]:

```
60000
```

In [140]:

```
s1 = {1, 2, 3, 3, 4, 5, 4, 6}
```

In [141]:

```
s1
```

Out[141]:

```
{1, 2, 3, 4, 5, 6}
```

In [142]:

```
s1.add(7)
```

In [143]:

```
s1
```

Out[143]:

```
{1, 2, 3, 4, 5, 6, 7}
```

In [144]:

```
s1.add(1)
```

In [145]:

```
1 s1
```

Out[145]:

```
{1, 2, 3, 4, 5, 6, 7}
```

In [146]:

```
1 s1[1]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-146-da05ae654f28> in <module>
----> 1 s1[1]

TypeError: 'set' object is not subscriptable
```

In [147]:

```
1 len(s1)
```

Out[147]:

```
7
```

In [148]:

```
1 sum(s1)
```

Out[148]:

```
28
```

In [149]:

```
1 max(s1)
```

Out[149]:

```
7
```

In [150]:

```
1 for element in s1:
2     print(element)
```

```
1
2
3
4
5
6
7
```

In [151]:

```
sentence = "be the change you wish to see in the world"
```

In [152]:

```
sentence.split(" ")
```

Out[152]:

```
['be', 'the', 'change', 'you', 'wish', 'to', 'see', 'in', 'the', 'worl
d']
```

In [155]:

```
len(set(sentence.split()))
```

Out[155]:

```
9
```

In [158]:

```
s1 = {1, 2, 3}
s2 = {3, 4}
s1.intersection(s2)
```

Out[158]:

```
{3}
```

In [159]:

```
s1.union(s2)
```

Out[159]:

```
{1, 2, 3, 4}
```

In [160]:

```
s1.difference(s2)
```

Out[160]:

```
{1, 2}
```

In [161]:

```
s1.symmetric_difference(s2)
```

Out[161]:

```
{1, 2, 4}
```

In [162]:

```
s = "hello"
a = s
```

In [163]:

```
1 e = "hello"
```

In [164]:

```
1 id(s)
```

Out[164]:

```
140454289235376
```

In [165]:

```
1 id(a)
```

Out[165]:

```
140454289235376
```

In [166]:

```
1 id(e)
```

Out[166]:

```
140454289235376
```

In [167]:

```
1 s = "yello"
```

In [168]:

```
1 s = "hello"
```

In [169]:

```
1 id(s)
```

Out[169]:

```
140454289235376
```