code with numbers

In [9]:
```
1+1 #Addition
```

Out[9]: 2

In [10]:
```
2-1 #Substration
```

Out[10]: 1

In [11]:
```
7*8 #Multiplication
```

Out[11]: 56

In [12]:
```
8/4 #Division
```

Out[12]: 2.0

In [13]:
```
8//4 # Float Division
```

Out[13]: 2

In [15]:
```
43/3 # Float Division
```

Out[15]: 14.333333333333334

In [16]:
```
43/3 # Integer Division
```

Out[16]: 14.333333333333334

In [17]:
```
6+2-4
```

Out[17]: 4

In [18]:
```
6-3+
```

```
  Cell In[18], line 1
    6-3+
        ^
SyntaxError: invalid syntax
```

In [19]:
```
5+5*5
```

Out[19]: 30

In [20]:
```
(5+5)*5
```

Out[20]: 50

In [22]:
```
2*2*2*2*2    #Exponential
```

Out[22]: 32

In [23]:
```
2*5
```

Out[23]: 10

In [24]:
```
2**5
```

Out[24]: 32

In [25]:
```
7**2
```

Out[25]: 49

In [26]:
```
15//3
```

Out[26]: 5

In [29]:
```
15%2 #Modulus
```

Out[29]: 1

In [30]:
```
10%2
```

Out[30]: 0

In [31]:
```
15%%3
```

```
  Cell In[31], line 1
    15%%3
       ^
SyntaxError: invalid syntax
```

In [32]:
```
-10%2
```

Out[32]: 0

In [33]:
```
-10//3
```

Out[33]: -4

In [35]:
```
7+"nit"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[35], line 1
----> 1 7+"nit"

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

In [36]:
```
6+"rainy"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[36], line 1
----> 1 6+"rainy"

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

In [37]:
```
7*"nit"
```

Out[37]:
```
'nitnitnitnitnitnitnit'
```

In [38]:
```
7 * " nit"
```

Out[38]:
```
' nit nit nit nit nit nit nit'
```

In [39]:
```
a,b,c,d,e = 22, 7.7, 'nit', 22+7j, True
print(a)
print(b)
print(c)
print(d)
print(e)
```

```
22
7.7
nit
(22+7j)
True
```

In [40]:
```
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'complex'>
<class 'bool'>
```

In [41]:
```
type(c)
```

Out[41]:
```
str
```

Lets work with the string

Python inbuild function - print & you need to pass the parameter in print() A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result.

In [42]:
```
print("She is fine")
```

```
She is fine
```

In [43]:
```
"Let's go for the class"
```

Out[43]: "Let's go for the class"

In [44]:
```
s = "Let's go for the class"
s
```

Out[44]: "Let's go for the class"

In [45]:
```
a = 22
b = 7
a + b
```

Out[45]: 29

In [46]:
```
c = a+b
c
```

Out[46]: 29

In [47]:
```
a=7
b='hii'
c=a+b
print(c)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[47], line 3
      1 a=7
      2 b='hii'
----> 3 c=a+b
      4 print(c)

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

In [48]:
```
print('Max it's"Technology"')
```

```
  Cell In[48], line 1
    print('Max it's"Technology"')
                                ^
SyntaxError: unterminated string literal (detected at line 1)
```

In [49]:
```
print('Max it\'s"Technology"')
```

```
Max it's"Technology"
```

In [51]:
```
print("Max it','Technology")
```

```
Max it','Technology
```

In [52]:
```
print('Max it','Technology')
```

```
Max it Technology
```

In [53]:
```
'nit'+'nit'
```

Out[53]: 'nitnit'

In [57]:
```
'nit '+'nit'
```

Out[57]: 'nit nit'

In [59]:
```
'nit'' nit'
```

Out[59]: 'nit nit'

In [60]:
```
5*'nit'
```

Out[60]: 'nitnitnitnitnit'

In [61]:
```
5*' nit'
```

Out[61]: ' nit nit nit nit nit'

In [67]:
```
print('c:\nit') #\n----new line # i will explain
```

```
c:
it
```

In [68]:
```
print(r'c:\nit') #raw string #i will explain later
```

```
c:\nit
```

Variable || Identifires || Objects

In [69]:
```
7
```

Out[69]: 7

In [70]:
```
x=7
x
```

Out[70]: 7

In [71]:
```
x + 7
```

Out[71]: 14

In [72]:
```
y=4
y
```

Out[72]: 4

In [73]:
```
x+y
```

Out[73]: 11

In [74]:
```
x=8
x
```

Out[74]: 8

In [75]:
```
x+y
```

Out[75]: 12

In [76]:
```
x+10
```

Out[76]: 18

In [77]:
```
_ + y #_ understand the previous result of the value
```

Out[77]: 22

In [79]:
```
_ + x
```

Out[79]: 30

In [80]:
```
y
```

Out[80]: 4

In [81]:
```
_ + y
```

Out[81]: 8

In [83]:
```
name = 'Angel'
name
```

Out[83]: 'Angel'

In [84]:
```
name + "Technology"
```

Out[84]: 'AngelTechnology'

In [86]:
```
name + " Technology"
```

Out[86]: 'Angel Technology'

In [87]:
```
'a' 'b'
```

Out[87]: 'ab'

In [88]:
```
name 'Technology'
```

```
  Cell In[88], line 1
    name 'Technology'
         ^
SyntaxError: invalid syntax
```

In [89]:
```
name
```

Out[89]: 'Angel'

In [90]:
```
len(name)
```

Out[90]: 5

In [91]:
```
name[0]
```

Out[91]: 'A'

In [92]:
```
name
```

Out[92]: 'Angel'

In [93]:
```
name[4]
```

Out[93]: 'l'

In [94]:
```
name[-1]
```

Out[94]: 'l'

In [95]:
```
name[7]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[95], line 1
----> 1 name[7]

IndexError: string index out of range
```

In [96]:
```
name[-2]
```

Out[96]: 'e'

Slicing

In [97]:
```
name
```

Out[97]: 'Angel'

In [98]:
```
name[0:1] #to print 2 character
```

Out[98]: 'A'

In [99]:
```
name[1:4]
```

Out[99]: 'nge'

In [100…
```
name
```

Out[100… 'Angel'

In [101…
```
name[5:9]
```

Out[101… ''

In [102…
```
name
```

Out[102… 'Angel'

In [107…
```
name1 = "fine" # chane the strin fine to dine
name1
```

Out[107… 'fine'

In [108…
```
name1[0:1]
```

Out[108… 'f'

In [109…
```
name[0:1] = 'd' # i want to change 1st character of fine (f) to d
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[109], line 1
----> 1 name[0:1] = 'd'

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

In [110…
```
name1[0] = 'd' #strings in python are immutable
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[110], line 1
----> 1 name1[0] = 'd'

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

In [111…
```
name1
```

Out[111… 'fine'

In [112…
```
name1[1:]
```

Out[112… 'ine'

In [113…
```
"d" +name1[1:]
```

Out[113… 'dine'

In [114…
```
len(name1)
```

Out[114… 4

List

In [116…
```
l=[]
```

In [117…
```
num = [10,20,30,40]
num
```

Out[117…
```
[10, 20, 30, 40]
```

In [118…
```
num[0]
```

Out[118…
```
10
```

In [119…
```
num[-1]
```

Out[119…
```
40
```

In [120…
```
num[1:]
```

Out[120…
```
[20, 30, 40]
```

In [121…
```
num[:1]
```

Out[121…
```
[10]
```

In [122…
```
num1 = ['hi','hello']
num1
```

Out[122…
```
['hi', 'hello']
```

In [124…
```
num2 = ['hi',8.9,34] # we cn assign multiple variable
num2
```

Out[124…
```
['hi', 8.9, 34]
```

In [125…
```
# can we have 2 list together
num3 = [num,num1]
num3
```

Out[125…
```
[[10, 20, 30, 40], ['hi', 'hello']]
```

In [126…
```
num4 = [num,num1,num2]
num4
```

Out[126…
```
[[10, 20, 30, 40], ['hi', 'hello'], ['hi', 8.9, 34]]
```

In [127…
```
num
```

Out[127…
```
[10, 20, 30, 40]
```

In [142…
```
num.append(50)
num
```

Out[142…
```
[10, 20, 30, 50, 50]
```

In [145…
```
num
```

Out[145… [10, 20, 30, 50, 50]

In [146…
```
num.remove(50)
```

In [147…
```
num
```

Out[147… [10, 20, 30, 50]

In [148…
```
num.pop(1)
```

Out[148… 20

In [149…
```
num
```

Out[149… [10, 30, 50]

In [150…
```
num.pop()
```

Out[150… 50

In [151…
```
num
```

Out[151… [10, 30]

In [152…
```
num1
```

Out[152… ['hi', 'hello']

In [153…
```
num1.insert(2,'nit')
```

In [154…
```
num1
```

Out[154… ['hi', 'hello', 'nit']

In [155…
```
num1.insert(0, 1)
```

In [156…
```
num1
```

Out[156… [1, 'hi', 'hello', 'nit']

In [157…
```
num2
```

Out[157… ['hi', 8.9, 34]

In [158…
```
del num2[2:1]
```

In [165…
```
num2
```

Out[165… ['hi', 8.9, 34, 29, 15, 20, 29, 15, 20, 29, 15, 20]

In [166…
```
num2.extend([29,15,20])
```

In [167…
```
num2
```
Out[167… ['hi', 8.9, 34, 29, 15, 20, 29, 15, 20, 29, 15, 20, 29, 15, 20]

In [168…
```
num2 = ['hi',8.9,34]
num2
```
Out[168… ['hi', 8.9, 34]

In [169…
```
num2.extend([29,15,20])
num2
```
Out[169… ['hi', 8.9, 34, 29, 15, 20]

In [170…
```
num3
```
Out[170… [[10, 30], [1, 'hi', 'hello', 'nit']]

In [174…
```
num3.extend(['a',5,6.7])
```

In [175…
```
num3
```
Out[175… [[10, 30, 'a', 5, 6.7], [1, 'hi', 'hello', 'nit'], 'a', 5, 6.7]

In [176…
```
num1 = ['hi','hello']
num1
```
Out[176… ['hi', 'hello']

In [177…
```
num2 = ['hi',8.9,34]
num2
```
Out[177… ['hi', 8.9, 34]

In [178…
```
num = [10,20,30,40]
num
```
Out[178… [10, 20, 30, 40]

In [179…
```
num3 = [num,num1]
num3
```
Out[179… [[10, 20, 30, 40], ['hi', 'hello']]

In [180…
```
num3.extend(['a',5,6.7])
```

In [181…
```
num3
```
Out[181… [[10, 20, 30, 40], ['hi', 'hello'], 'a', 5, 6.7]

In [182…
```
num
```

Out[182… [10, 20, 30, 40]

In [183…
```
min(num)
```

Out[183… 10

In [184…
```
max(num)
```

Out[184… 40

In [185…
```
num1
```

Out[185… ['hi', 'hello']

In [186…
```
min(num1)
```

Out[186… 'hello'

In [187…
```
min(num3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[187], line 1
----> 1 min(num3)

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

In [191…
```
m = ['ze','za','g']
m
```

Out[191… ['ze', 'za', 'g']

In [192…
```
min(m)
```

Out[192… 'g'

In [193…
```
max(m)
```

Out[193… 'ze'

In [194…
```
sum(num)
```

Out[194… 100

In [195…
```
num.sort()
```

In [196…
```
num
```

Out[196… [10, 20, 30, 40]

In [197…
```
l=[1,2,3]
l
```

Out[197… [1, 2, 3]

In [198…
```
l[0]=100
l
```

Out[198… [100, 2, 3]

Tuple

In [199…
```
tup = (15,25,22,7,30)
tup
```

Out[199… (15, 25, 22, 7, 30)

In [201…
```
tup[0]
```

Out[201… 15

In [205…
```
tup [0] = 10
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[205], line 1
----> 1 tup [0] = 10

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

In [ ]:

Set

In [206…
```
s = {}
```

In [207…
```
s1 = {21,6,34,58,5}
s1
```

Out[207… {5, 6, 21, 34, 58}

In [208…
```
s2 = {50,35,53,'nit',53}
```

In [209…
```
s2
```

Out[209… {35, 50, 53, 'nit'}

In [210…
```
s1[1]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[210], line 1
----> 1 s1[1]

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

In [ ]:

Dictionary

In [211…
```
dict = {1:'apple',2:'banana',4:'orange'}
dict
```

Out[211…
```
{1: 'apple', 2: 'banana', 4: 'orange'}
```

In [212…
```
dict[4]
```

Out[212…
```
'orange'
```

In [213…
```
dict[3]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[213], line 1
----> 1 dict[3]

KeyError: 3
```

In [214…
```
dict.get(2)
```

Out[214…
```
'banana'
```

In [215…
```
dict.get(3)
```

In [216…
```
print(dict.get(3))
```

```
None
```

In [217…
```
dict.get(1,"Not Found")
```

Out[217…
```
'apple'
```

In [218…
```
dict.get(3,"Not Found")
```

Out[218…
```
'Not Found'
```

In [219…
```
dict[5] = "five"
dict
```

Out[219…
```
{1: 'apple', 2: 'banana', 4: 'orange', 5: 'five'}
```

In [220…
```
dict[3] = "mango"
dict
```

Out[220…
```
{1: 'apple', 2: 'banana', 4: 'orange', 5: 'five', 3: 'mango'}
```

In [221…
```
del dict[5]
```

In [222…
```
dict
```

Out[222…
```
{1: 'apple', 2: 'banana', 4: 'orange', 3: 'mango'}
```

In [223…
```
del dict[4]
```

In [224…
```
dict
```

Out[224…
```
{1: 'apple', 2: 'banana', 3: 'mango'}
```

In [225…
```
prog = {'python': ['vscode', 'pycharm'],'machine learning': 'sklearn','datascience'
```

In [226…
```
prog
```

Out[226…
```
{'python': ['vscode', 'pycharm'],
 'machine learning': 'sklearn',
 'datascience': ['jupyter', 'spyder']}
```

In [227…
```
prog["python"]
```

Out[227…
```
['vscode', 'pycharm']
```

In [228…
```
prog['machine learning']
```

Out[228…
```
'sklearn'
```

In [229…
```
prog['datascience']
```

Out[229…
```
['jupyter', 'spyder']
```

Introduce to ID()

In [231…
```
num = 5
id(num)
```

Out[231…
```
140714803210808
```

In [232…
```
name = 'nit'
id(name)
```

Out[232…
```
2681224288880
```

In [233…
```
a = 10
id(a)
```

Out[233…
```
140714803210968
```

In [234…
```
b = a
```

In [235…
```
id(b)
```

Out[235… 140714803210968

In [236…
```
id(10)
```

Out[236… 140714803210968

In [237…
```
k=10
id(k)
```

Out[237… 140714803210968

In [238…
```
a = 20
id(a)
```

Out[238… 140714803211288

In [239…
```
id(b)
```

Out[239… 140714803210968

what ever the variale we assigned the memory and we not assigned anywhere then we can use as garbage collection.|| VARIABLE - we can change the values || CONSTANT - we cannot change the value -can we make VARIABLE as a CONSTANT (note - in python you cannot make variable as constant)

In [240…
```
pi = 3.14    #in math this is alway constant but python we can change
pi
```

Out[240… 3.14

In [241…
```
pi = 3.15
pi
```

Out[241… 3.15

In [242…
```
type(pi)
```

Out[242… float

DATA TYPES & DATA STRUCTURES--

1- NUMERIC || 2-LIST || 3-TUPLE || 4-SET || 5-STRING || 6-RANGE || 7-DICTIONARY

In [243…
```
x = 7.7
type(x)
```

Out[243… float

In [244…
```
(a)
```

Out[244… 20

In [245…
```
x2 = 22+7j
type(x2)
```

Out[245… complex

In [246…
```
a = 7.22
b = int(a)
```

In [247…
```
b
```

Out[247… 7

In [248…
```
type(a)
```

Out[248… float

In [249…
```
k = float(b)
```

In [250…
```
k
```

Out[250… 7.0

In [251…
```
print(a)
print(b)
print(k)
```

```
7.22
7
7.0
```

In [252…
```
k1 = complex(b,k)
```

In [253…
```
print(k1)
type(k1)
```

```
(7+7j)
```

Out[253… complex

In [254…
```
b<k
```

Out[254… False

In [256…
```
condition = b<k
condition
```

Out[256… False

In [257…
```
type(condition)
```

Out[257… bool

In [258…
```
int(True)
```
Out[258… 1

In [259…
```
int(False)
```
Out[259… 0

In [260…
```
l = [1,2,3,4]
print(l)
type(l)
```
[1, 2, 3, 4]

Out[260… list

In [261…
```
s = {1,2,3,4}
s
```
Out[261… {1, 2, 3, 4}

In [262…
```
type(s)
```
Out[262… set

In [263…
```
s1 = {1,2,3,4,4,3,11} #duplicates are not allowed
s1
```
Out[263… {1, 2, 3, 4, 11}

In [264…
```
t = (10,20,30)
t
```
Out[264… (10, 20, 30)

In [265…
```
type(t)
```
Out[265… tuple

In [266…
```
str = 'nit'
type(str)
```
Out[266… str

In [267…
```
st = 'n'
type(st)
```
Out[267… str

Range

In [268…
```
r = range(0,7)
r
```

Out[268… range(0, 7)

In [269…
```
type(r)
```

Out[269… range

In [270…
```
# if you want to print the range
list(range(7,22))
```

Out[270… [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]

In [271…
```
r1 = list(r)
r1
```

Out[271… [0, 1, 2, 3, 4, 5, 6]

In [273…
```
#if you want to print even number
even_number = list(range(2,100,2))
even_number
```

Out[273…
```
[2,
 4,
 6,
 8,
 10,
 12,
 14,
 16,
 18,
 20,
 22,
 24,
 26,
 28,
 30,
 32,
 34,
 36,
 38,
 40,
 42,
 44,
 46,
 48,
 50,
 52,
 54,
 56,
 58,
 60,
 62,
 64,
 66,
 68,
 70,
 72,
 74,
 76,
 78,
 80,
 82,
 84,
 86,
 88,
 90,
 92,
 94,
 96,
 98]
```

In [274…
```
d = {1:'one', 2:'two', 3:'three'}
d
```

Out[274…
```
{1: 'one', 2: 'two', 3: 'three'}
```

In [275…
```
type(d)
```

Out[275…
```
dict
```

In [276…
```
# print the keys
d.keys()
```

Out[276…
```
dict_keys([1, 2, 3])
```

In [277…
```
# how to get particular value
d[2]
```

Out[277…
```
'two'
```

In [ ]:
```
# other way to get value as
d.
```