In [1]:

```
1 + 1
```

Out[1]:

```
2
```

In [2]:

```
1 * 3
```

Out[2]:

```
3
```

In [3]:

```
1/2
```

Out[3]:

```
0.5
```

In [4]:

```
2 ** 4
```

Out[4]:

```
16
```

In [5]:

```
4 % 2
```

Out[5]:

```
0
```

In [6]:

```
5 % 2
```

Out[6]:

```
1
```

In [7]:

```
(2+3)*(5+5)
```

Out[7]:

```
50
```

In [8]:

```
name_of_var = 2
```

In [9]:

```
x = name_of_var ** 4
```

In [10]:

```
x
```

Out[10]:

```
16
```

In [11]:

```
x = 2
y = 3
```

In [12]:

```
w = x+y
```

In [13]:

```
w
```

Out[13]:

```
5
```

In [14]:

```
greeting = "hello"
```

In [15]:

```
greeting2 = 'hello'
```

In [16]:

```
greeting
```

Out[16]:

```
'hello'
```

In [17]:

```
greeting2
```

Out[17]:

```
'hello'
```

In [18]:

```
quote = "wrap lot's of other quotes"
```

In [19]:

```
quote
```

Out[19]:

```
"wrap lot's of other quotes"
```

In [20]:

```
print(quote)
```

```
wrap lot's of other quotes
```

In [21]:

```
print(x)
```

```
2
```

In [22]:

```
name = 'Yasin'
age = "25"
```

In [23]:

```
print("My nae is {one}, I am {two} years of age".format(one=name,two=age))
```

```
My nae is Yasin, I am 25 years of age
```

In [24]:

```
print("My nae is {one}, I am {two} years of age".format(name,age))
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Input In [24], in <cell line: 1>()
----> 1 print("My nae is {one}, I am {two} years of age".format(name,age))

KeyError: 'one'
```

In [25]:

```
print("My nae is {}, I am {} years of age".format(name,age))
```

```
My nae is Yasin, I am 25 years of age
```

In [26]:

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

Out[26]:

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

In [27]:

```
['hi',1,[1,2]]
```

Out[27]:

```
['hi', 1, [1, 2]]
```

In [30]:

```
my_list = ['a','b','c']
```

In [31]:

```
my_list.append('d')
```

In [32]:

```
my_list
```

Out[32]:

```
['a', 'b', 'c', 'd']
```

In [41]:

```
my_list(0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [41], in <cell line: 1>()
----> 1 my_list(0)

TypeError: 'list' object is not callable
```

In [42]:

```
my_list[0]
```

Out[42]:

```
'a'
```

In [43]:

```
my_list[1]
```

Out[43]:

```
'b'
```

In [44]:

```
my_list[1:]
```

Out[44]:

```
['b', 'c', 'd']
```

In [45]:

```
my_list[:1]
```

Out[45]:

```
['a']
```

In [51]:

```
my_list[0]= 'NEW'
```

In [46]:

```
my_list
```

Out[46]:

```
['a', 'b', 'c', 'd']
```

In [53]:

```
nest = [1,2,3,[4,5,['target']]]
```

In [ ]:

```
nest = [1,2,3,[4,5,['target']]]
```

In [54]:

```
nest[3]
```

Out[54]:

```
[4, 5, ['target']]
```

In [55]:

```
nest[3][2]
```

Out[55]:

```
['target']
```

In [56]:

```
nest[3][2][0]
```

Out[56]:

```
'target'
```

In [57]:

```
nest[3][2]
```

Out[57]:

```
['target']
```

In [58]:

```
greeting3 = "Hello" + ",world!"
```

In [59]:

```
greeting3
```

Out[59]:

```
'Hello,world!'
```

In [60]:

```
nest
```

Out[60]:

```
[1, 2, 3, [4, 5, ['target']]]
```

In [61]:

```
my_list
```

Out[61]:

```
['NEW', 'b', 'c', 'd']
```

In [62]:

```
my_list.append(3.14)
```

In [63]:

```
my_list
```

Out[63]:

```
['NEW', 'b', 'c', 'd', 3.14]
```

In [64]:

```
dictionary = {
    'hello':'world',
    'test':3,
    'foo':'fighters'
}
```

In [65]:

```
dictionary{'hello'}dictionary
```

```
  Input In [65]
    dictionary{'hello'}dictionary
              ^
SyntaxError: invalid syntax
```

In [66]:

```
dictionary['hello']
```

Out[66]:

```
'world'
```

In [67]:

```
dictionary['test']
```

Out[67]:

```
3
```

In [68]:

```
dictionary['foo']
```

Out[68]:

```
'fighters'
```

In [69]:

```
t = (1,2,3)
```

In [70]:

```
t
```

Out[70]:

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

In [71]:

```
t[0]
```

Out[71]:

```
1
```

In [72]:

```
t[2]
```

Out[72]:

```
3
```

In [73]:

```
t[0] = 'NEW'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [73], in <cell line: 1>()
----> 1 t[0] = 'NEW'

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

In [74]:

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

Out[74]:

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

In [75]:

```
{1,2,31,2,1,2,3,3,3,3,2,2,2,1,1,2}
```

Out[75]:

```
{1, 2, 3, 31}
```

In [76]:

```
1>2
```

Out[76]:

```
False
```

In [77]:

```
2>1
```

Out[77]:

```
True
```

In [78]:

```
1>=1
```

Out[78]:

```
True
```

In [79]:

```
1<=1
```

Out[79]:
```
True
```

In [80]:
```
1 == 4
```

Out[80]:
```
False
```

In [81]:
```
1==1
```

Out[81]:
```
True
```

In [82]:
```
'hi' == 'bye'
```

Out[82]:
```
False
```

In [83]:
```
'hi' == 'hi'
```

Out[83]:
```
True
```

In [84]:
```
hi1 = 'hi'
```

In [85]:
```
hi2 = 'hi'
```

In [86]:
```
hi1 == hi2
```

Out[86]:
```
True
```

In [87]:
```
hi1 === hi2
```

```
  Input In [87]
    hi1 === hi2
          ^
SyntaxError: invalid syntax
```

In [88]:
```
(1 > 2) and (2 < 3)
```

Out[88]:
```
False
```

In [89]:
```
(1 > 2) or (2 < 3)
```

Out[89]:
```
True
```

In [90]:

```
( 1 == 2 ) or ( 2 == 3 ) or (4 == 4)
```

Out[90]:

```
True
```

In [91]:

```
 if 1 < 2:
        print("Yep!")
print('Print what so ever')
```

```
Yep!
Print what so ever
```

In [92]:

```
if 1 < 2:
    print('yep!')
```

```
yep!
```

In [93]:

```
 if 1 > 2:
        print("Yep!")
print('Print what so ever')
```

```
Print what so ever
```

In [94]:

```
 if 1 < 2:
    print("first")
else:
    print('second')
```

```
first
```

In [95]:

```
if 1 > 2:
    print("first")
else:
    print('second')
```

```
second
```

In [96]:

```
if 1 == 2:
    print("first")
elif 3 == 3:
    print('second')
else:
    print('last')
```

```
second
```

In [97]:

```
if 1 == 2:
    print("first")
elif 3 != 3:
    print('second')
else:
    print('last')
```

```
last
```

In [98]:

```
seq = [1,2,3,4,5]
```

In [99]:

```
for item in seq:
    print(item)
```

```
1
2
3
4
5
```

In [100…

```
for item in seq:
    print('Yep')
```

```
Yep
Yep
Yep
Yep
Yep
```

In [101…

```
for item in seq:
    print(item ** 2)
```

```
1
4
9
16
25
```

In [102…

```
for item in seq:
    print(item + item)
```

```
2
4
6
8
10
```

In [103…

```
i = 1
while i < 5:
    print('i is {}'.format(i))
    i = i+1
```

```
i is 1
i is 2
i is 3
i is 4
```

In [33]:

```
range(7)
```

Out[33]:

```
range(0, 7)
```

In [34]:

```
for i in range(7):
    print(i)
```

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

In [35]:

```
list(range(7))
```

Out[35]:

```
[0, 1, 2, 3, 4, 5, 6]
```

In [36]:

```
for n in range(10,20):
    print(n)
```

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

In [39]:

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

In [40]:

```
out = []
for item in x:
    out.append(item ** 2)
print(out)
```

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

In [104…

```
[item**2 for item in x]
```

Out[104…

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

In [106…

```
def square_num(val=0):
    """
    This function squares the input value provided
    IE: val^2
    """
    print(val**2)
```

In [107…

```
square_num
```

Out[107…

```
<function __main__.square_num(val=0)>
```

In [108…

```
square_num()
```

```
0
```

In [109…

```
square_num(5)
```

```
25
```

In [110…

```
square_num(val=11)
```

```
121
```

In [111…

```
def cube(x):
    return x**3
```

In [112…

```
cubes = cube(3)
```

In [113…

```
cubes
```

Out[113…

```
27
```

In [116…

```
cubes2 = [cube(x) for x in range(1,5)]
```

In [117…

```
cubes2
```

Out[117…

```
[1, 8, 27, 64]
```

In [118…

```
def hello_world():
    print("hello world!")
```

In [119…

```
hello_world()
```

```
hello world!
```

In [126…

```
def hello_person(name=None):
    if (name == None):
        print("hello!")
    else:
        print("hello " + name + "!")
```

In [127…

```
hello_person()
```

```
hello!
```

In [128…

```
hello_person("Yasin")
```

```
hello Yasin!
```

In [130…

```
def times2(var):
    return var*2
```

In [131…

```
times2(5)
```

Out[131…

```
10
```

In [148…

```
var_lambda = lambda test:test*3
```

In [149…

```
print(var_lambda(3))
```

```
9
```

In [150…

```
[var_lambda(i) for i in range(9)]
```

```
Out[150…  [0, 3, 6, 9, 12, 15, 18, 21, 24]
```

```
In [160…  seq = [1,2,3,4,5]
```

```
In [161…  seq
```

```
Out[161…  [1, 2, 3, 4, 5]
```

```
In [162…  map(var_lambda,seq)
```

```
Out[162…  <map at 0x7fbe9bfd9df0>
```

```
In [163…  list(map(lambda lambda_exp:2**lambda_exp,seq))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [163], in <cell line: 1>()
----> 1 list(map(lambda lambda_exp:2**lambda_exp,seq))

TypeError: 'list' object is not callable
```

```
In [165…  sseq = [1,2,3,4,5]
          li = list(map(lambda lambda_exp:2**lambda_exp,seq))
          print(li)
          # works on
          # https://www.w3schools.com/python/trypython.asp?filename=demo_lambda2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [165], in <cell line: 2>()
      1 sseq = [1,2,3,4,5]
----> 2 li = list(map(lambda lambda_exp:2**lambda_exp,seq))
      3 print(li)

TypeError: 'list' object is not callable
```

```
In [166…  seq = [1,2,3,4,5]
          lambda_e = lambda lambda_exp:2**lambda_exp
          li = list(map(lambda_e,seq))
          print(li)
          # works on
          # https://www.w3schools.com/python/trypython.asp?filename=demo_lambda2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [166], in <cell line: 3>()
      1 seq = [1,2,3,4,5]
      2 lambda e = lambda lambda exp:2**lambda_exp
----> 3 li = list(map(lambda_e,seq))
      4 print(li)

TypeError: 'list' object is not callable
```

In [168…

```
filter(lambda item:item%2 == 0,seq)
```

Out[168…

```
<filter at 0x7fbe9c478ee0>
```

In [170…

```
list(filter(lambda item:item%2 == 0,seq))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [170], in <cell line: 1>()
----> 1 list(filter(lambda item:item%2 == 0,seq))

TypeError: 'list' object is not callable
```

In [171…

```
seq = [1,2,3,4,5]
result = list(filter(lambda item:item%2 == 0,seq))
print(result)
# works on
# https://www.w3schools.com/python/trypython.asp?filename=demo_lambda2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [171], in <cell line: 2>()
      1 seq = [1,2,3,4,5]
----> 2 result = list(filter(lambda item:item%2 == 0,seq))
      3 print(result)

TypeError: 'list' object is not callable
```

In [172…

```
st = 'Hello My Name Is Yasin'
```

In [173…

```
st
```

Out[173…

```
'Hello My Name Is Yasin'
```

In [174…

```
st.lower()
```

Out[174…

```
'hello my name is yasin'
```

In [175…

```
st.upper()
```

Out[175…

```
'HELLO MY NAME IS YASIN'
```

In [176…

```
st.split()
```

Out[176…

```
['Hello', 'My', 'Name', 'Is', 'Yasin']
```

In [177…

```
tweet = 'Go Galatasaray, #Sports'
```

In [178…

```
tweet.split('#')
```

Out[178…

```
['Go Galatasaray, ', 'Sports']
```

In [179…

```
tweet.split('#')[0]
```

Out[179…

```
'Go Galatasaray, '
```

In [180…

```
tweet.split('#')[1]
```

Out[180…

```
'Sports'
```

In [181…

```
dictionary
```

Out[181…

```
{'hello': 'world', 'test': 3, 'foo': 'fighters'}
```

In [182…

```
dictionary.keys()
```

Out[182…

```
dict_keys(['hello', 'test', 'foo'])
```

In [183…

```
dictionary.items()
```

Out[183…

```
dict_items([('hello', 'world'), ('test', 3), ('foo', 'fighters')])
```

In [186…

```
lst = [1,2,3]
```

In [187…

```
lst.pop()
```

Out[187…

```
3
```

In [188…

```
lst
```

Out[188…

```
[1, 2]
```

In [189…

```
'x' in [1,2,3]
```

Out[189…

```
False
```

In [190…

```
'x' in ['x','y','z']
```

Out[190…

```
True
```

In [ ]: