```
# https://github.com/docaotien/Lab-MKTG5883.N22.CTTT
```

```
1 + 1
```

```
     2
```

```
1 * 3
```

```
     3
```

```
1 / 2
```

```
     0.5
```

```
2 ** 4
```

```
     16
```

```
4 % 2
```

```
     0
```

```
5 % 2
```

```
     1
```

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

```
     50
```

```
'single quotes'
```

```
     'single quotes'
```

```
'double quotes'
```

```
     'double quotes'
```

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

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

```
x = 'Hello'
```

```
x
```

```
     'Hello'
```

```
print(x)
```

```
     Hello
```

```
num = 12
name = 'Sam'
```

```
print('My name is:{one}, and my number is: {two}'.format(one= 12, two = 'Sam') )
```

```
     My name is:12, and my number is: Sam
```

```
print('My number is: {}, and my name is: {}'.format(num,name))
```

```
     My number is: 12, and my name is: Sam
```

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

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

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

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

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

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

```
my_list
```

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

```
my_list[0]
```

```
     'a'
```

```
my_list[1]
```

```
     'b'
```

```
my_list[1:]
```

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

```
my_list[:1]
```

```
     ['a']
```

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

```
my_list
```

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

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

```
nest[3]
```

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

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

```
     ['target']
```

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

```
     'target'
```

```
d={'key':'item1','key2':'item2'}
```

```
d
```

```
     {'key': 'item1', 'key2': 'item2'}
```

```
d.keys()
```

```
     dict_keys(['key', 'key2'])
```

```
True
```

```
     True
```

```
False
```

```
     False
```

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

```
t[0]
```

```
     1
```

```
t1=list(t)
t1[0]= 'NEW'
t=list(t1)
```

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

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

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

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

```
1>2
```

```
     False
```

```
1<2
```

```
     True
```

```
1 >=1
```

```
     True
```

```
1 <=4
```

```
     True
```

```
1 == 1
```

```
     True
```

```
'hi' == 'bye'
```

```
     False
```

```
(1>2) and (2<3)
```

```
False
```

```
(1>2) or (2<3)
```

```
True
```

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

```
True
```

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

```
Yeb!
```

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

```
yeb!
```

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

```
first
```

```
if 1 == 2:
  print('first')
elif 3 == 3:
    print('midle')
else:
  print('Last')
```

```
midle
```

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

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

```
1
2
3
```

```
    4
    5
```

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

```
    yeb
    yeb
    yeb
    yeb
    yeb
```

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

```
    2
    4
    6
    8
    10
```

```
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
```

```
range(5)
```

```
    range(0, 5)
```

```
for i in range(5):
  print(i)
```

```
    0
    1
    2
    3
    4
```

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

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

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

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

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

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

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

```
def my_func(param1='default'):
  """
  Docstring goes here.
  """
  print(param1)
```

```
my_func
```

```
    <function __main__.my_func(param1='default')>
```

```
my_func()
```

```
    default
```

```
my_func('new param')
```

```
    new param
```

```
my_func(param1='new param')
```

```
    new param
```

```
def square(x):
  return x**2
```

```
out = square(2)
```

```
print(out)
```

```
     4
```

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

```
times2(2)

     4
```

```
lambda var: var*2

     <function __main__.<lambda>(var)>
```

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

```
map(times2,seq)

     <map at 0x7fac67420ee0>
```

```
list(map(times2,seq))

     [2, 4, 6, 8, 10]
```

```
list(map(lambda var: var*2,seq))

     [2, 4, 6, 8, 10]
```

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

     <filter at 0x7fac674479d0>
```

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

     [2, 4]
```

```
st = 'hello my name is Sam'
```

```
st.lower()

     'hello my name is sam'
```

```
st.upper()

     'HELLO MY NAME IS SAM'


st.split()

     ['hello', 'my', 'name', 'is', 'Sam']


tweet = 'Go Sport: #Sports'


tweet.split('#')

     ['Go Sport: ', 'Sports']


tweet.split('#')[1]

     'Sports'


d

     {'key': 'item1', 'key2': 'item2'}


d.keys()

     dict_keys(['key', 'key2'])


d.items()

     dict_items([('key', 'item1'), ('key2', 'item2')])


lst =[1,2,3]


lst.pop()

     3


lst

     [1, 2]


'x' in [1,2,3]

     False
```

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

```
True
```

```
7 **4
```

```
2401
```

```
s = "Hi there Sam!"
```

```
t = s.split()
```

```
t1=list(t)
t1[2]= 'dad!'
t=list(t1)
```

```
t
```

```
['Hi', 'there', 'dad!']
```

```
planet = "Earth"
diameter = 12742
```

```
print('The diameter of {}  is {} kilometers'.format(planet,diameter))
```

```
The diameter of Earth  is 12742 kilometers
```

```
lst = [1,2,[3,4],[5,[100,200,['hello']],24,11],1,7]
```

```
lst[3][1][2]
```

```
['hello']
```

```
d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
```

```
d['k1'][3]['tricky'][3]['target'][3]
```

```
'hello'
```

```
# Tupe is immutable
```

```
def domainGet(email):
    return email.split('@')[-1]
```

```
domainGet('user@domain.com')
```

```
'domain.com'
```

```
def findDog(st):
    return 'dog' in st.lower().split()
```

```
findDog('Is there a dog here?')
```

```
True
```

```
def countDog(st):
    count = 0
    for word in st.lower().split():
        if word == 'dog':
            count += 1
    return count
```

```
countDog('This dog runs faster than the other dog dude!')
```

```
2
```

```
seq = ['soup','dog','salad','cat','great']
```

```
list(filter(lambda word: word[0]=='s',seq))
```

```
['soup', 'salad']
```

```
def caught_speeding(speed, is_birthday):

    if is_birthday:
        speeding = speed - 5
    else:
        speeding = speed

    if speeding > 80:
        return 'Big Ticket'
    elif speeding > 60:
        return 'Small Ticket'
    else:
        return 'No Ticket'
```

```
caught_speeding(81,True)
```

```
'Small Ticket'
```

```
caught_speeding(81,False)
```

```
'Big Ticket'
```

```
caught_speeding(40,0)
```

```
'No Ticket'
```