```
# An empty tuple
empty_tuple = ()
print (empty_tuple)


# Creating non-empty tuples
# One way of creation
tup = 'python', 'rpics'
print(tup)

# Another way for doing the same
tup = ('python', 'rpics')
print(tup)


# Code for concatenating 2 tuples

tuple1 = (0, 1, 2, 3)
tuple2 = ('python', 'rpics')

# Concatenating above two
print(tuple1 + tuple2)


# Creating nested tuples
tuple1 = (0, 1, 2, 3)
tuple2 = ('python', 'rpics')
tuple3 = (tuple1, tuple2)
print(tuple3)



# Create a tuple with repetition

tuple4 = ('rpics',)*3
print(tuple4)



#code to test that tuples are immutable
tuple1 = (0, 1, 2, 3)
tuple4=(4,1,2,3)
tuple1[0] = 4
print(tuple1)
```

```
# code to test slicing

tuple1 = (0 ,1, 2, 3)
print(tuple1[1:])
print(tuple1[::-1])
print(tuple1[2:4])



# Printing the length of a tuple
tuple2 = ('python', 'rpics')
print(len(tuple2))

#Some methods in tuples
tup1=('a','b','a')
tup1.count('a')
##index shows the first time index appears in the tuple
tup1.index('a')
tup1.index('b')
##Booloean output with in as keyword
'a' in tup1
1 in tup1



##Multiple assignments
t1=(1,2,3)
t1
a,b,c=t1
a
b
c



# Split a two-digit number into its tens and ones digit
def divprob(n):
   tens = n // 10 ##returns the quotient
   ones = n % 10 ##returns the remainder
   return (tens, ones)

y = 57
ten, one = divprob(y)
print(y, "has tens digit = ", ten, "and ones digit = ", one)
```

```
##Pass a tuple to a function
def hello(str1):
   return str1[0]+str1[2]+str1[3]

d=('h','e','l','l','o')
print(hello(d))



##Call module
import Area

r = 6
h = 10
a1 = Area.circle(r)       #  Call a module function
a2 = Area.cylinder(r, h)   #  Call a module function
a3 = Area.sphere(r)       #  Call a module function
a4= Area.cone(r,h)

#path to a module
import Area
Area.__file__


##Another way of calling a module
from Area import circle



##Images

from PIL import Image
filename="Bloom.jpg"
im = Image.open(filename)
im.show()
im.size

print("Here's the information about", im)
print(im.format, im.size, im.mode)
```

```
##Crop
im2 = im.crop((0, 0, 600, 600))
im2.show()

##Rotate
im3=im.rotate(180)
im3.show()

##Convert to grayscale
gray_im = im.convert('L')
gray_im.show()
#Resize the image
scaled = gray_im.resize((128, 128))
#Check the attributes
print(scaled.format, scaled.size, scaled.mode)
#Show the new image
scaled.show()
##Save the new file
scaled.save(filename + "_scaled.jpg")



##Copy and paste
from PIL import Image
filename1="sheep.jfif"
im4 = Image.open(filename1)
im4.show()
im4.size



#New blank image
im5 = Image.new('RGB', (500, 375*2))
im5.show()
im5.paste( im4, (0,0))   ##not assigning the result of paste to a new variable
im5.show()
im5.paste( im4, (0, 380))
im5.show()
```

```
##Paste Images
from PIL import Image

im6 = Image.open("lego.jfif")
im6.show()
w,h = im6.size

## Crop out three columns from the image
## Note: the crop function returns a new image
part1 = im6.crop((0,0,w//3,h))
part2 = im6.crop((w//3,0,2*w//3,h))
part3 = im6.crop((2*w//3,0,w,h))

## Create a new image
newim = Image.new("RGB",(w,h))
newim.show()
## Paste the image in different order
## Note: the paste function changes the image it is applied to
newim.paste(part3, (0,0))
newim.paste(part1, (w//3,0))
newim.paste(part2, (2*w//3,0))
newim.show()
##Original image was:
im6.show()
```