```
# Functions
# Set of Tasks (Block of Code) grouped together to achieve certain
functionality
# Advantages - Re-usability, Easy to Maintain, Reduces time while
modifying, enhances modularity

def chkMailId(mail_id):
  msg = ""
  length = 0

  if (mail_id.count('@') == 1):
    msg = "Valid Mail Id"
    length = len(mail_id)
  else:
    msg = "Invalid Mail Id"

  return (mail_id, msg, length)

# Invoke the function
mail_id, status, length = chkMailId("mohan@gmail.com")
if (status is None):
  print ("Returned no values")
else:
  print (mail_id, "->", status, "->", length)

mail_id, status, length = chkMailId("mohangmail.com")
if (status is None):
  print ("Returned no values")
else:
  print (mail_id, "->", status, "->", length)

print
("---------------------------------------------------------------------")

# Passing Parameters
# Fixed Number of Arguments -> Positional Arguments
# All the arguments should be passed and order of passing arguments
should be maintained

print ("Fixed Number of Arguments -> Positional Arguments")

def displayEmp_1(name, location, salary):
  print ("Name - {0}, Location - {1}, Salary - {2}".format(name,
location[0], salary))
  location.append("Chennai")
  print ("Location Updated ->", location)

displayEmp_1("Justin", ["Bangalore", "Goa"], 15000)
displayEmp_1("Justin", [15000], "Bangalore")
```

```
print
("--------------------------------------------------------------------")

# Keyword Parameters
# All the arguments should be passed and order of passing arguments can
be changed

print ("Keyword Parameters")

def displayEmp_2(name, location, salary):
  print ("Name - {0}, Location - {1}, Salary - {2}".format(name,
location, salary))

displayEmp_2(name="justin", location="chennai", salary=20000)
displayEmp_2(salary=25000, location="bangalore", name="davies")

print
("--------------------------------------------------------------------")

# Default Parameters
# Arguments which doesn't have default value declared not required to
pass
# Non Default Arguments is followed by Default Arguments

print ("Default Parameters")

def displayEmp_3(name, location, salary=30000):
  print ("Name - {0}, Location - {1}, Salary - {2}".format(name,
location, salary))

displayEmp_3(name="justin", location="chennai")
displayEmp_3(salary=25000, location="bangalore", name="davies")

print
("--------------------------------------------------------------------")

# Variable Length Arguments
# User can pass any number of arguments

def displayNames(*names):
  for name in names:
    print ("Name ->", name)

  print ("************************")

displayNames("justin", "davies")
displayNames("justin", "davies", "tris")
displayNames("justin", "davies", "tris", "damien", "gifford")

def displayDetails(**details):
  for name, detail in details.items():
    print (name, '->', detail)
  print ("************************")

displayDetails(salary=25000, location="bangalore", name="davies")
displayDetails(name="justin", location="chennai")

print
("--------------------------------------------------------------------")

# Lambda Function
# One line function and it is also called as anonymous function
# It can take any number of parameters, but it evaluates only one
expression

sqr = lambda num: num ** 2
print ("Square of 25 ->", sqr(25))

calc = lambda a, b, c, d, e: ((a + b) * (c + d) * e) * 5
print ("Calculator ->", calc(5, 4, 6, 7, 8))

chk_number = lambda num1, num2: (num1 > num2)
print ("Check Number 1 ->", chk_number(5, 3))
print ("Check Number 2 ->", chk_number(3, 5))

print
("--------------------------------------------------------------------")
```