```python
def function(x):
    print("Id inside function before assignment:", id(x))
    x = 10
    print("Id inside function after assignment:", id(x))
    print("Value inside funtion:", x)

x = 5
print("Id ouside function before function call:", id(x))
function(x)
print("Id ouside function after function call:", id(x))
print("Value ouside function:", x)
```

```
Id ouside function before function call: 140715184067520
Id inside function before assignment: 140715184067520
Id inside function after assignment: 140715184067680
Value inside funtion: 10
Id ouside function after function call: 140715184067520
Value ouside function: 5
```

```python
def function(x):
    print("Id inside function before assignment:", id(x))
    x = 5
    print("Id inside function after assignment:", id(x))
    print("Value inside funtion:", x)

x = 5
print("Id ouside function before function call:", id(x))
function(x)
print("Id ouside function after function call:", id(x))
print("Value ouside function:", x)
```

```
Id ouside function before function call: 140715184067520
Id inside function before assignment: 140715184067520
Id inside function after assignment: 140715184067520
Value inside funtion: 5
Id ouside function after function call: 140715184067520
Value ouside function: 5
```

```python
def function(y):
    print("Id inside function before assignment:", id(y))
    y = 10
    print("Id inside function after assignment:", id(y))
    print("Value inside funtion:", y)

x = 5
print("Id ouside function before function call:", id(x))
function(x)
print("Id ouside function after function call:", id(x))
print("Value ouside function:", x)
```

```
Id ouside function before function call: 140715184067520
Id inside function before assignment: 140715184067520
Id inside function after assignment: 140715184067680
Value inside funtion: 10
Id ouside function after function call: 140715184067520
Value ouside function: 5
```

```python
def fun(my_str):
    print("Id inside fun before append:", id(my_str))
    my_str + " world"
    print("Id inside fun after append:", id(my_str))
    print("my_str inside fun:", my_str)
my_str = "Hello"
print("Id outside fun before fun call:", id(my_str))
print("my_str outside fun before fun call:",my_str)
fun(my_str)
print("Id outside after fun call:", id(my_str))
print("my_str outside after fun call:", my_str)
```

```
Id outside fun before fun call: 2013692429848
my_str outside fun before fun call: Hello
Id inside fun before append: 2013692429848
Id inside fun after append: 2013692429848
my_str inside fun: Hello
Id outside after fun call: 2013692429848
my_str outside after fun call: Hello
```

```python
def fun(my_str):
    print("Id inside fun before append:", id(my_str))
    my_str = my_str + " world"
    print("Id inside fun after append:", id(my_str))
    print("my_str inside fun:", my_str)
my_str = "Hello"
print("Id outside fun before fun call:", id(my_str))
print("my_str outside fun before fun call:",my_str)
fun(my_str)
print("Id outside after fun call:", id(my_str))
print("my_str outside after fun call:", my_str)
```

```
Id outside fun before fun call: 2013692429848
my_str outside fun before fun call: Hello
Id inside fun before append: 2013692429848
Id inside fun after append: 2013692619824
my_str inside fun: Hello world
Id outside after fun call: 2013692429848
my_str outside after fun call: Hello
```

```python
def fun(my_list1):
    print("my_list1 before append", my_list1)
    print("my_list1 Id before append:", id(my_list1))
    my_list1.append(9)
    print("my_list1 Id after append:", id(my_list1))
    print("my_list1 inside fun:", my_list1)
my_list = [5,7,8]
print("my_list Id before fun call:", id(my_list))
print("my_list before fun call:",my_list)
fun(my_list)
print("my_list Id after fun call:", id(my_list))
print("my_list after fun call:", my_list)
print("my_list1 outside fun:", my_list1)
```

```
my_list Id before fun call: 2013692513544
my_list before fun call: [5, 7, 8]
my_list1 before append [5, 7, 8]
my_list1 Id before append: 2013692513544
my_list1 Id after append: 2013692513544
my_list1 inside fun: [5, 7, 8, 9]
my_list Id after fun call: 2013692513544
my_list after fun call: [5, 7, 8, 9]
NameError: name 'my_list1' is not defined
```

```python
def fun(my_list):
    print("Id inside fun before append:", id(my_list))
    my_list.append(9)
    print("Id inside fun after append:", id(my_list))
    print("my_list inside fun:", my_list)
my_list = [5,7,8]
print("Id ouside fun before fun call:", id(my_list))
print("my_list outside fun before fun call:", my_list)
fun(my_list)
print("Id ouside fun after fun call:", id(my_list))
print("my_list ouside fun after fun call:", my_list)
```

```
Id ouside fun before fun call: 2013692463368
my_list outside fun before fun call: [5, 7, 8]
Id inside fun before append: 2013692463368
Id inside fun after append: 2013692463368
my_list inside fun: [5, 7, 8, 9]
Id ouside fun after fun call: 2013692463368
my_list ouside fun after fun call: [5, 7, 8, 9]
```

```python
def fun(my_list):
    print("my_list inside fun before assignment", my_list)
    print("Id inside fun before assignment:", id(my_list))
    my_list = [1,'IC152',2,3]
    print("Id inside fun after assignment:", id(my_list))
    print("my_list inside fun:", my_list)
my_list = [5,7,8]
print("Id ouside fun before fun call:", id(my_list))
print("my_list outside fun before fun call:", my_list)
fun(my_list)
print("Id ouside fun after fun call:", id(my_list))
print("my_list ouside fun after fun call:", my_list)
```

```
Id ouside fun before fun call: 2013692514824
my_list outside fun before fun call: [5, 7, 8]
my_list inside fun before assignment [5, 7, 8]
Id inside fun before assignment: 2013692514824
Id inside fun after assignment: 2013692617992
my_list inside fun: [1, 'IC152', 2, 3]
Id ouside fun after fun call: 2013692514824
my_list ouside fun after fun call: [5, 7, 8]
```

Home Work

Questions:

1. Do the same exercise with list for different kind of list methods?
2. Pass tuple or set to function and apply tuple or string methods inside function and figure out what will happen?

Variable Length Argument

```
def Districts(MyDistrict, *Neighbour):
   print("My district is " + MyDistrict)

   if len(Neighbour) > 0:
      for i in range(len(Neighbour)):
         print("Neighbour {:2d} is {}.".format(i+1,
             Neighbour[i]))

Neighbouring = ("Bilaspur", "Hamirpur", "Kangra", "Kullu",
    "Shimla")
Districts("Mandi", Neighbouring)
```

```
My district is Mandi
Neighbour  1 is ('Bilaspur', 'Hamirpur', 'Kangra', 'Kullu',
    'Shimla').
```

Variable Length Argument

```
def Districts(MyDistrict, *Neighbour):
   print("My district is " + MyDistrict)

   if len(Neighbour) > 0:
      for i in range(len(Neighbour)):
         print("Neighbour {:2d} is {}.".format(i+1,
             Neighbour[i]))

Districts("Mandi")
Districts("Mandi", "Bilaspur", "Hamirpur", "Kangra",
    "Kullu", "Shimla")
```

```
My district is Mandi
My district is Mandi
Neighbour  1 is Bilaspur.
Neighbour  2 is Hamirpur.
Neighbour  3 is Kangra.
Neighbour  4 is Kullu.
Neighbour  5 is Shimla.
```

Variable Length Argument

```
def Districts(MyDistrict, *Neighbour):
   print("My district is " + MyDistrict)

   if len(Neighbour) > 0:
      for i in range(len(Neighbour)):
         print("Neighbour {:2d} is {}.".format(i+1,
             Neighbour[i]))

Neighbouring = ("Bilaspur", "Hamirpur", "Kangra", "Kullu",
    "Shimla")
Districts("Mandi", *Neighbouring)
```

```
My district is Mandi
Neighbour  1 is Bilaspur.
Neighbour  2 is Hamirpur.
Neighbour  3 is Kangra.
Neighbour  4 is Kullu.
Neighbour  5 is Shimla.
```

Variable Length Argument

```
def Districts(MyDistrict, *Neighbour):
   print("My district is " + MyDistrict)

   if len(Neighbour) > 0:
      for i in range(len(Neighbour)):
         print("Neighbour {:2d} is {}.".format(i+1,
             Neighbour[i]))

Neighbouring = ("Bilaspur", "Hamirpur", "Kangra", "Kullu",
    "Shimla")
Districts(*Neighbouring)
```

```
My district is Bilaspur
Neighbour  1 is Hamirpur.
Neighbour  2 is Kangra.
Neighbour  3 is Kullu.
Neighbour  4 is Shimla.
```

Variable Length Argument

```
def fun(**course):
    print("---------------")
    print(type(course))
    for key, value in course.items():
        print ("%s : %s" %(key, value))
    print("---------------")

fun(IC150 = "Computing", IC110 = "Mathematics", IC160 =
    "Electrical")
```

```
---------------
<class 'dict'>
IC150 : Computing
IC110 : Mathematics
IC160 : Electrical
---------------
```

Variable Length Argument

```
def Mean(*x):
    S = 0
    for i in range(len(x)):
        S += x[i]
    if len(x) > 0:
       S = S/len(x)
    print ("The mean is ", S)
    return S

Mean()
Mean(10)
Mean(67, 52, 70, 69, 86)
Marks = [67, 52, 70, 69, 86]
Mean(*Marks)
Mean(Marks)
```

```
The mean is  0
The mean is  10.0
The mean is  68.8
The mean is  68.8
Error
```

Variable Length Argument

```
def fun(**course):
    print("---------------")
    print(type(course))
    for key, value in course.items():
        print ("%s : %s" %(key, value))
    print("---------------")

CourseDic = {"IC150" : "Computing", "IC110" :
    "Mathematics", "IC160" : "Electrical"}
fun(**CourseDic)
```

```
---------------
<class 'dict'>
IC150 : Computing
IC110 : Mathematics
IC160 : Electrical
---------------
```

Function Call

```
def fun():
    print(S, id(S))

S = "Data Science"
print(S, id(S))
fun()
print(S, id(S))
```

```
Data Science 2139013498352
Data Science 2139013498352
Data Science 2139013498352
```

global Variables

```
def fun():
    global S
    print(S, id(S))
    S = S + " & Engineering"
    print(S, id(S))

S = "Data Science"
print(S, id(S))
fun()
print(S, id(S))
```

```
Data Science 2139013317616
Data Science 2139013317616
Data Science & Engineering 2139013437360
Data Science & Engineering 2139013437360
```

global Variables

```
def fun():
    global S
    print(S, id(S))
    S = "Data Scinece & Engineering"
    print(S, id(S))

S = "Data Science"
print(S, id(S))
fun()
print(S, id(S))
```

```
Data Science 2139013500080
Data Science 2139013500080
Data Scinece & Engineering 2139013437360
Data Scinece & Engineering 2139013437360
```