Python for Engineers

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Range Functions in python

The built-in function `range()` generates the **integer numbers between the given start integer to the stop integer**.

It returns a range of object.
Using `for` loop, we can iterate over a sequence of numbers produced by the `range()` function.

Range() function syntax and arguments

Range(start, stop, step)

It takes three arguments. Out of the three 2 arguments are optional. I.e., start and step are the optional arguments.

A **start** argument is a starting number of the sequence. i.e., lower limit. By default, it starts with 0 if not specified.

A **stop** argument is an upper limit. i.e., generate numbers up to this number, The range() doesn't include this number in the result.

The **step** is a difference between each number in the result. The default value of the step is 1 if not specified.

Example

```
print("Python range() example")
print("Get numbers from range 0 to 6") for i
in range(6):
  print(i, end=', ‘)
```

NOTE- This will generate numbers from 0 to 5 because range() function doesn’t include the last number

Writing User Defined Functions in Python

- A function is a block of code which only runs when it is called.
- You can pass data, known as parameters, into a function.
- A function can return data as a result.

Functions cntd…

```
def my_function():
  print("Hello from a function")
```

my_function()

contd.....

```
def my_function(fname):
  print(fname + " Refsnes")

my_function("Emil")
my_function("Tobias")
my_function("Linus")
```

```
def my_function(fname, lname):
  print(fname + " " + lname)

my_function("Emil", "Refsnes")
```

Parameter passing

```
def sum(a,b):
    c=a+b
    return(c)


Print(sub(90,10))
```

```
def sum(a,b):
    c=a+b
    return(c)
k=90
m=89
Print(sub(k,m))
```

Passing list to a function

```
def my_function(food):
  for x in food:
    print(x)

fruits = ["apple", "banana", "cherry"]

my_function(fruits)
```

Recursion

```
def tri_recursion(k):
  if(k > 0):
    result = k + tri_recursion(k - 1)
    print(result)
  else:
    result = 0
  return result

print("\n\nRecursion Example Results")
tri_recursion(6)
```

Default Arguments

```
def printinfo( name, age ):
"This prints a passed info into this function"
print "Name: ", name
print "Age ", age
return; # Now you can call printinfo function printinfo( age=50, name="miki" )
```

```
total = 0
 # This is global variable.
# Function definition is here def sum( arg1, arg2 ):
          # Add both the parameters and return them.”
          total = arg1 + arg2
         # Here total is local variable.
          print "Inside the function local total : ", total
          return total;
# Now you can call sum function
sum( 10, 20 );
print "Outside the function global total : ", total
```

```
def changeme( mylist ):
      "This changes a passed list into this function”
       mylist.append([1,2,3,4]);
      print "Values inside the function: ", mylist
      Return
 # Now you can call changeme function
 mylist = [10,20,30];
changeme( mylist );
print "Values outside the function: ", mylist
```

```
def changeme( mylist ):
    "This changes a passed list into this function"
    mylist = [1,2,3,4];
    # This would assig new reference in mylist
    print "Values inside the function: ", mylist
    return
# Now you can call changeme function
mylist = [10,20,30];
changeme( mylist );
print "Values outside the function: ", mylist
```

```
def printinfo( name, age ):
    "This prints a passed info into this
    function"
    print "Name: ", name
    print "Age ", age
    return;
# Now you can call printinfo function
printinfo( age=50, name="miki" )
```

```
# Function definition is here
def printinfo( name, age = 35 ):
    #"This prints a passed info into this function"
    print "Name: ", name
    print "Age ", age
    return;
# Now you can call print info function
printinfo( age=50, name="miki" )
printinfo( name="miki" )
```

Recursive Functions

```
sum=0
print ('Hello World')
def test(sum,n):
    if(n==0):
        return
    else:

        d=int(n%10)
        sum=sum+d
        print(sum)
        test(sum,int(n/10))

test(0,789)
```

Recursive Function

```
def fib(n):
    if(n<=1):
        return(n)
    else:
        return(fib(n-1)+fib(n-2))

for i in range(0,5):
    print(fib(i))
```

Recursive Function

```
def bin_search(low,high,L,key):


    if(low<=high):
        mid=int((low+high)/2)
        print(mid)
        if(L[mid]==key):
            print("match")
            #return(mid)
            print("match at",mid)
            return(mid)
    elif(key>L[mid]):
            bin_search(mid+1,high,L,key)
        else:
            bin_search(low,mid-1,L,key)

L=[23,45,67,89,90]
key=int(input("enter key to search"))
bin_search(0,4,L,key)
```

```
def bubblesort(L):
  for i in range(0,(len(L)-1)):
      comp=0
      swap=0
      for j in range(0,(len(L)-1-i)):
         if(L[j]>L[j+1]):
            temp=L[j]
            L[j]=L[j+1]
            L[j+1]=temp
            swap=swap+1
         comp=comp+1
      print("Output of iteration:",i+1)
      print(L)
      print("number of comparisions=",comp)
      print("Number of swappings=",swap)
```

```
def selection_sort(L):
    for i in range(0,len(L)): n
        min=L[i]
        pos=i
        for j in range(i+1,len(L)): n
            if(min>L[j]):
                min=L[j]
                pos=j
        temp=L[i]
        L[i]=L[pos]
        L[pos]=temp
        print("output of iteration ",i+1)
        print(L)
```

MIT-WPU
।। विश्वशान्तिर्ध्रुवं ध्रुवा ।।

Thank You!