Loops

Topics covered

- While loop
- break
- continue
- for loop
- nested loops

While Loop - Key Concept

- Syntax:

```
while(condition):
    code_block
```

- Infinite Loop

```
while True:
    code_block

while(1):
    code_block
```

In [1]:
```
i = 0
while i<5:
    print (i*i)
    i += 1
```

```
0
1
4
9
16
```

break statement

- used to break out of the innermost loop.

Example: The while loop to print the squares of all the numbers upto 5, can be modified using the break statement.

In [2]:
```
i = 0

while (1):
    print (i*i)
    i += 1
    if i == 5:
        break
```

```
0
1
4
9
16
```

In [3]:
```
while True:
    user_input = input()
    if user_input == "quit":
        print ("thank you!")
        break
    else:
        print ("You typed '%s'" %user_input)
```

```
eYRC-2020
You typed 'eYRC-2020'
quit
thank you!
```

Continue statement

- used to skip execution of the rest of the loop on this iteration and continue to the end of the iteration.

Example:

Modify the program to print squares of all number other than zero and one

In [4]:
```
i = 0

while (i<5):
    if i<2:
        i += 1
        continue

    print (i*i)
    i += 1
```

```
4
9
16
```

Range function

Syntax: range (arg1, arg2, arg3)

- First argument: start value
- Second argument: stop value
- Third argument: step-size.

The range function returns a list of values from the start value to the stop value (not included), moving in steps of size given by the step-size argument.

In [5]:
```
for i in range (0,5):
    print (i)
```

```
0
1
2
3
4
```

In [6]:
```
for i in range(0,10,2):
    print (i)
```

```
0
2
4
6
8
```

In [7]:
```
for i in range (1,5):
    print ("")
    for j in range (1,5):
        print ("%d * %d = %d" %(i,j,i*j))
```

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12

4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
```