WHILE LOOPS

- Similar to a **for** loop, **while** loops output each element, but based on a **while** statement being True.
- Once the **while** statement is no longer true, the **while** loop ends.

In [51]:

```
x = 50
while x <= 100:
    print(x)
    x += 10
```

```
50
60
70
80
90
100
```

In [52]:

```
num = 1
while num < 20:
    print("The multiplication is: ", num)
    num *= 2
```

```
The multiplication is:  1
The multiplication is:  2
The multiplication is:  4
The multiplication is:  8
The multiplication is:  16
```

In [54]:

```
div = 1000
while div > 10:
    print("The division is: ", round(div), 2)
    div /= 3
```

```
The division is:  1000 2
The division is:  333 2
The division is:  111 2
The division is:  37 2
The division is:  12 2
```

In [56]:

```
floor = 200
while floor > 10:
    print("The floor is: ", floor)
    floor /= 2
```

```
The floor is:  200
The floor is:  100.0
```

```
The floor is:  50.0
The floor is:  25.0
The floor is:  12.5
```

In [57]:

```
sub = 10
while sub > 0:
    print("Count down to lift off in ", str(sub) + " seconds!")

    sub -= 1
    if sub == 0:
        print("Blast off!")
```

```
Count down to lift off in  10 seconds!
Count down to lift off in  9 seconds!
Count down to lift off in  8 seconds!
Count down to lift off in  7 seconds!
Count down to lift off in  6 seconds!
Count down to lift off in  5 seconds!
Count down to lift off in  4 seconds!
Count down to lift off in  3 seconds!
Count down to lift off in  2 seconds!
Count down to lift off in  1 seconds!
Blast off!
```

In [59]:

```
p = 1
while p < 8:
    print("{} + {} = {}".format(p, p, (p+p)))
    p += 1
```

```
1 + 1 = 2
2 + 2 = 4
3 + 3 = 6
4 + 4 = 8
5 + 5 = 10
6 + 6 = 12
7 + 7 = 14
```

In [65]:

```
a = 1
while a < 8:
    print("%f * %f = %f" % (a, a, (a* a)))
    a += 1
```

```
1.000000 * 1.000000 = 1.000000
2.000000 * 2.000000 = 4.000000
3.000000 * 3.000000 = 9.000000
4.000000 * 4.000000 = 16.000000
5.000000 * 5.000000 = 25.000000
6.000000 * 6.000000 = 36.000000
7.000000 * 7.000000 = 49.000000
```

In [67]:

```
p = 1 ; o = 1
while p < 15 and o < 15:
    print("%i / %f = %i" % (p, o, (p/o)))
```

```
    p += 1; o += 1
```

```
1 / 1.000000 = 1
2 / 2.000000 = 1
3 / 3.000000 = 1
4 / 4.000000 = 1
5 / 5.000000 = 1
6 / 6.000000 = 1
7 / 7.000000 = 1
8 / 8.000000 = 1
9 / 9.000000 = 1
10 / 10.000000 = 1
11 / 11.000000 = 1
12 / 12.000000 = 1
13 / 13.000000 = 1
14 / 14.000000 = 1
```

In [69]:

```
num = 5 ; cake = 10
while num < 12:
    while cake < 17:
        print("{0} + {1} + {0} = {2}".format(num, cake, num, num+cake))
        cake += 1 ; num += 1
```

```
5 + 10 + 5 = 5
6 + 11 + 6 = 6
7 + 12 + 7 = 7
8 + 13 + 8 = 8
9 + 14 + 9 = 9
10 + 15 + 10 = 10
11 + 16 + 11 = 11
```

In [73]:

```
lang = ["Python", "Java", "JavaScript", "R", "VBA", "C#", "C++", "Julia",
        "HTML", "CSS", "C", "Go"]
x = 1
while x < len(lang):
    print(lang[x]) ; x+= 2
```

```
Java
R
C#
Julia
CSS
Go
```

In [76]:

```
g = 0
while g < 5:
    print(g) ;g += 1
```

```
0
1
2
3
4
```

In [75]:

```
for d in range(5):
    print(d)
```

```
0
1
2
3
4
```

In [77]:

```
a = 1 ; b = 1
while a < 5:
    while b < 5:
        print("{} + {} = {}".format(a, a, b+a))
        a += 1 ; b += 1
```

```
1 + 1 = 2
2 + 2 = 4
3 + 3 = 6
4 + 4 = 8
```

In [78]:

```
for j  in range(1, 5):
    for k in range(1, 5):
        print("{} + {} = {}".format(j, k, j+k))
```

```
1 + 1 = 2
1 + 2 = 3
1 + 3 = 4
1 + 4 = 5
2 + 1 = 3
2 + 2 = 4
2 + 3 = 5
2 + 4 = 6
3 + 1 = 4
3 + 2 = 5
3 + 3 = 6
3 + 4 = 7
4 + 1 = 5
4 + 2 = 6
4 + 3 = 7
4 + 4 = 8
```

In [79]:

```
for t in range(1, 5):
    print("{} + {} = {}".format(t, t, t+t))
```

```
1 + 1 = 2
2 + 2 = 4
3 + 3 = 6
4 + 4 = 8
```

In [81]:

```
num1 = 0
while num1 < 110:
    print("while num1 = ", num1)
    for g in [50, 80, 110]:
```

```
        res = num1 + g
        print(res)
    num1 += 20
```

```
while num1 =  0
50
80
110
while num1 =  20
70
100
130
while num1 =  40
90
120
150
while num1 =  60
110
140
170
while num1 =  80
130
160
190
while num1 =  100
150
180
210
```