

Lesson 3

Loops

The nice thing about computers is that we can make them do things as many times as we want. This method of repeating a task is called a 'loop'. We can either loop for a specified number of times (the 'for' loop) or until a condition is met. Below are some samples you can run.

loop_samples.py

```
#Commands
```

```
# Animal classify
```

```
animals = ['frog','penguin','tarantula','cat','tiger','tarantula']
```

```
count = 0
```

```
for a in animals:
```

```
    # print( a ) # Explain it
```

```
    if (a == 'tarantula'):
```

```
        print('Find it!!')
```

```
        count = count + 1
```

```
    else:
```

```
        print('Aww...')
```

```
print('tarantula is',count,'times on the list')
```

```
# Countdown
```

```
for n in range(10,0,-1):
```

```
    print( n, "!" )
```

```
    if (n == 1)
```

```
        print( "BLASTOFF!" )
```

```
# find how many consecutive numbers summed to exceed given number
```

```
maxsum = 5
```

```
sum_ = 0
```

```
n = 1
```

```
while (sum <= maxsum):
```

```
sum = sum + n  
n = n+1 #counter
```

```
print( "max consecutive numbers summed that exceed %d : %d" % (maxsum, n-1) )
```

Nested Loops and Decision Structures

The decision structures and loops we introduced are quite simple, but things can get more complex when we start dealing with multidimensional arrays and decisions. Let's say for example you have an array of daily temperatures, arranged with each week as a separate row, with each day of the week as a separate column. For the month of February you want to find how many days have temperatures that fall within the comfort zone of 65 F to 75 F. Let's look at the data and a function to analyze it.

temp.py

```
#Library
```

```
import numpy as np
import temp_module as tm
```

```
#Commands
```

```
T_array = np.array([[59, 70, 63, 61, 60, 62, 65],[66, 67, 63, 61, 60, 62, 64],[65, 68, 70,
71, 69, 65, 63],[61, 58, 54, 55, 57, 59, 61]])
```

```
countT = tm.temp_analyze(T_array,65,75)
print(countT)
```

temp_module.py

```
#Library
```

```
Import numpy as np
```

```
#Commands
```

```
def temp_analyze(T_array, lowT, highT):
    niceTcount = 0 # first things first, we have to define a variable
    dims = np.shape(T_array)

    for row in range(dims[0]): #dim[0] extract number of rows
        for col in range(dims[1]): # note the nested loop
            if (T_array[row,col] >= lowT):
                if (T_array[row,col] <= highT): # nested if!
                    niceTcount = niceTcount + 1; #counter
    return(niceTcount)
```

Of course, this could be done much easier with the following two commands:

```
condition_temp = T_array[(T_array>=65) & (T_array<=75)]  
print(np.size(condition_temp))
```

But if you want to do something complicated like count by days of the week, you may be back to using these nested loops and decision structures.
