Lecture 8 - Loops / For Loops / Range

To deal with data that is not individual data we have 2 different tools. The first is functions that call themselves - this is recursion.

The second is Loops. Loops are the more common (ex1.py).

```
for i in range(3):
    print ( "i = {}".format(i) )
```

So… This uses the “range” operator. range encodes a set of values that can be used in a loop.

```
print ( range(5) )
```

When it is specified like this it is the same as `range(0, 5)`.

A range is a start value that increments up to but not including the end value. So `range(0, 5)` will have values 0, 1, 2, 3, 4.

This is useful whenever we have a loop to control how many times the loop will go around.

Let’s walk through a loop (ex3.py):

```
ll = [ "dog", "cat", "goldfish", "parakeet" ]
print ( "list = {}".format(ll) )
print ( "length of list = {}".format(len(ll)) )
for i in range ( len(ll) ):
    print ( "i = {} list[{}] = {}".
            format ( i, i, ll[i] ) )
print ( "done" )
```

So we can use this to search a list (ex4.py):

```
ll = [ "dog", "cat", "goldfish", "parakeet" ]
for i in range ( len(ll) ):
    if ll[i] == "cat":
        print ( "found 'cat' in list! list[{}] = {}".
                format ( i, ll[i] ) )
```

A better way to search the values is to use the for loop with the value instead of an index (ex5.py).

```
ll = [ "dog", "cat", "goldfish", "parakeet" ]
for v in ll:
    if v == "cat":
        print ( "found 'cat' in list!  {}".format ( v ) )
```

Python has a 2nd kind of loop, a while loop (ex6.py).

```
i = 0
while ( i < 3 ):
    print ( "i = {}".format ( i ) )
    i = i + 1
```

We can change this to print 1, 2, 3 (ex7.py):

```
i = 0
while ( i < 3 ):
    i = i + 1
    print ( "i = {}".format ( i ) )
```

We can go thru our pets (ex8.py):

```
ll = [ "dog", "cat", "goldfish", "parakeet" ]
i = 0
while ( i < len(ll) ):
    print ( "i = {} list[{}] = {}".format ( i, i, ll[i] ) )
    i = i + 1
```

But.... We can also break it (ex9.py).

```
i = 0
while ( i < 3 ):
    print ( "i = {}".format ( i ) )
    # bad loop!
```

Copyright