range()

In this short lecture we will be discussing the range function. We haven't developed a very deep level of knowledge of functions yet, but we can understand the basics of this simple (but extremely useful!) function.

range() allows us to create a list of numbers ranging from a starting point *up to* an ending point. We can also specify step size. Lets walk through a few examples:

```
In [7]: range(0,10)

Out[7]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [8]: x =range(0,10)
        type(x)

Out[8]: list
```

```
In [3]: start = 0 #Default
        stop = 20
        x = range(start,stop)
```

```
In [4]: x

Out[4]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

Great! Notice how it went *up to* 20, but doesn't actually produce 20. Just like in indexing. What about step size? We can specify that as a third argument:

```
In [5]: x = range(start,stop,2)
        #Show
        x

Out[5]: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

Awesome! Well thats it...or is it?

Python 3 Alert!

You might have been wondering, what happens if I want to use a huge range of numbers? Can my computer store that all in memory?

Great thinking! This is a dilemma that can be solve with the use of a generator. For a simplified explanation: A generator allows the generation of generated objects that are provided at that instance but does not store every instance generated into memory.

This means a generator would not create a list to generate like range() does, but instead provide a one time generation of the numbers in that range. Python 2 has a built-in range generator called xrange(). It is recommended to use xrange() for **for** loops in Python 2.

The good news is in Python 3, range() behaves as a generator and you don't need to worry about it. Let's see a quick example with xrange()

```
In [9]: for num in range(10):
            print num
```

```
0
1
2
3
4
5
6
7
8
9
```

```
In [10]: for num in xrange(10):
             print num
```

```
0
1
2
3
4
5
6
7
8
9
```

So the main takeaway here is for Python 2, if you are using range() in a way that you don't need to save the results in a list, use xrange() instead. For Python 3, use range() in any instance.

You should now have a good understanding of how to use range() in either version of Python.