SECTION 5; PYTHON STATEMENTS, 1 hour 15 mins, 7 Parts

- **5/7 List Comprehensions in Python**
    - **-> list comprehensions**
        - a method of creating lists in Python
        - -> e.g an alternative to a append statement

    - **-> in the .ipynb file**
        - mystring = 'hello' <- we have a string
        - mylist = []
        - for letter in mystring: <- **iterate through the string and for each letter it's adding that letter to the empty list (as an element of the list)**
            - mylist.append(letter)
        - mylist <- this returns ['h,'e',.....,'o'] <- then printing out the list once done

        - **-> you can do the same thing in a string comprehension**
            - **mylist =[letter for letter in mystring]**
                - **and mystring = 'hello'**
                - **-> it's element for element in the name of the variable which is storing the string**
                    - **-> to make the characters of the string into a**
                    - **-> it's a flattened out for loop**

        - mylist = [x for x in 'word']
        - mylist = ['w','o','r','d']
        - **-> more "efficient" -> either through computational time or in lines of code**
        - -> mylist = [x for x in 'word']
            - -> it returns ['w','o','r','d']
            - -> x is a temporary variable name

        - **-> you can do this with the range parameter**
            - **-> mylist = [num for num in range(0,11)]**
            - **-> it's generating an array from 0 to 11**
            - **-> you can perform operations on the first variable name -> e.g num**2**
            - -> it squares all of the numbers
            - -> flatten out the for loop
            - -> you can also do num**2 instead of the first num

            - **-> an alternative approach to this**
                - **mylist = [x for x in range(0,11) if x%2==0]**
                    - **iterate through numbers in the range from 0-11**
                    - **and do so if x/2 has no remainder (only the even numbers)**
                    - **-> it's printing out the even numbers**
                    - **-> you can also do x%2!==0 for even numbers**

            - **-> temperature example**
                - Celsius = [0,4,16]
                - Farenheight = [(((9/5)*temp +32) for temp in celsius]
                    - it's a flattened out for loop

            - **-> list comprehensions**
                - these are compact -> but it's important to make them readable

            - **-> they are harder to read when you come back to them**
            - **-> if statements in list comprehensions**
                - results = [x if x%2==0 else 'ODD' for x in range(0,11)]
                - -> this returns the odd numbers in the range

- **-> nested loops**
    - mylist = []
    - for x in [2,4,6]:
        - for y in [100,200,300]:
            - mylist.append(x*y)
            - -> this returns a list from 200-1800
            - -> increasing in intervals of 2
            - -> we're iterating through an array in an array
    - -> the same thing as a list comprehension is -> mylist = [x*y for x in [2,4,6] for y in [1,10,1000]]