List Basics

In this course we will study four data structures in Python: `lists`, `tuples`, `sets`, and `dictionaries`. They are called sequences because they hold sequences of elements. These data structures are also called `iterables`, because one can loop through them one value at a time. Generally speaking, `iterables` are anything that can be looped over (i.e. you can loop over a string or file). We also use the term `containers` to refer to them.

This notebook introduces the first of the data structures discussed above - `lists`. Lists are a sequence of elements enclosed in square brackets and separated by commas. There is a lot of similarity between lists and strings but also some important differences.

Below is a brief comparison between Lists and Strings.

Differences between Strings and Lists:

1. Strings are immutable while lists are mutable. That means individual elements of a list can be changed as needed.
2. Strings can only contain a sequence of characters while list are a sequence of any combination of numbers, strings, lists, and other data structures.

Similarities between String and Lists:

1. Like strings, individual elements of a list can be accessed using the appropriate index. This is because lists, like strings are an ordered collection.
2. The len() function can also be applied on lists to find the number of elements in a list.
3. List slicing works exactly like string slicing and can be used to extract portions of a list.
4. List traversal can be done using the for construct just like we did in the case of string traversal.

In [3]:

```
'''
    This example displays a few list constants.
'''
from random import random
print([15,22,32,25,11])
print([random(), random(), random(), random(), random()])
print(['a', 'e', 'i', 'o', 'u'])
```

```
[15, 22, 32, 25, 11]
[0.27921865695678905, 0.8753023900452765, 0.9122213291953117, 0.6133781761526196,
0.3484301123855277]
['a', 'e', 'i', 'o', 'u']
```

In [4]:

```
'''
    List variables can be created by assigning a list constant to a variable name.
    Note that a list can contain elements of different types including other lists.
'''
test_lst = ['5', 4, 3.3, 2, '1', True, [8,9], (5, 9)]
print(test_lst)
```

```
['5', 4, 3.3, 2, '1', True, [8, 9], (5, 9)]
```

In [5]:

```
'''
    List elements can be any valid Python element; int, float, strings, other lists, expressions,
etc.
'''
def fn(n):
    return n**2
x = 5
y = 8
z = [x,y]
u = [x,y,z,x+12, fn(5)]
print(z)
print(z)
print(u)
```

```
[5, 8]
[5, 8, [5, 8], 17, 25]
```