**Brief Miscellanea**

To stop the print statement from adding a newline using end, one needs to be using python3.

print ("Stripped of newline", end='')  
print ("Seriously")

There is no simple built-in string function that gives the index for all matches to the ‘find’ method argument, but regular expression functions in the re module do enable this. These are to be covered later.

Python allows “string multiplication”

print ('a' \* 20)

**Introduction to Lists**

One data structure that can hold many pieces of information is a *list*. Each item in the list is an *element*. A list is indexed with an integer beginning with zero. There is a start [0] and an end e.g. [4] and a length e.g. (5). Lists can have one element or no elements.

cards= ['7D', '2S', '10H', '3S', 'AC']

print ("first is " + cards[0])

The integer in the bracket is the array index.

dna=['A','T','C']

We can see the contents of a list by printing it.

print(dna)

print ("DNA is " + dna[0] + " " + dna[1] + " " + dna[2])

print ("DNA is " + " ".join(dna))

Assuming the list is a list of strings then one can use the *join* method. The *join ()* method is a string method. Above, the method acts on a string, here a whitespace, and list of things to join is the method argument.

We can concatenate lists with the + symbol, as we did with strings.

nonsense=cards + dna

**Working with list elements**

We can replace elements within a list by using the index of that item.

dna[0]='G'

print ("DNA is " + " ".join(dna))

We can replace multiple elements at one time.

(dna[0], dna[2])=(dna[2], dna[0])

Assigning a negative number to the index counts from the tail end. [-1] is the last element.

print(dna[-1])

We can give a start and stop position, separated by a colon, to specify a range of elements.

print (cards[2:3])

This command uses the same nomenclature we used for substrings (e.g. string[start:end])

When we took the substrings from the string, we are treating the variable as a list even though it was not.

Instead of getting the list element with the index, we can get the index with the list element using the index() method on the list.

cards.index("7D")

#cards.index will have value of 0.

To add elements to the end of the list, we can use the *append()* method. Note that append actually changes the cards list.

cards.append("7H")

nt='G'

dna.append(nt)

print (" ".join(cards))

print (" ".join(dna))

print ("Last DNA is " + dna[-1])

The extend() method is like append() but takes a list. Note that this method also changes the list.

cards.extend(["8C","9H","10D"])

print(cards)

Two more methods that change the variable are reverse() and sort().

cards.sort()

print(cards)

By default, python sorts strings in alphabetical order and numbers in ascending order.

It can be useful to know the number of elements a list contains. We can get the length of strings using the *len()* function.

print(str(len(dna))

Finally, we can split a string into an array using the split() method.

card\_string="7D,2S,10H,3S,AC"

cards2=card\_string.split(",")

print ("The first card is " + cards2[0])