Lesson 3

Lists

For those of you that have done some coding before may be familiar with an **array**. Well in Python, a **list** is very similar to an array in other languages. If you have no idea what an array or list is, that's okay!

A list is just what is says, its a list, usually of a particular variable type (ints, floats, strings, etc.). Lists save us time by allowing us to use one list variable, rather than have a seperate variable for each item in the list. Let's say that we have a list that stores peoples' names and another list that stores their age. The lists would look like this:

```python
names = ["John", "Mary", "Adam", "Laura", "Mark"] # list containing names (strings)
ages = [26, 47, 9, 30, 62] # list containing ages (ints)
```

List Item Access

Similar to what we have done before in the previous lesson, we can access and use particular items within our list.

```python
print("Name: " + names[0] + " Age: " + ages[0]) # prints "Name: John Age: 26"
```

Note: Remember that a string is just a list of characters (letters). Accessing a list element is just like accessing a character in a string.

Tasks:
1. Print out the remaining elements of the list, just like we did in the above example!

Consequences Project

Write code to create a Python version of the party game 'Consequences', where people make up stories by throwing random characters, places and events together.

- Create a list of several male characters and another list of several female characters. For example:

```
man = ["a fire fighter", "a cowboy", "your father", "a clown"]

woman = ["a doctor", "a unicorn", "an accountant", "a soldier"]
```

- Create lists for any other parts in your story. Some ideas for what you could include in your story could be (feel free to add more yourself):
  - place (list of places your story could take place)
  - heWore (list of things that your male character could wear in the story)
  - sheWore (list of things that your female character could wear in the story)
  - heSays (list of things that your male character could say in the story)
  - sheSays (list of things that your female character could say in the story)
  - Consequence (list of possible consequences for your characters)

- Import the random library to choose a random male character, female character and location.

```
import random
```

- Start the story by choosing random characters and places, and put them into a sentence.

```
print(random.choice(woman) + " met " + random.choice(man) + random.choice(place))
```

Note: random.choice(myList) picks a random item from your list.

- Expand your story by doing the same with the other lists in your code:

```
print("She was wearing " + random.choice(sheWore))
print("He was wearing " + random.choice(heWore))
print("She said " + random.choice(womanSays))
print("He said " + random.choice(manSays))
print("The consequence was " + random.choice(consequence))
```

Extra Tasks

Insert/Remove List Element

After we initially declare our list, we can both insert and remove list elements as we see fit. We can use the following methods to accomplish this:

append()   -   Inserts element to the end of list
insert()   -   Inserts element to a given index position in list
remove()   -   Removes a given element from the list

See the examples below and try them out.

```python
print(names) # prints "["John", "Mary", "Adam", "Laura", "Mark"]"

names.append("Julia") # insert "Julia" to end of list

print(names) # prints "["John", "Mary", "Adam", "Laura", "Mark", "Julia"]"

names.insert(2, "Patrick") # insert "Patrick" to the second index of list

print(names) # prints "["John", "Mary", "Patrick", "Adam", "Laura", "Mark",
"Julia"]"

names.remove("John") # remove "John" from list

print(names) # prints "["Mary", "Patrick", "Adam", "Laura", "Mark", "Julia"]"

names.remove(names[0]) # remove first element ("Mary") from list

print(names) # prints "["Adam", "Laura", "Mark", "Julia"]"
```

Each time we insert or remove elements from our list, our "names" list changes.

Tasks:
1. **Add** a new name to the **end** of the list and print it out.
2. **Add** a new name to the **middle** of the list and print it out.
3. **Remove** a **random element** from the list and print it out.

    Hint: use the random.randint() method that we used for the Number Guessing Game in Lesson 1 to generate a random number in the square brackets ("0" is currently used in the example above).

Additional Information

List Length

Just like when we obtained the length of a string, the length of a list can be obtained using the **len()** method.

Sort List

Usually, when we declare lists they may not be in the order we would like them to be. We can change this by using the **sort()** method. This method gives the option of sorting a list of Strings in alphabetical or reverse alphabetical order and a list of numbers in ascending or descending order.

```python
names.sort() # sort names in alphabetical order

print(names) # prints "["Adam", "John", "Laura", "Mark", "Mary"]"

ages.sort() # sort ages in ascending order

print(ages) # prints "[9, 26, 30, 47, 62]"

names.sort(reverse=True) # sort names in reverse alphabetical order

print(names) # prints "["Mary", "Mark", "Laura", "John", "Adam"]"

ages.sort(reverse=True) # sort ages in descending order

print(ages) # prints "[62, 47, 30, 26, 9]"
```

This method of sorting the elements in a list is just another tool we have under our belts to make dealing with lists that bit more manageable!