CompSci 101 Lab 6
More on Lists

Slicing Lists

Inbuilt Functions

Updating Elements in a List

List Methods

Slicing Lists

[start : stop : step]

- Lists can be sliced to extract a smaller list

Examples:

```
numbers_list = [4, 7, 2, 3, 9, 8, 5]
slice = numbers_list[0:6:2]
print(slice)
```

```
[4, 2, 9]
```

```
another_slice = numbers_list[-2:]
print(another_slice)
```

```
[8, 5]
```

Inbuilt functions that work with lists

- len(a_list)
  - returns the number of elements in a list
- min(a_list)
  - returns the minimum element in a list
- max(a_list)
  - returns the maximum element in a list
- sum(a_list)
  - returns the sum of the elements in a list
  - (for lists that contain numbers only).

Iterating through a list using a for … in … loop

Example:

```
numbers_list = [6, 3, 0, 8]
for number in numbers_list:
    print(number)
```

would print the following:

```
6
3
0
8
```

Changing Elements in a List

Iterating through a list using a "for .. in range" loop

- If we wish to loop through a list ***changing the value*** of the list elements we cannot use the basic "for … in" loop.
- We need to access each element using the **position** (i.e. the **index**) of the element. To do this, we need to use a for .. in range() loop.

Example:

```
for index in range(0, len(marks_list)):
    marks_list[index] = round(marks_list[index])
```

  - This code would round each mark in the marks_list to the nearest whole number.

index() method

- **index**(*value*)
  - Returns the index (i.e. the position) of the first element in the list which is equal to the parameter *value*
  - An error occurs if the parameter value does not exist
  - So always check first to make sure the value exists

Example:

```
colours_list = ["purple", "pink", "black"]
if "pink" in colours_list:
    position = colours_list.index("pink")
    print("pink is in Position, position)
```

The pop() method

- Removes and returns the element at the position specified in the parameter.
- If no parameter is specified, pop() removes the last element in the list.

Example:

```
colours_list = ["purple", "pink", "black", "blue"]
colours_list.pop(2)
last_colour = colours_list.pop()
first_colour = colours_list.pop(0)
```

The insert() method

- insert(*index, value)*
  - inserts the parameter value into the specified position

Example:

```
colours_list = ["purple", "pink", "black"]
colours_list.insert(1, "blue")
print(colours_list)
```

would print:

```
["purple", "blue", "pink", "black"]
```

The sort() method

- The sort() method sorts a list into ascending order

Example:

```
marks_list = [75, 2, 9, 81, 5, 22]
marks_list.sort()
print(marks_list)
```

Would print:

[2, 5, 9, 22, 75, 81]

The reverse() method

- The reverse() method reverses the order of the elements in the list

Example:

```
marks_list = [75, 2, 9, 81, 5, 22]
marks_list.reverse()
print(marks_list)
```

Would print:

[22, 5, 81, 9, 2, 75]

The list() function

- The list function will create a list from any sequence (e.g. a string, a tuple, a range of numbers, another list)

Example 1:

```
word = "tree"
letters_list = list(word)
print(letters_list)
```

Would print:

['t', 'r', 'e', 'e']

The list() function (continued)

Example 2:

```
numbers_list = list(range(9, 19, 3))
print(numbers_list)
```

Would print:

[9, 12, 15, 18]

The list() function (continued)

Example 3:

```
numbers_list = [9, 12, 15, 18]
copy_of_numbers_list = list(numbers_list)
print(copy_of_numbers_list)
```

Produces a copy of the numbers_list:

[9, 12, 15, 18]

Next week

Lab 7 will be on tuples and file processing

See you all next week ☺