Working with Lists - Lab

Introduction

Now that we have a sense of how to read from a list and alter a list in Python, let's put this knowledge to use.

Objectives

You will be able to:

- Use indexing to access elements in a list
- Apply list methods to make changes to a list
- Change elements of a list

Instructions

In the previous lesson, we had a list of top travel cities.

```
top_travel_cities = ['Solta', 'Greenville', 'Buenos Aires', 'Los Cabos', 'Walla Wall
```

> Remember to press shift+enter to run each gray block of code (including the one above). Otherwise, the variables will not be defined.

In this lab we will work with a list of associated countries corresponding to each of the top travel cities.

```
countries = ['Croatia',
 'USA',
 'Argentina',
 'Mexico',
 'USA',
 'Morocco',
 'New Mexico',
 'Finland',
 'Argentina',
 'Italy',
 'Canada',
 'South Korea']
```

> Run the code in the cell above by pressing shift + enter.

The list of countries associated with each city has been assigned to the variable `countries`. Now we will work with this list.

Accessing elements from lists

First, set the variable `italy` to be equal to the third to last element from `countries`.

> **Note:** If you see an **error** stating that `countries` is undefined, it means you must press shift+enter in the second gray box where `countries` variable is assigned.

```
italy = countries[-3] # 'Italy'
italy
```

```
'Italy'
```

> We assigned the varible `italy` equal to `None`, but you should change the word `None` to code that uses the `countries` list to assign `italy` to `'Italy'`. We wrote the variable `italy` a second time, so that you can see what it contains when you run the code block. Currently, nothing is displayed below as it equals `None`, but when it's correct it will match the string which is commented out, `'Italy'`.

```
italy # 'Italy'
```

```
'Italy'
```

Now access the fourth element and set it equal to the variable `mexico`.

```
mexico = countries[3]
mexico
```

```
'Mexico'
```

Notice that the second through fifth elements are all in a row and all in the Western Hemisphere. Assign that subset of elements to a variable called `kindof_neighbors`.

```
kindof_neighbors = countries[1:5]
kindof_neighbors
```

```
['USA', 'Argentina', 'Mexico', 'USA']
```

Changing Elements

Ok, now let's add a couple of countries onto this list. At the end of the list, add the country 'Malta'.

```
countries.append('Malta') # add code here
```

Then add the country 'Thailand'.

```
countries.append('Thailand') # add code here # add code here
```

Now your list of countries should look like the following.

```
countries
# ['Croatia', 'USA', 'Argentina', 'Mexico', 'USA', 'Morocco', 'New Mexico', 'Finland
# 'Argentina', 'Italy',  'Canada', 'South Korea',  'Malta',  'Thailand']
```

```
['Croatia',
 'USA',
 'Argentina',
 'Mexico',
 'USA',
 'Morocco',
 'New Mexico',
 'Finland',
 'Argentina',
 'Italy',
 'Canada',
 'South Korea',
 'Malta',
 'Thailand']
```

You may have noticed that "New Mexico" is included in our list of countries. That doesn't seem right. Let's change 'New Mexico' to 'USA'.

```
countries[6] = "USA" # add code here
```

```
countries
# ['Croatia', 'USA', 'Argentina', 'Mexico', 'USA', 'Morocco', 'USA', 'Finland',
# 'Argentina', 'Italy',  'Canada', 'South Korea',  'Malta',  'Thailand']
```

```
['Croatia',
 'USA',
 'Argentina',
 'Mexico',
 'USA',
 'Morocco',
 'USA',
 'Finland',
 'Argentina',
 'Italy',
```

```
 'Canada',
 'South Korea',
 'Malta',
 'Thailand']
```

Finally, let's remove Thailand from the list. No good reason, we're acting on whimsy.

```
countries.pop() # 'Thailand'
```

```
'Thailand'
```

```
print(countries)
```

```
['Croatia', 'USA', 'Argentina', 'Mexico', 'USA', 'Morocco', 'USA', 'Finland',
'Argentina', 'Italy', 'Canada', 'South Korea', 'Malta']
```

Exploring Lists with Methods

Ok, now we notice that some countries are mentioned more than once. Let's see how many repeat countries are on this list.

First, use the `set` and `list` functions to return a unique list of countries. Set this list equal to the variable `unique_countries`.

```
unique_countries = list(set(countries))
```

```
unique_countries # ['Canada', 'Italy', 'USA', 'Mexico', 'Finland',
#'Malta', 'Morocco', 'Croatia', 'Argentina', 'South Korea']
```

```
['South Korea',
 'Malta',
 'USA',
 'Canada',
 'Croatia',
 'Morocco',
 'Argentina',
 'Finland',
```

```
 'Mexico',
 'Italy']
```

Now the number of repeat countries should be the number of countries minus the number of unique countries. So use the `len` function on both `unique_countries` and `countries` to calculate this and assign the result to the variable `num_of_repeats`.

```
num_of_repeats = len(countries) - len(unique_countries)
num_of_repeats # 3
```

```
3
```

Summary

In this lesson, we practiced working with lists in Python. We saw how to index lists to select specific elements, how to use list methods to change lists, and how to add and remove elements from a list. Finally, we saw how to use a set to calculate the number of unique elements in the list.

Releases

No releases published

Packages

No packages published

Contributors 6

Languages

- **Jupyter Notebook** 89.0%
- **Python** 11.0%