Select and Drop Entries in Series

In this lesson, an explanation on how to select and drop entries in a series is provided.

We'll cover the following

Selecting entries

There are many different ways to select entries from a `Series` object. This operation will be used throughout the course to manipulate data, so some of the selection techniques will be discussed here.

Selection in `Series`

Elements in a `Series` object can be selected in a number of ways, but for this course, we will use the following techniques.

The following methods in the code snippet select elements based on the *index* name or *index* number.

```
import numpy as np
import pandas as pd

srs = pd.Series(np.arange(0, 6, 1), index = ['ind0', 'ind1', 'ind2', 'ind3', 'ind4', 'ind5'])
srs.index.name = "Index"
print("The original Series:\n", srs)

print("\nSeries element at index ind3:")
print(srs['ind3']) # Fetch element at index named ind3

print("\nSeries element at index 3:")
print(srs[3]) # Fetch element at index 3

print("\nSeries elements at multiple indexes:\n")
print(srs[['ind1', 'ind4']]) # Fetch elements at multiple indexes
```

In the above code example, elements from the `Series` are selected in 3 different ways.

- On **line 9**, the element is selected based on the index name.
- On **line 12**, the element is selected based on the index number. Keep in mind that index numbers start from `0`.
- On **line 15**, multiple elements are selected from the `Series` by selecting multiple *index* names inside the `[]`.

The following methods in the code snippet select a range of elements.

```
import numpy as np
import pandas as pd

srs = pd.Series(np.arange(0, 6, 1), index = ['ind0', 'ind1', 'ind2', 'ind3', 'ind4', 'ind5'])
srs.index.name = "Index"
print("The original Series:\n", srs)

print("\nSeries elements after and including index 4:\n", srs[4: ]) # Fetch elements in a range

print("\nAll Series elements greater than equal to 3:\n", srs[srs >= 3]) # Fetch elements with a c
```

In the above code example, elements from the `Series` are selected in 2 different ways.

- On **line 8**, multiple elements from the `Series` are selected in a given range.
- On **line 10**, multiple elements are selected based on a certain condition. The condition would be declared inside the `[]`. In this case, all elements that are greater or equal to 3 are selected.

Dropping entries

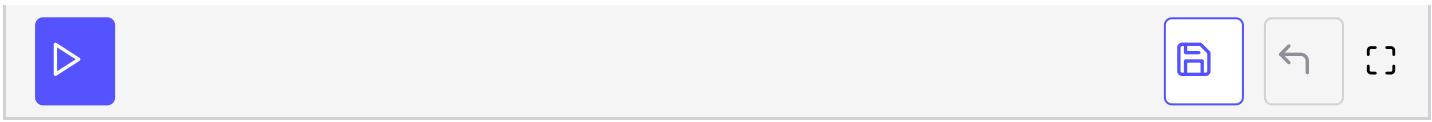
Dropping an unwanted *index* from the `Series` object is also an important function. If the `drop(index_name)` function is called with a given index on a `Series` object, the desired index name is deleted from the `Series`.

```
import numpy as np
import pandas as pd
```

```
srs = pd.Series(np.arange(0, 6, 1), index = ['ind0', 'ind1', 'ind2', 'ind3', 'ind4', 'ind5'])
srs.index.name = "Index"
print("The original Series:\n", srs)

srs = srs.drop('ind2') # drop index named ind2

print("The New Series:\n", srs)
```



It can be seen from the output that the `ind2` index is dropped, and the new `Series` object does not have this *index* name in it. Also, an index can only be dropped by specifying the index name and not the number. So, `srs.drop(srs[2])` does not work.

In the next lesson, selecting and dropping entries in a `DataFrame` are discussed.