Lists: Takeaways

Syntax

- Create a list by surrounding a collection of items with square brackets (`[]`) and separate the items with commas:

```
first_row = [6.1, True, False, True, "Scikit-learn", 124267]
```

- Read data from a list by using its index:

```
first_row[0]   # indexing the first item in the list first_row
first_row[0:5] # Use multiple, consecutive indices to get a slice of first_row
first_row[:5]  # Omitting the start index is a shortcut for starting from the beginning of a list
first_row[-1]  # Using negative indices to reference elements at the end of a list
first_row[-3:] # Omitting the stop index is a shortcut for including everything to the end of a list
```

- Add data to the end of a list with the `append()` function:

```
monthly_salary = first_row / 12
first_row.append(monthly_salary)
```

- Update data from a list by indexing a location and reassigning its value:

```
first_row[6] = 130000
first_row[1:4] = [False, False, False]
```

- Delete data from a list or variables in general with the `del` statement:

```
del first_row[1] # delete a single item in first_row
del first_row    # delete the first_row variable
```

Concepts

- Lists are ordered collections of data. They are mutable because we can add, update and delete data from a list. They are ordered because there is a notion of a "first item" and "second item" and so on.
- Python is a zero-indexed programming language, meaning that `0` indicates the first position in a list instead of `1` .
- One way we can represent a large dataset in a list is through a list of lists. The outer list represents the collection of many different rows of data. The inner lists represent the individual rows of data.

Resources

- More On Adding Data To Lists
- More On Removing Data From Lists