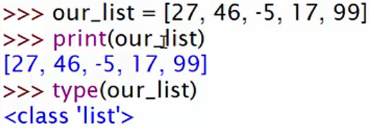
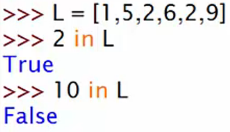
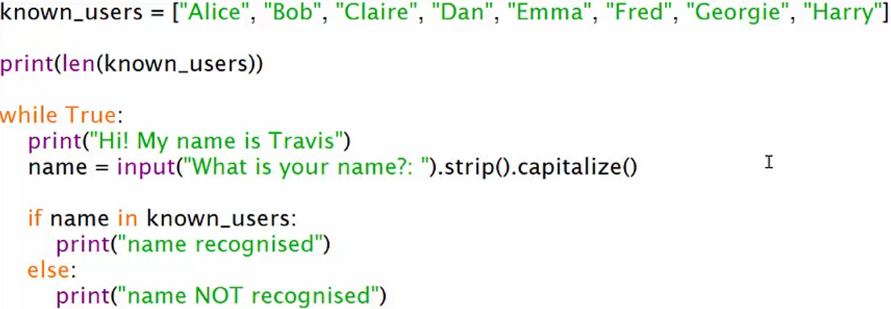
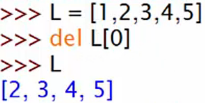
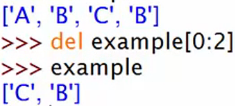
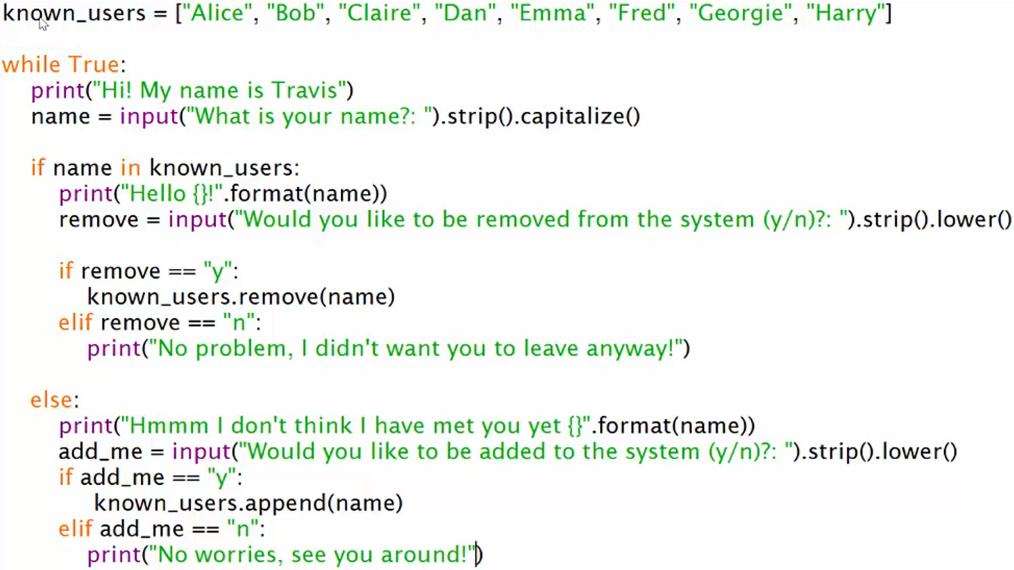
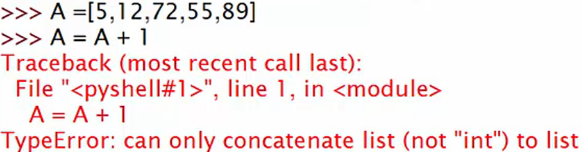
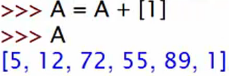
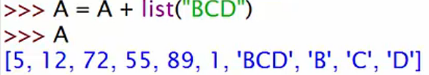
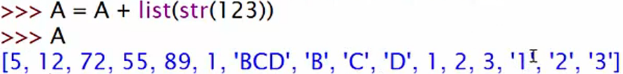
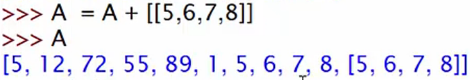
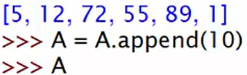
**Section Overview**  
\* **Data Structures in Python are pretty much a way to combine together pieces of information into 1 nice easy to use place**.  
**List  
Tuple  
Dictionary**

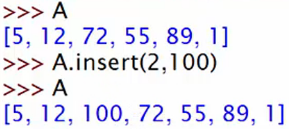
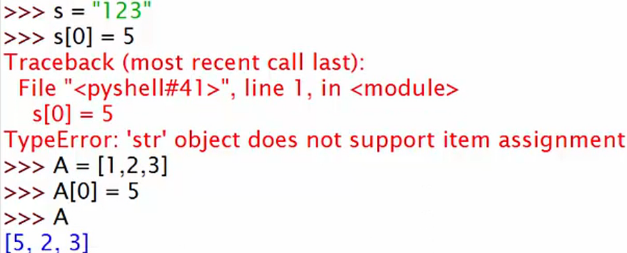
**What are Lists?**  
\* **The most common of all Python Data Structures**.  
**list** => **[]   
=> Iterable Datatype, each element has its own index.**  
\* **Python Lists unlike those in other languages, can have multiple datatypes inside them**.  
\* **For taking out more than 1 element, you need a slice**.  
**[start:end:step] => takes a copy, doesn’t modify the original list.**  
\* **Lists can have other lists inside them**.  
**[row][column]**

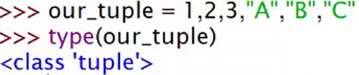
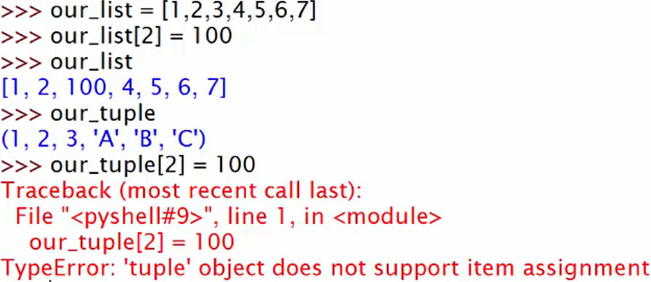
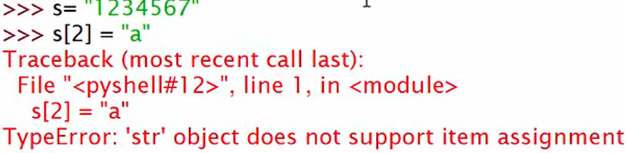
**PROJECT 4 - Travis the Ridiculous Security System - Part 1**  
**len()**  
**in => in condition**  
  


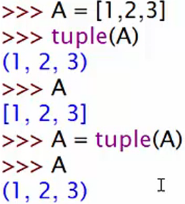
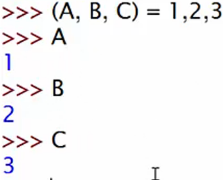
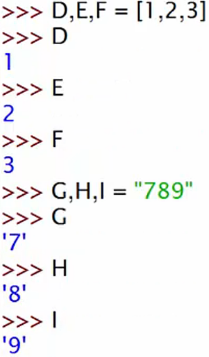
**PROJECT 4 - Travis the Ridiculous Security System - Part 2**  
**list remove() => removes the 1st instance**  
**del()** **list => removes the element at index**  
  


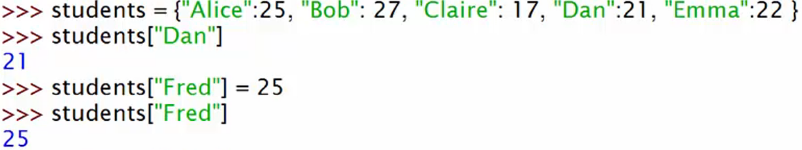
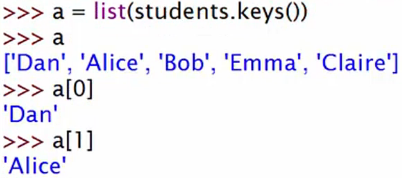
**PROJECT 4 - Travis the Ridiculous Security System - Part 3**  
**list** **append() => appends to the end of the list**  


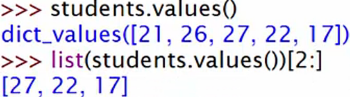
**More ways to add items to lists**  
**list** **append()**  
  
\* **List Concatenation**  
  
**list() => converts an ITERABLE such as a string into an array of elements, list(“BCD”) => [‘B’, ‘C’, ‘D’]**  
  
\* You can’t get this functionality for integers because integers aren’t iterable.  
\* We can convert the integers to strings of course:  
  
\* **We can add extra [] to insert a list**.  
  
\* **The append() method returns an empty value**.  
  
=> **The A is now empty**.  
**NoneType** => **blank space**.  

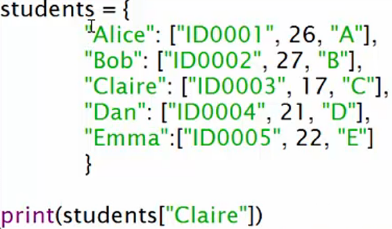
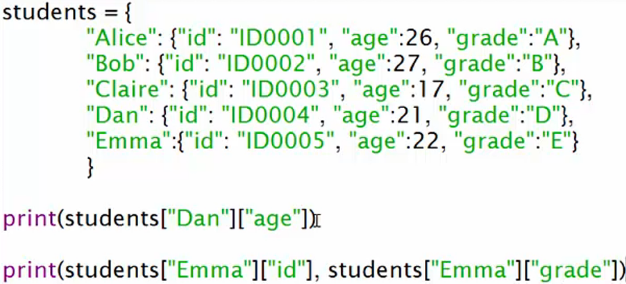

**list** **insert() => inserts a value at specified index.**  
  
\* **The insert() method also returns an empty value**.  
  
\* Lists are **MUTABLE** datatypes.  
\* **We can’t mutate a string but we can mutate a list**:  
  
\* **The remove() method also returns an empty value**.  
=> All these list methods pretty much work the same.

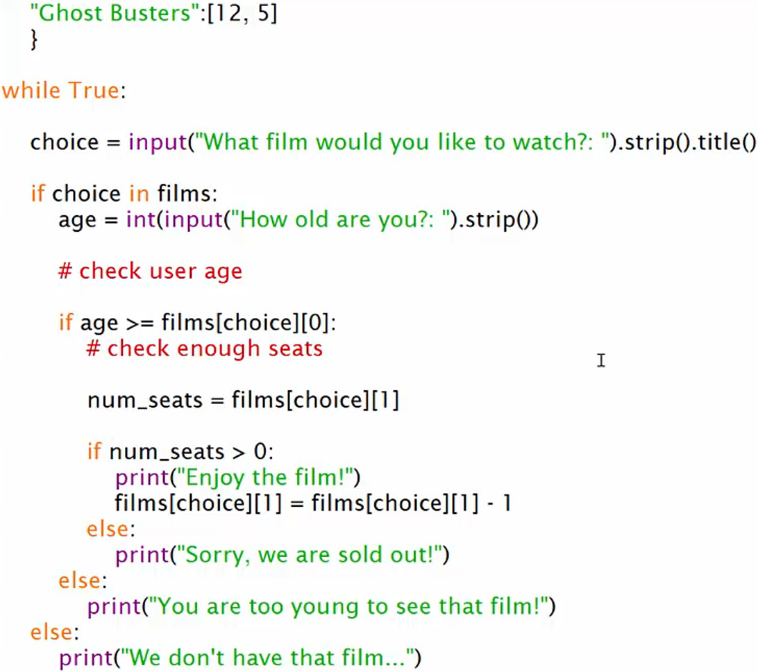
**Tuples**  
**Tuple** => **Quite common in Python and very similar to lists in how we use them**.  
=> **The only difference is that once you’ve create a Tuple, it can’t be changed**.  
=> **Immutable**=> **Iterable => made up of little pieces called elements and we can select elements and slices.**  
  
\* **It’s much more common for the sake of clarity to surround Tuples in parentheses**.  
  
\* **Mutable lists VS Immutable tuples**  
  
\* **Same with strings**.  
  
\* **Tuples are excellent for when you want to store pieces of data but you don’t want that data to ever be accidentally changed or altered in any way**.

**tuple()** => **convert other pieces of data to a tuple**.  
  
\* You can use tuple for **multiple assignment**, to **define multiple variables at once**.  
  
\* **You can do this with other Iterable datatypes as well, like lists and strings**.  


**Dictionaries Part 1 - What are Dictionaries?**  
**Dictionary** => **key: value**  
**key is a string/number**  
**name[key]**  
  
**del name[key]**  
**dict** **keys() => returns Iterable dict\_keys datatype but it’s not Indexable.**  
  
=> **We can turn the keys() to a list**.  


**dict** **values() => returns dict\_values datatype, works similarly to the dict\_keys.**\* **Both dict\_keys and dict\_values are compatible with loops, especially FOR loops.**  
  
\* **Dictionary doesn’t have an order, the only way to access an item is to use its key, you don’t use an index**.  
\* **Each key-value pair is called an item**.  
**dict** **items() => returns all the items - dict\_items - a list of tuples.**

**Dictionaries Part 2 - Build a Rough and Ready Database!**  
\* **How to combine dictionaries with lists and dictionaries**.  
  
  
  
  
  


**PROJECT 5 - Cinema Simulator!**  
**pass** => **just move on**

**Section Review**