Python Data Structures

List : Ordered, indexed

- Lists are written with square brackets.
- Access Items: sampleList[1]
- Add Element: sampleList.append("newItem")
- Add Indexed Element: sampleList.insert(1, "fruits")
- Remove Element: sampleList.remove("item")
- Remove Indexed Element: sampleList.pop(1)
- Using Delete function: del sampleList[0]
- Check the Length: len(sampleList)
- Iterate: for eachItem in sampleList:
- Element Exists? : if "apple" in sampleList:
- Clear List: sampleList.clear()

Set: Unordered, No Duplicates

- Sets are written with curly brackets
- Add Element: sampleSet.add("mango")
- Iterate: for eachItem in sampleSet:
- Element Exists? : if "apple" in sampleSet:
- Add another set: sampleSet.update(anotherSet)
- Check Length: len(sampleSet)
- Remove Element: sampleSet.remove("banana")
- Discard (No Error) : sampleSet.discard("apple")
- Clear Set: sampleSet.clear()

Tuple: Unchangeable

- Access Element: sampleTuple[1]
- Tuples are written with round brackets.
- Iterate: for eachItem in sampleTuple:
- Element Exists? if "apple" in sampleTuple:
- Check Length: len(sampleTuple)
- You cannot add, remove, delete element from Tuple
- Delete Tuple: del sampleTuple

Dictionary: Key Value Pairs

- Accessing Items: sampleDictionary["101"]
- Access with Get: sampleDictionary.get("102")
- Iterate Keys: for eachKey in sampleDictionary:
- Iterate Key, Value : for key, value in sampleDictionary.items():
- Check Key Exists? if "101" in sampleDictionary:
- Check Value Exists? if "item" in sampleDictionary.values():
- Add key value pair: sampleDictionary["104"] = "NewItem"
- Remove Item with Key: sampleDictionary.pop("101")
- Remove Last Added Item: sampleDictionary.popitem()
- Check Length: len(sampleDictionary)