Lists and Dictionaries

Arrays and Python

- Arrays are not accessible by default in Python
- Modules can be used for Arrays
- Sets - mainly Lists and Dictionaries are used like Arrays
- Lists are like Number Indexed Arrays
- Dictionaries are like Named Key Arrays
- Getting even basic math functions like Average require a bit of code

Importance of Data Formatting

- Parsing Files
- XML
- CSV files
- Example = fake-data.txt

Lists

- my_list = [8,9,77,3,2]
- my_list = [‘bob’, ‘Tim’, ‘sue’, ‘frank’]
- my_list = [‘bob’, 22, ‘large’, True]
- print(my_list[0])
- Example = list.py

List Functions

Functions - Like a tiny app

- max()
- min()
- len()
- Not an Array, No Average or Real Math Functions
  - For an Average use a For loop to add all values then divide by length of list
- Example = list-function.py

List Methods

Methods - modify variables

- sort()
- reverse()
- append()
- pop()
- remove()
- Example = list-method.py

Dictionaries

- my_dict = {'name':'bob','age':19,'size':'large','disclaimer':True}
- my_dict['name']
- Example = dict.py

Dictionary Methods

- keys()
- values()
- items()
- update()
- pop()
- Example = dict-function.py

For Loops

- for x in my_list:
- for key, value in my_dict.items():
- Example = for-loop.py

Nested Lists and Dictionaries

- Lists and Dictionaries can contain Lists and Dictionaries
- Example = nested.py

While Loop

- Code loops until condition is met
- Permanent Loop - Make sure the condition will be met
- while x < 10:
- Example = while.py

While True:

- while True:
- Goes forever unless you have a break
- Example = while-true.py

Labs

- lab-append.py
- lab-search.py
- lab-repayment.py