Dictionaries

Next time

Data Structures

Dictionaries

- Collections of **key/value pairs**
- Defined by curly brackets { }
- Slicing uses **keys**
- **Order** is not preserved
    - (well, for 3.6+ it is: https://stackoverflow.com/questions/39980323/are-dictionaries-ordered-in-python-3-6)

Python

```
1 params = {"parameter1" : 1.0, "parameter2" : 2.0,
2 "parameter3" : 3.0,}
3 print(type(params))
4 print(params)
```

DeeperDive_Dictionaries_Functions

Dictionaries

- A dictionary is a collection of **key-value pairs.**
- A key-value pair is a set of values **associated with each other.**
- A dictionary is **accessed by key** (not position)
- A **key is unique** and *must* be immutable.

Dictionaries

Below is a phone directory, which is a great example of a dictionary.

Why is that?

```
#-----------------------------#
# Luna           | 444 - 4444
# Tee            | 123 - 4567
# Ada Lovelace   | 101 - 0101
#-----------------------------#
```

DeeperDive_Dictionaries_Functions

Nested Dictionaries

Here is an example of a nested dictionary:

```
nested_example = {'info': {42: 1, type(''): 2}, 'spam': [1,2,3,'four']}
```

If we wanted to access the value associated with the key 42, you would use the syntax below:

```
print(nested_example['info'][42]) # fetches 1
```

Nested Dictionaries

- You can nest a dictionary inside another dictionary.
- For example, if you have several users for a website, each with a unique username, you can use the usernames as the keys in a dictionary.
- You can then store information about each user by using a dictionary as the value associated with their username.