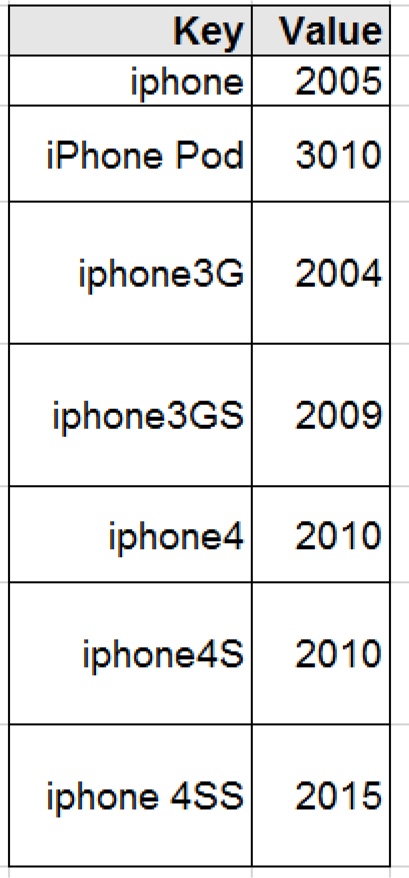
Part 3, Dictionaries Cheat Sheet

Dictionaries, the other way to store things in python

Whereas lists assign an integer index to each item, according to the order they are in, Dictionaries do something different, where you can define a "key" for each item in the dictionary, and access the item that way, it would look something like this in a spreadsheet



Key	Value
iphone	2005
iPhone Pod	3010
iphone3G	2004
iphone3GS	2009
iphone4	2010
iphone4S	2010
iphone 4SS	2015

where the key is kind of like an internal variable name, and the value is the item you are storing.

Creating dictionaries in python

You can create on with `{}` of the `dict()` coercion function, if using `{}`, you separate keys and values with a `:`. If `dict()` is used, they are separated with a `=` and the keys do not need to be written as strings. See below

```
test_dictionary = {'key_one':'value_one'}
test_dictionary2 = dict(key_one='value_one')
```

Accessing data in a dictionary

to get a value out of the dictionary, you, do so just like a list (using `[]`), but using a string of the key instead of an integer index:

```
dictionary_name['key_name']
```

You can use this code structure to access the value of a dictionary, assign that key a new value, or create a whole new key-value pair in the dictionary, combined with the :

```
test_dictionary = {'key_one':'value_one'}
print(test_dictionary['key_one']) # prints value one
test_dictionary['key_one'] = 'a new value' # assigns a new value to 'key_one'
test_dictionary['new key'] = 'another new value' # adds a new key and value to the dictionary
del test_dictionary['new key'] # deletes the key and value associated with 'new key'
```