# Python Class 3

**Dictionaries**

Dictionaries are data structures that allow you access data based on “keys” instead of position. For example, a phone book associated phone numbers with names.

>>> phoneBook = {'Alistair':1234567, 'Beatrice':1234568, 'Cordelia':1234569}

>>>phoneBook['Beatrice']

1234568

>>>phoneBook['Beatrice'] = 9876543 #update an entry

>>>phoneBook

{'Beatrice': 9876543, 'Cordelia': 1234569, 'Alistair': 1234567}

>>>phoneBook['Daphne'] = 1234570

>>>phoneBook

{'Daphne': 1234570, 'Beatrice': 9876543, 'Cordelia': 1234569, 'Alistair': 1234567}

Just like lists and strings, dictionaries have methods.

>>> phoneBook.keys()

['Daphne', 'Beatrice', 'Cordelia', 'Alistair']

>>>phoneBook.pop('Beatrice')

1234568

>>> phoneBook

{'Daphne': 1234570, 'Cordelia': 1234569, 'Alistair': 1234567}

*Exercises*

1. Use a dictionary data structure to create a very small regular dictionary that associates three words with their definitions.
2. Use the “has\_key” method to check if a word is in your dictionary.

**Loops and Dictionaries**

In the loop examples above, we used a loop to iterate over a list. Loops can also iterate over a dictionaries.

>>>phoneBook = {'Alistair':1234567, 'Beatrice':1234568, 'Cordelia':1234569}

>>> for k in phoneBook:

print(phoneBook[k])

1234568

1234569

1234567

#### *Exercises:*

1. The phone company has incremented all phone numbers by 1! Use a loop to update your phone book.

#### Data Types Review: Casting from one type to another

Here are the data types we’ve dealt with so far:

String: s = "I am a string"

Number: n = 9, n = 2.4

Boolean: b = True, b = False

List: list = [1,2,3,4]

*Exercise*

1. What data type will the last line in each example return? When possible give the result.

* "test" + "ing" + " 1" + " 2" + " 3"
* 7 == 8
* "8" == 8
* int("3") < 8
* str(3) == "3"
* str(3) + str(5)
* "test"[2]

1. Grocery list: Here we’ll write a program to keep a grocery list and allow the user to add and remove items. In a new file, follow these steps to make the program:
   1. Create a default list of groceries.
   2. Write a function to to print out a list with one item per line.
   3. Use a while loop (*example 4*) to display the list, and ask the user if they want to. If the user types quit, break from the loop..
   4. Modify the program to give the user the options to add or remove an item. If so, ask what they want to add or remove and modify the list accordingly.
   5. Modify the program so it deals with unexpected input.

**Introduction to the Python Standard Library**

Python has built in functions like int() and str() that we learned above. Python is also distributed with a standard library of additional functions. These functions are organized into “modules” and you have to tell Python when you want to use a module. We’re going to use a standard library function that randomly selects items from a list.

First, try

>>>random.choice([1,2,3,4,5])

Now, import the library and try again.

>>>import random

>>>random.choice([1,2,3,4,5])

>>>random.choice([1,2,3,4,5]) #run this line a couple times to see what happens

The complete list of functions in the standard library is available here:<http://docs.python.org/library/>

Exercises:

1. **Grocery - Feeling Lucky:** Update your Grocery List program to return a random item when you type "feeling lucky"
2. **Excuses Challenge**: For those of you who are finished, write a program that has two lists of strings, one called tasks that has elements like "take out the garbage", or "do my homework", and another one called excuses that has elements like "I was abducted by aliens" or "the dog ate it." Now have your program output a random task and excuse sentence like:  
   "I didn’t take out the garbage, because I was abducted by aliens."

#### Reading and Writing from Files

We can read and write to files from python. This is the code for opening a new file:

>>>fid = open("//directory//filename.txt","w")

Where directory is the path to some directory on your computer. The second argument, "w", tells Python to open this file and let us write in it.

Try:

>>>print fid

Write some content to the file

>>>fid.write("blah blah blah there is content in this file.\n”)

>>>fid.write(“This is the second line of the file.\n") #\n creates a new line.

Always close the file when you have done working with it:

### fid.close()

Now find the file on your computer and open it in a text editor or word processor. Is the text you just wrote there? Close the file before continuing.

### Open a file for reading:

>>>fid = open("//directory//filename.txt","r") # the ‘r’ is for **r**ead

>>>fid.read() # returns the content of the file

### *Exercises:*

1. Try the closing the file, reopening it, and reading the content with readline() or readlines()
2. Try closing the file and opening by replacing “w” with “a”. What happens when you write a line to the file?
3. Grocery - Write:Modify your grocery list program from above so that it writes the grocery list to file each time you update the list (replacing the whole file will do).
4. Grocery - Read: Write a separate program (in a new .py file) that reads your grocery list and prints out each item.

#### Processing Data After Reading

>>>fid.readlines() # returns a list of the lines in the file

### Once you have a line from a file you often will want to use the split function to :

>>>text\_line = “This is a line of text with an \n empty line".”

>>>text\_line.split() # split by default splits on any type of white space

['this', 'is', 'a', 'line', 'of', 'text', 'with', 'an', 'emptyline']

# Notice how the new line was left out as well as the spaces.

### You may also be reading a line of data. The most common and easiest way to do that is to read from a CSV, or Comma Separated File.

>>>data\_line = “342,banana,999,-100”

>>>text\_line.split() # split by default splits on any type of white space

['342,banana,999,-100']

>>>text\_line.split(‘,’) # you can also pass in a specific character or string to split with

['342', 'banana', '999', '-100']

### *Exercises:*

1. Take in a typical data file with a header row with various headers. Calculate an average of a column based on what name the user gave.
2. Talk like a pirate (Adapted from the Stanford Nifty assignment) read in a file of text and translate it into pirate speak. Below is a helpful chart of pirate speak. You should also add a few “arrr”s in for good measure. Try adding the arr in at consistent intervals, and at random intervals.

|  |  |  |
| --- | --- | --- |
| English word/phrase |  | Pirate equivalent |
| Greetings: | hello | avast |
| People: | sir | matey |
|  | madam | proud beauty |
|  | officer | foul blaggart |
| Articles: | the | th' |
|  | my | me |
|  | your | yer |
| Places: | restroom | head |
|  | restaurant | galley |
|  | hotel | fleabag inn |

### Homework:

1. If you didn’t complete it in class, do the ‘excuses’ exercise (number 18).
2. Given two words test to see if one is an anagram of the other.
3. Write a program that asks a user for a word and outputs a list with the letters from the word in a random order.

*Hint: use the list() function to convert a string to a list (myList = list(myString)) and ''.join(myList) to convert a list back to a string.*

1. Rewrite the previous program to be a game where the user has to guess the word that the computer came up with.

**Cliff Notes version:**

* Dictionaries are a data type that associates keys with values.
* Some data types can be cast to another data type.
* Some data types can be caste to other data types. For example, a string can become an integer, int(100) = “100”
* Python has some built in functions but the standard library has even more. To use these functions, import the library.
* There is a standard library function that can pick a random item from a list.

>>>import random

>>>random.random([1,2,3,4])

* Python can read and write files.