CS 133 - Introduction to Computational and Data Science

Instructor: Renzhi Cao
Computer Science Department
Pacific Lutheran University
Spring 2017

Announcement

- Mid-term exam
- Quiz 3 and the questions
- Should we by default write out code into functions? When and what parameters?
- How to open, write and close file?
- How to integrate loops into lists?
- About while loops and how to use it?
- How to use the dictionary?

Getting data

- Read Chapter 9.
- We have learned the probability and statistics.
- Today we are going to give more details about processing data, read files with different format, including CSV, html.

How to be a data scientist

1. Two parts, first you need Data! And then have a scientific thinking, to be a scientist on data!
   A. Acquiring it
   B. Cleaning it
   C. Transforming it

We all agree that typing the data in yourself is not practical.

Let's use python

Getting data with Python summary

1. stdin and stdout
2. Files
3. CSV files
4. HTML
5. APIs

Stdin and stdout

Stand for standard input and standard output.

You can pass information from the command line with those commands.

Need to import sys module

stdin is a list

stdin[0] is the name of the program

stdin[1] is the value after the name of the program

python test.py 1

What is stdin?

What is stdin[0]?

What is stdin[1]?

python test.py 1

stdin: [test.py,1]

stdin[0]: test.py

stdin[1]: 1

Examples

```
import sys


x = sys.argv
print "the parameters " + str(x)
print "It has " + str(len(x)-1) +" parameters"
print "which are " + str(sys.argv[1]) +" and " + str(sys.argv[2])
total= int(sys.argv[1]) + int(sys.argv[2])
sys.stdout.write("Addition is" + str(total) +"\n")
```

What happens if data is not there?

The program will crash!

How to avoid it?

Try Except

In other words:

If you are not able to do this part of the code, send an error message and usually a termination signal (sys.exit(1))

The 1 stands for “unsuccessful exit”

sys.exit(0) is a “successful exit” (no errors)

Try except

```
import sys
try:
        x = sys.argv
        a = str(sys.argv[1])
        b = str(sys.argv[2])
except:
        print " Incorrect number of paramters. Usage: std_example number1 number2"
        sys.exit(1)
# Rest of the code
```

Let's do together

1. Create a program that reads a string with 5 words and a number between 1 and 5.
2. It will only print to the command line one word: The word located in the position of the number given

Example:

Python test.py “Hello everybody today is Tuesday” 5

Tuesday

Files

We covered files before. What else can we say about them?

fp = open(“file.txt”,”r”) # Read

fp = open(“file.txt”,”w”) # Write

fp = open(“file.txt”,”a”) # Append

How to write to files?

fp.write(“This is a test”)

fp.close() #VERY IMPORTANT!!!!

Create CSV

Use list of lists!

```
s= [["a","b"], ["x","y"]]
import csv
with open("data.csv",'w') as fp:
    writer = csv.writer(fp)
    writer.writerows(s)
```

Delimited files

CSV file or other types of delimiters? (Comma, space, tab, etc)

Import csv

fp = open("data.csv","r")

reader = csv.reader(fp, delimiter = "\t") # for tabs

What about files that have headers?

Ignore them:

next(reader,None)

Use them!

Use headers

Assuming a file called test.txt has the following header

ID|Name|Day

Data

```
fp = open("test.txt",r)

reader = csv.DictReader(fp, delimiter = "|")

for row in reader:
    val1 = row["ID"]
    val2 = row["Name"]
    val3 = row["Day"]
    #Do something with the values before it loops again and you lose those values!
```

HTML

HTML = Hyper Text Markup Language

Let's look at examples

```
<html><head><title>A story</title></head>
<body>
<p>Once upon a time there were three little sisters; and their names were
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>
</body>
</html>
```

We are going to use the external modules BeautifulSoup and requests

What do we need to learn about the site before we start?

You can get banned by sending multiple requests to a site.

If we are working with a company, it is always good practice to look at their terms before you do any work

http://oreilly.com/terms/

All websites will have a file called robots.txt that determines who and how often you can request information from their site

http://shop.oreilly.com/robots.txt/

Beautiful Soup

http://www.crummy.com/software/BeautifulSoup/bs4/doc/

Let's follow the tutorial together

Example: O'Reilly Books about data

Page 110 to page 114

Let's work on that example

Work in groups or on your own.

More Exercises

1. Create a file called “numbers.txt” in which you will insert 300 random numbers between 1 and 100 (one per line)
2. Close the file
3. Reopen the file and calculate the mean of the values
4. Close the file
5. Open the file to append 1000 more random numbers between 1 and 100
6. Close the file
7. Reopen the file to calculate the mean of the values
8. Close the file