

Introduction to Data Analysis

Session 3, Nov. 15



Today's Overview

- Morning lecture
 - New tools
 - Data visualization
 - Harry Potter on Wikipedia
 - Working on data sets in Google docs
- Lunch
- Afternoon sessions
 - Continuation of this morning's lecture
 - `matplotlib` session
 - Self-directed
- Closing remarks

while loops

```
>>> while True:  
...     ((do something))  
...  
...     if ((condition)):  
...         break
```

string.join

```
>>> list = ['a', 'b', 'c']
```

```
>>> list
```

```
['a', 'b', 'c']
```

```
>>> ','.join(list)
```

```
'a,b,c'
```

File Reading Review

```
>>> file = open('myfile', 'r')
```

- 'r' stands for “read” mode

```
>>> line = file.readline()
```

```
>>> line
```

```
Some text from the file
```

Read the file in other ways

```
>>> for line in file:  
...     ((do something))
```

- Iterate over lines in a file

```
>>> file_as_string = file.read()
```

```
>>> print(file_as_string)
```

```
'Lots of text from the file in  
string form...'
```

- Read file into Python all at once as a single string

csv Module

- It's common to store data in “comma-separated value” format (csv)
- Data looks like

```
"Harry_Potter", "Discospinster", 2014-10-30 22:48:24, 112947, False, True, 631814175
```
- We want to be able to process this to aggregate and do analysis
- Built-in csv module allows us to do so

csv.reader()

```
>>> # Read in file as line list
>>> import csv
>>> with open('hp_wiki.csv', 'r') as csvfile:
...     reader = csv.reader(csvfile,
...                           delimiter=',')
...     for row in reader:
...         ((do something))
```

csv.DictReader()

```
>>> # Read in file as list of dictionaries
>>> import csv
>>> reader = csv.DictReader(open('file', 'r'))
>>> for line in reader:
...     ((do something))
```

Live Data Processing Demos!