

CS101: Intro to Computing

Fall 2015

Lecture 14

Administrivia

- Homework 11 released on Monday
- On RELATE, not Codelab
- Test your code before submitting it
- Install Anaconda

<https://www.continuum.io/downloads>

REVIEW

???

```
math.degrees(2*pi)
```

What should replace the ???

a)

```
from math import pi  
import math
```

b)

```
from math import pi,degrees
```

c)

```
import pi  
import math
```

d)

```
import math
```

```
x=1
```

```
x+=1 # x+=1
```

```
, , ,
```

```
x+=1
```

```
, , ,
```

```
x+=1
```

What is the final value of x?

a) 1

b) 2

c) 3

d) 4

USES FOR DICTIONARIES

Dictionaries to Join/Merge Data

- We can link data based on a common field

```
zip={"Bill":60644,  
    "Jim":41073,"Beth":63103}  
city={60644:"Chicago",  
      41073:"Cincinnati",  
      63103:"St. Louis"}  
for name in zipcode:  
    print name,city[zipcode[name]]
```

Exercise

- Print the album, artist, and track names for each song in tracks.csv


```
import csv
# read in artist data from csv file
artist_file=open("artist.csv")
artist_d={}
artist_reader=csv.DictReader(artist_file)
for artist in artist_reader:
    artist_id=artist["ArtistId"]
    name=artist["Name"]
    artist_d[artist_id]=name
artist_file.close()
```

```
# read in album data from csv file
album_file=open("album.csv")
album_reader=csv.DictReader(album_file)
album_d={}
for album in album_reader:
    album_id=album["AlbumId"]
    album_t=album["Title"]
    artist=album["ArtistId"]
    # store artist/album as a tuple
    album_d[album_id]=(album_t,artist)
album_file.close()
```

```
#read track data and merge with album
# and artist data to print it out
track_file=open("track.csv")
track_reader=csv.DictReader(track_file)
for track in track_reader:
    name=track["Name"]
    album_id=track["AlbumId"]
    title,artist_id=album_d[album_id]
    arstist=artist_d[artist_id]
    print name,",",title,",",artist
track_file.close()
```

MATPLOTLIB MODULE

Matplotlib

- Plotting library (module) for Python
- Not included in standard Python libraries
- Pylab interface imitates Matlab

```
import matplotlib.pyplot as plt
x=[ 0,1,2 ]
y=[ 1,2,3 ]
plt.plot(x,y)
plt.show( )
```

Title and Axis Labels

- Should always label axes and title our graphs

```
plt.title("Example plot")
```

```
plt.xlabel("X data")
```

```
plt.ylabel("Y data")
```

Plot

- Function takes at least two arguments
 - List of x and y coordinates
- Can optionally take a string indicating the color/shape of the line

```
plt.plot(x,y, 'r.')
```

- Can also take a *label* keyword argument

```
plt.plot(x,y, 'r.', label="Fun!")
```

Plot

- Can plot multiple lines at once

```
x=[ 1 , 2 , 3 , 4 ]
```

```
y1=[ 2 , 4 , 6 , 8 ]
```

```
y2=[ 3 , 6 , 9 , 12 ]
```

```
plt.plot(x,y1, 'r-', label="2x" )
```

```
plt.plot(x,y2, 'g--', label="3x" )
```


Legend

- With multiple lines, a legend is helpful

```
plt.legend( )
```

Bar charts

- Used to show amounts associated with a set of values

```
a=[ 1 , 2 , 3 ]
```

```
b=[ 3 , 4 , 4 ]
```

```
ticks=[ "Donna" , "Jo" , "Sandy" ]
```

```
plt.xticks(x,ticks)
```

```
plt.bar(a,b,label="Bar" )
```

Exercise

- Plot the year vs. HR in baseball data for 1980-present

```
import csv
import matplotlib.pyplot as plt

homers={}
f=open("./lahman-csv_2015-01-24/Batting.csv")
for r in csv.DictReader(f):
    y=int(r["yearID"])
    if y<1980:
        continue
    h=r["HR"]
    h.strip()
    if h=="":
        h=0
    else:
        h=int(r["HR"])
    if y not in homers:
        homers[y]=0
    homers[y]+=h
f.close()
```

```
x=[]
y=[]
ticks=[]
for year in homers:
    hr=homers[year]
    x.append(year)
    y.append(hr)
    ticks.append(str(year))

plt.xticks(x,ticks)
plt.bar(x,y)
plt.title("Year vs. HR\nin MLB")
plt.show()
```

Histograms

- Used to show ***distribution*** of values

```
counts=[1,1,2,3,1,2,1,3,1,2,1]
```

```
plt.hist(counts)
```

- Bins the values and counts ***frequency*** of values that range

Exercise

- Plot distribution of word use in Jeopardy questions

```
counts={}
for line in open("jeopardy.txt"):
    if line[0]!="#":
        split=line.split("\t")
        air_date,answer,category,question,
            rnd,show_number,value=split
        value=value.replace(",","").strip()
        if value=="":
            continue
        words=question.lower.split()
        for word in words:
            if word not in counts:
                counts[word]=0
            counts[word]+=1
```



```
x=[ ]
for word in counts:
    x.append(counts[word])

x.sort()
x=x[-100:] # plot only the top 100 counts

import matplotlib.pyplot as plt
plt.hist(x)
plt.xlabel("Number of times used")
plt.ylabel("Number of words")
plt.show()
```

Scatter Plot

- Scatter plots graph points in 2D

```
plt.scatter(x,y,  
            label="Data",  
            color='g',  
            marker='*')  
plt.show()
```

Exercise

- Plot the year vs. RBI for Pete Rose

```
import csv
import matplotlib.pyplot as plt

f=open("./lahman-csv_2015-01-24/Batting.csv")
years=[]
rbis=[]
for r in csv.DictReader(f):
    if r["playerID"]=="rosepe01":
        y=int(r["yearID"])
        rbi=int(r["RBI"])
        years.append(y)
        rbis.append(rbi)

f.close()

plt.scatter(years,rbis)
plt.show()
```