CS101: Intro to Computing
Spring 2016

Lecture 15
Administrivia

• Homework 8 released on RELATE
• Reminder: one homework assignment dropped
REVIEW
NameError: name 'y' is not defined

What code produces this error?

a) \( x=1 \)
   \( y=x*2 \)

b) \( x=0 \)
   \( y+=1 \)

c) \( x=\text{"ABCD"} \)
   \( y=x[2] \)
What code produces this error?

a) `x = "ABCD" + "E"
   
   x[5]`

b) `x = [1, 2]
   
   x[2]`

c) `x = {1: 2, 2: 3}
   
   y = x[2]`
What code produces this error?

a) if x < "HAPPY":
    print(x.lower()[1])

b) if x in "ABCD":
    print("E"+x[0])

c) if x = (1,2,3):
    print(x[2]+1)
DEBUGGING
“My code doesn’t work.”
“My code doesn’t work.”

1. How do I know it isn’t working?
2. What do I expect it to do?
3. What is my code doing instead? Why?

Eliminate all other factors, and the one which remains must be the truth.
Errors = Clues

1. Read message and **think** about it.
   
   **AttributeError**: 'int' object has no attribute 'append'
   
   **IndexError**: list index out of range

2. Google the error message
   
   **TypeError**: unsupported operand type(s) for %: 'int' and 'str'
Don’t just stare at your code!

• Make the **computer** work, not you!
• Add print statements (especially in loops!)

Data! Data! Data!
I can't make bricks without clay.
Debugging Tips

1. Comment out lines to narrow down your search for the problem.
2. Add print statements.
3. Trace troublesome input through your code line by line.
4. Explain the problem to someone else.
5. Take a break.
Coding tips

1. **START EARLY!**
2. Break the problem down and write functions.
3. Document your functions *before* writing them.
4. Test every function (even every *line*) you write to make sure it works (write then execute, write then execute).
Ultimate Coding Secret

• If you’re confused, frustrated, and can’t make progress…

Always approach a case with an absolutely blank mind.

• … start over.
You know my methods. Apply them.
Course Summary (so far…)

1. Python fundamentals
2. Data wrangling
3. Data visualization
4. Simulation
5. Random processes
6. Optimization
MATPLOTLIB MODULE
Matplotlib

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

```python
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

```python
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
  
  ```python
  plt.plot(x,y,'r.')
  ```
• Can also take a `label` keyword argument
  
  ```python
  plt.plot(x,y,'r.', label="Fun!")
  ```
Plot

• Can plot multiple lines at once

```python
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

\[ a = [1, 2, 3] \]
\[ b = [3, 4, 4] \]
\[ \text{ticks} = ["Donna", "Jo", "Sandy"] \]
\[ \text{plt.xticks}(x, \text{ticks}) \]
\[ \text{plt.bar}(a, b, \text{label}="Bar") \]
Which of these will correctly display a plot?

a) import matplotlib.pyplot as plot
   plot.plt([[1,2,3],[4,5,6]])

b) import matplotlib.pyplot as plt
   plt.plot([[1,2,3],[4,5,6]])

c) from matplotlib import pyplot
   pyplot.plt([[1,2,3],[4,5,6]])

d) None of the other answers
```python
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 in MLB")
plt.show()
```
Histograms

- Used to show *distribution* of values
  
  \[
  \text{counts} = [1, 1, 2, 3, 1, 2, 1, 3, 1, 2, 1]
  \]
  
  ```python
  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

```python
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()
USES FOR DICTIONARIES
Dictionaries to Join/Merge Data

• We can link data based on a common field.

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