

# SAMS

## Programming A/B

Week 6 Lecture – Odds & Ends  
Files and Simple Graphics  
August 7, 2017

Mark Stehlik

# Outline for Today

---

- Questions about the HW/upcoming Quiz!
  - HINT – test (or at least call) your functions!
- File input and output
- Simple graphics with the Tkinter module
- Life after SAMS

# Files

---

- Most data is stored in files (text documents, music libraries, photo albums, etc.)
- Python has some builtin tools to read and write (create) data files
- Reading files
  - `f = open("name-of-file.txt")`
  - what does *open* return, do you think?
  - what happens if the file is not there?

# Files

---

- The *open* function returns a file object
- If the file is not there, an error/exception occurs
- This exception can be handled explicitly with  
try:  
Except:
- Files can also be opened for writing
  - `outfile = open("name-of-file.txt", "w")`
- There is even a library for reading URL's
  - `import urllib.request`

# Files

---

- Some useful methods that act on file objects:
  - readline() – returns a string containing the line
  - readlines() – returns a list of all the strings (lines) in the file
  - close() – once you're done reading, close the file
- And so, it's back to strings...
  - And some useful string methods
    - <str>.rstrip( ), <str>.lstrip( ), <str>.strip( )
    - <str>.split( )

# Let's try some examples...

---

- Reading a small file, a line at a time
- Reading the entire file into a list
- Reading a web page

# Graphics with Tkinter

---

- Tkinter is the Python interface to the Tk Graphical User Interface (GUI), a set of *widgets* that can be used to create a graphical interface.
  - (And it was created by CMU alum John Ousterhout)
- We will use the canvas widget to draw some simple graphics; documentation for the canvas widget can be found here:
  - <http://effbot.org/tkinterbook/canvas.htm>

# The canvas widget

---

- First you need to create an empty canvas:

```
from tkinter import *
```

```
root = Tk()
```

```
canvas = Canvas(root, width=500, height=300)
```

```
canvas.pack()
```

```
# your drawing code goes here
```

```
root.mainloop()
```

```
# this blocks, so close the window when finished!
```



# Drawing with Tkinter

---

- The basic idea is
  - `canvas.create_graphic(coordinates, options)`
  - E.g., `canvas.create_line(x1, y1, x2, y2, options)`
- Shapes require a *bounding box*:
  - Two x,y coordinate pairs
  - Upper-left corner, lower-right corner
  - (0,0) is where in the window?

# Functions to draw shapes

---

- The functions to draw shapes that we will use:
  - `canvas.create_arc(bounding-box, options)`
  - `canvas.create_image(position, option)`
  - `canvas.create_line(coordinates, options)`
  - `canvas.create_oval(bounding-box, options)`
  - `canvas.create_polygon(coordinates, options)`
  - `canvas.create_rectangle(bounding-box, options)`
  - `canvas.create_text(position, option)`
- Let's do some drawing...