

UNIT 10A

Visualizing Data: Graphics in Python

15110 Principles of Computing, Carnegie
Mellon University - CORTINA

1

Drawing using Python

- We will be using Python's interface to Tcl/Tk, a cross-platform graphics library.
 - To use this, you should be logged in directly into the Andrew machines or logged in remotely (using ssh) with an X client running

15110 Principles of Computing, Carnegie
Mellon University - CORTINA

2

Creating a window for drawing

```
import tkinter
window = tkinter.Tk()
c = tkinter.Canvas(window, width=1000, height=1000)
c.pack()
```

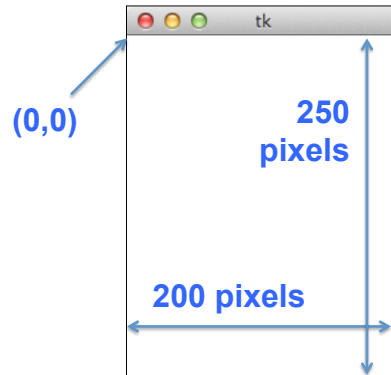
- The **window** is what pops up on your computer.
- The **canvas** is what we create graphics on within the window.
- **Packing** the canvas tells it to occupy the entire window.

Coordinate System

- When drawing on a canvas, the location of the origin is at the TOP LEFT, not bottom left.
 - x increases left to right
 - y increases top to bottom
- Coordinates are based on **PIXELS**, not other units like inches or millimeters.

Coordinate System

```
>>> import tkinter
>>> window = tkinter.Tk()
>>> c = tkinter.Canvas(window,width=200,height=250)
>>> c.pack()
```



15110 Principles of Computing, Carnegie Mellon University - CORTINA

5

Drawing Rectangles

```
canvas_name.create_rectangle(
    x0, y0, x1, y1, optional_params)
```

- Draw a rectangle from top left (x_0, y_0) to bottom right (x_1, y_1) in units of pixels.
- Optional parameters:
 - `fill="color"` (default: none)
 - `outline="color"` (default: "BLACK")
 - color can also be specified in hex as "#RRGGBB"
 - `width=width` (default: 1)

15110 Principles of Computing, Carnegie Mellon University - CORTINA

6

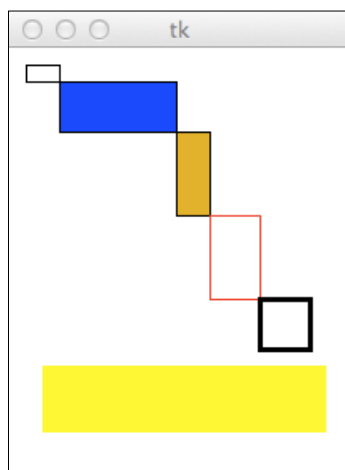
Drawing Rectangles

```
>>> from tkinter import *
>>> window = Tk()
>>> c = Canvas(window, width=200, height=250)
>>> c.pack()
>>> c.create_rectangle(10,10,30,20)
>>> c.create_rectangle(30,20,100,50,fill="blue")
>>> c.create_rectangle(100,50,120,100,fill="#DAA520")
>>> c.create_rectangle(120,100,150,150,outline="red")
>>> c.create_rectangle(150,150,180,180,width=3)
>>> c.create_rectangle(20,190,190,230,
    fill="#FFFF00",width=0)
```

15110 Principles of Computing, Carnegie
Mellon University - CORTINA

7

Drawing Rectangles



15110 Principles of Computing, Carnegie
Mellon University - CORTINA

8

Circles

```
canvas_name.create_oval (  

    x0, y0, x1, y1, optional_params)
```

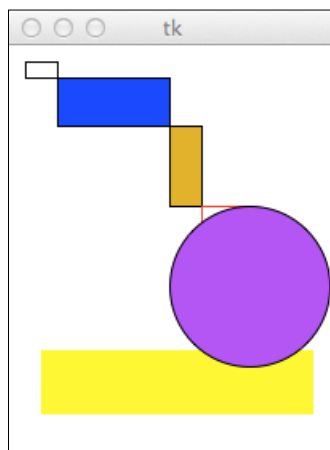
- Draw an oval inscribed in a bounding box from top left (x_0, y_0) to bottom right (x_1, y_1) in units of pixels.

```
>>> c.create_oval(100, 100, 200, 200,  

                    fill="#FF00FF")
```

Note how the window acts like a painter's canvas.

Circles



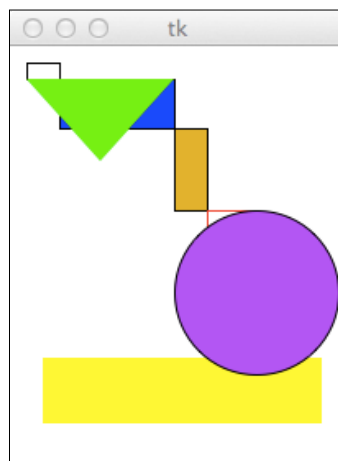
Polygons

```
canvas_name.create_polygon(
    x0, y0, x1, y1, ..., xn, yn,
    optional_params)
```

- Draw a Polygon with vertices taken from the list of points as follows: $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$. (The last vertex is connected back to the first vertex.)

```
>>> c.create_polygon(10, 20, 100, 20,
                    55, 70, fill="green")
```

Polygons



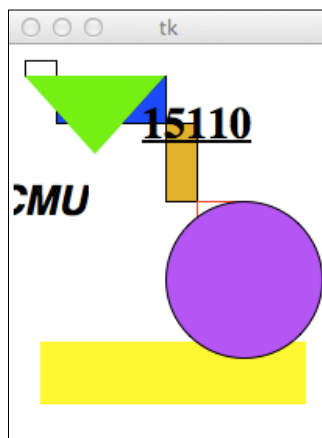
Text on the canvas

```
canvas_name.create_text(  
    x0, y0, text=string, font=font)
```

- Draw a text object centered on the given x and y coordinates.

```
>>> c.create_text(20,100,text="CMU",  
                 font="Helvetica 26 bold italic")  
>>> c.create_text(120,50,text="15110",  
                 font="Times 28 bold underline")
```

Text

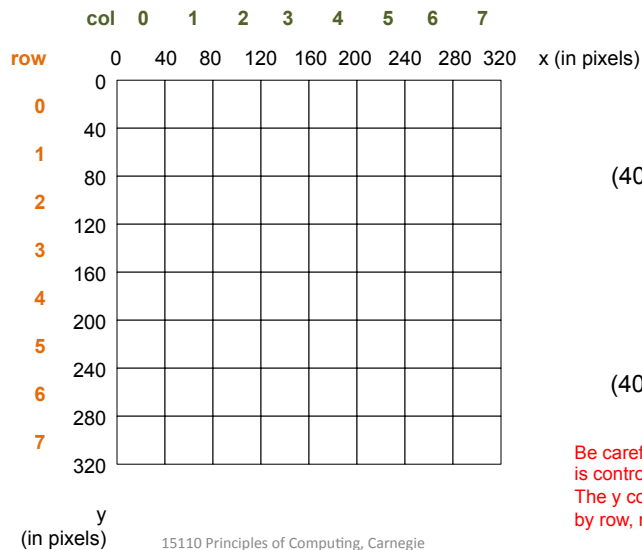


Example

In `graphicsdemo.py`, write a Python function `demo()` that creates a window of size 320 by 320 and draws a grid of 8 by 8 squares, each of size 40 by 40 pixels, and colored a random color of red, green or blue for each square.

The random number generator is seeded with the number 15110 to generate the same sequence of pseudorandom numbers each time the function is called.

Coordinates for the Squares



$(40 * col, 40 * row)$

each square

$(40 * (col + 1), 40 * (row + 1))$

Be careful! The x coordinate is controlled by col, not row. The y coordinate is controlled by row, not col.

Programming Example

```

from tkinter import *
from random import randint, seed
def demo():
    window = Tk()
    c = Canvas(window, width=320, height=320)
    c.pack()
    colors = ["red", "green", "blue"]
    seed(15110)
    for row in range(0,8):
        for col in range(0,8):
            randcolor = colors[randint(0,2)]
            c.create_rectangle(40*col, 40*row, \
                40*(col+1), 40*(row+1), fill=randcolor)
    window.mainloop() #keep the window open until we close it
    return None

```

15110 Principles of Computing, Carnegie
Mellon University - CORTINA

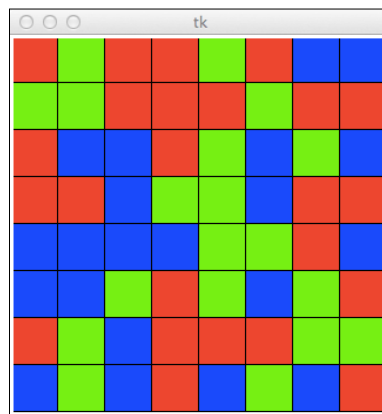
17

Results

```

python3 -i graphicsdemo.py
>>> demo()
>>>

```



15110 Principles of Computing, Carnegie
Mellon University - CORTINA

18