

Announcements

Academic Integrity

<https://www.cs.cmu.edu/~112/syllabus.html#academicIntegrity>



“Spies Like Us” <https://youtu.be/d7Aot4Wr-Yo?t=46>



15-112
Lecture 2

Week 2 Thu
Strings &
Graphics

Instructor: Pat Virtue

Strings (cont.)

Poll 7

Which code block prints this? Select ALL that apply

'abc def'

- A. `print("'abc def'")`
- B. `print("'abc\tdef'")`
- C. `print("'abc/tdef'")`
- D. `print(repr("'abc def'"))`
- E. `print(repr("'abc\tdef'"))`
- F. `print(repr("'abc/tdef'"))`

String methods

From notes:

s	isalnum	isalpha	isdigit	islower	isspace	isupper
ABCD	True	True	False	False	False	True
ABcd	True	True	False	False	False	False
abcd	True	True	False	True	False	False
ab12	True	False	False	True	False	False
1234	True	False	True	False	False	False
	False	False	False	False	True	False
AB?!	False	False	False	False	False	True

Design Challenge

```
def toCamelCase(s):  
    pass
```

```
assert(toCamelCase('goodToGo') == 'goodToGo')
```

```
assert(toCamelCase('Hi Linus') == 'hiLinus')
```

```
assert(toCamelCase('add_all_the_numbers') == 'addAllTheNumbers')
```

Poll 8

Which of the following would you want to use in your `toCamelCase(s)` implementation?

Select ALL that apply

- A. For loop over the characters in the string
- B. `isalnum`
- C. `isalpha`
- D. `isdigit`
- E. `islower`
- F. `isspace`
- G. `isupper`
- H. None of the above

Design Challenge

```
def toCamelCase(s):  
    pass
```

```
assert(toCamelCase('goodToGo') == 'goodToGo')
```

```
assert(toCamelCase('goodtogo') == 'goodtogo') # Oh well
```

```
assert(toCamelCase('Hi Linus') == 'hiLinus')
```

```
assert(toCamelCase('add_all_the_numbers') == 'addAllTheNumbers')
```


Design Challenge

```
def printFunctionInfo(code):  
    pass
```

```
code = """  
def f(x):  
    return 4*x
```

```
def ct(x, y, z):  
    return f(x) + f(y+1) + f(x+2)
```

```
print(ct(2))  
"""
```

```
print(code)  
#print(repr(code))
```

```
printFunctions(code)
```

Output

Function f takes parameters:

x

Function ct takes parameters:

x

y

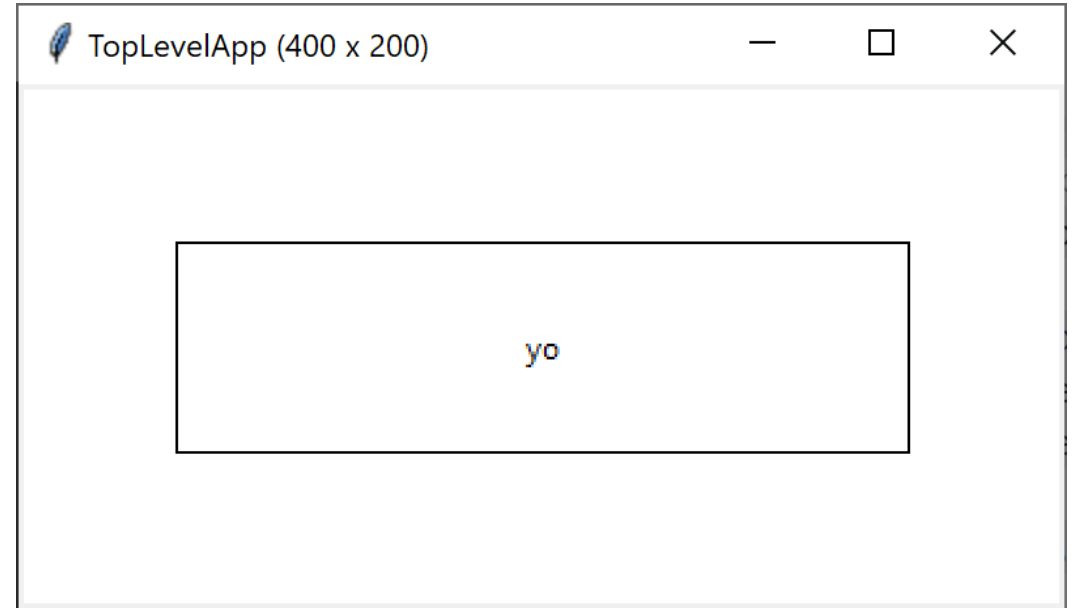
z

Graphics (Tkinter)

Tkinter comes from Tk graphics framework

CMU 112 Graphics

```
from cmu_112_graphics import *  
  
def redrawAll(app, canvas):  
    canvas.create_rectangle(60, 60, 340, 140)  
    canvas.create_text(200, 200, text='yo')  
  
runApp(width=400, height=200)
```



A few helpful tips

- Make sure `cmu_112_graphics.py` is in same folder as your code
 - (Yes, you'll be copying this all over the place this semester)
- Fix the name if you ended up with copies: `cmu_112_graphics(2).py`

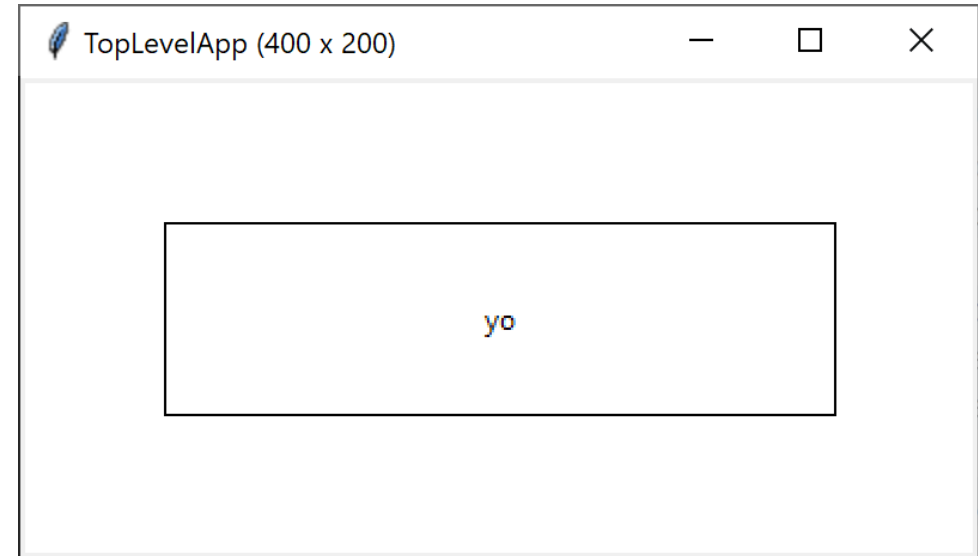
```
▪ from cmu_112_graphics import *
```

NOT: `from cmu_112_graphics.py import *`

NOT: `import cmu_112_graphics`

Coordinate for graphics

```
from cmu_112_graphics import *  
  
def redrawAll(app, canvas):  
    canvas.create_rectangle(  
        60, 60, 340, 140)  
  
    canvas.create_text(  
        200, 200,  
        text='yo')  
  
runApp(width=400, height=200)
```



Rectangles, ovals, and lines

```
canvas.create_rectangle(x0, y0, x1, y1)
```

```
canvas.create_oval(x0, y0, x1, y1)
```

```
canvas.create_line(x0, y0, x1, y1)
```

```
canvas.create_line(x0, y0, x1, y1, x2, y2, x3, y3) # etc.
```

Common parameters (myColor1 = 'purple3')

```
fill = myColor1
```

```
outline = myColor1
```

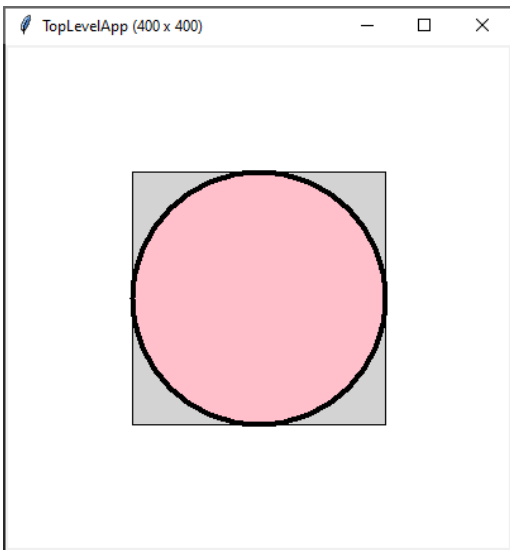
```
width = 7
```

Poll 9

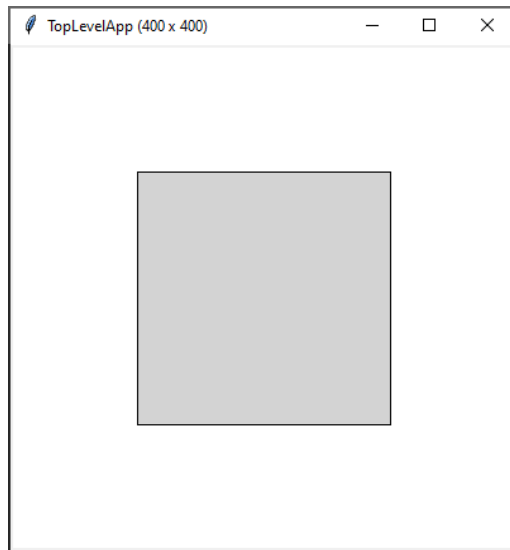
Which does this draw?

```
def drawBubble(app, canvas):  
    canvas.create_oval(100, 100, 300, 300, fill='pink')  
    canvas.create_rectangle(100, 100, 300, 300, fill='lightgray')
```

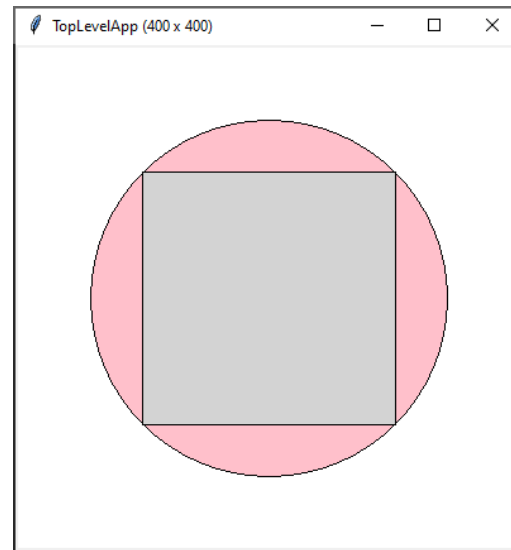
A.



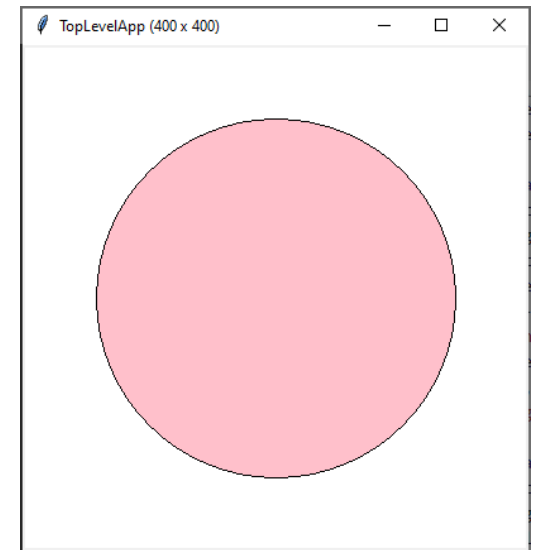
B.



C.



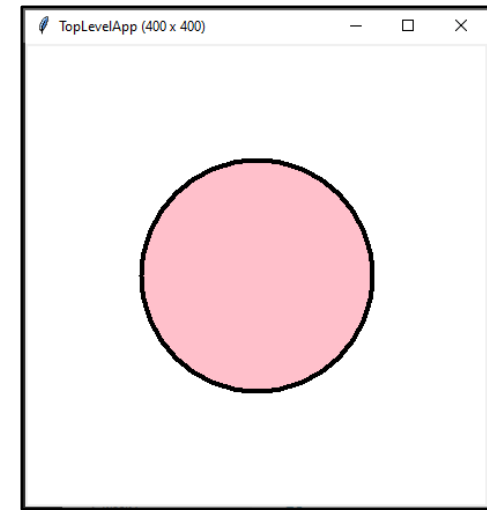
D.



E. I have no idea

Poll 10

Which of these is best?

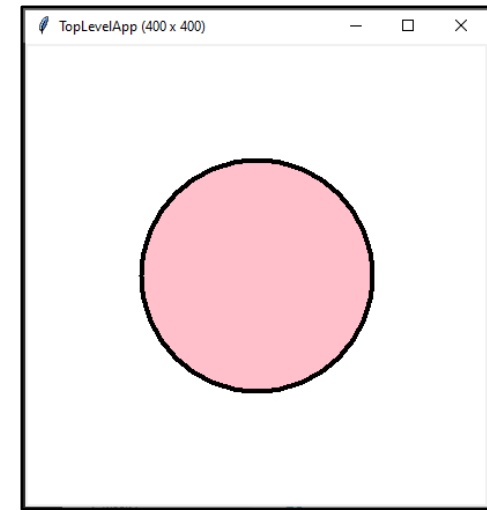


```
def drawBubbleA(app, canvas):  
    canvas.create_oval(  
        100, 100, 300, 300,  
        fill='pink', width=4)
```

```
def drawBubbleB(app, canvas):  
    topLeftX = 100  
    topLeftY = 100  
    bottomRightX = 300  
    bottomRightY = 300  
    canvas.create_oval(  
        topLeftX, topLeftY,  
        bottomRightX, bottomRightY,  
        fill='pink', width=4)
```

Poll 11

Which of these is best?

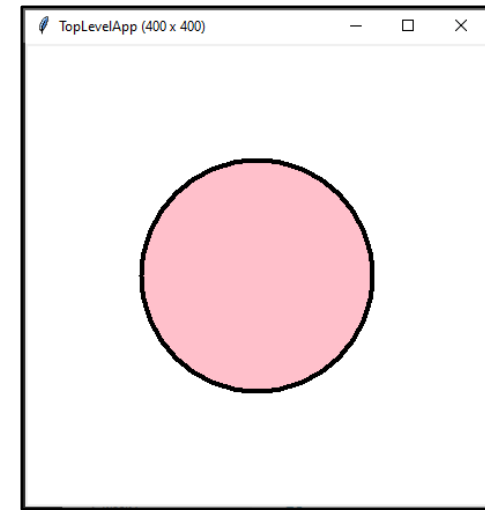


```
def drawBubbleC(app, canvas):  
    topLeftX = (1/4)*app.width  
    topLeftY = (1/4)*app.height  
    bottomRightX = (3/4)*app.width  
    bottomRightY = (3/4)*app.height  
    canvas.create_oval(  
        topLeftX, topLeftY,  
        bottomRightX, bottomRightY,  
        fill='pink', width=4)
```

```
def drawBubbleB(app, canvas):  
    topLeftX = 100  
    topLeftY = 100  
    bottomRightX = 300  
    bottomRightY = 300  
    canvas.create_oval(  
        topLeftX, topLeftY,  
        bottomRightX, bottomRightY,  
        fill='pink', width=4)
```


Poll 12

Which of these is best?



```
def drawBubbleC(app, canvas):  
    topLeftX = (1/4)*app.width  
    topLeftY = (1/4)*app.height  
    bottomRightX = (3/4)*app.width  
    bottomRightY = (3/4)*app.height  
    canvas.create_oval(  
        topLeftX, topLeftY,  
        bottomRightX, bottomRightY,  
        fill='pink', width=4)
```

```
def drawBubbleD(app, canvas):  
    cx = app.width/2  
    cy = app.height/2  
    r = app.width/4  
    canvas.create_oval(  
        cx-r, cy-r, cx+r, cy+r,  
        fill='pink', width=4)
```

Text

```
myString = 'yo'
```

```
canvas.create_text(x, y, text=myString)
```

Common parameters

```
font = myFont
```

```
anchor = myAnchor
```