Building GUIs with tkinter

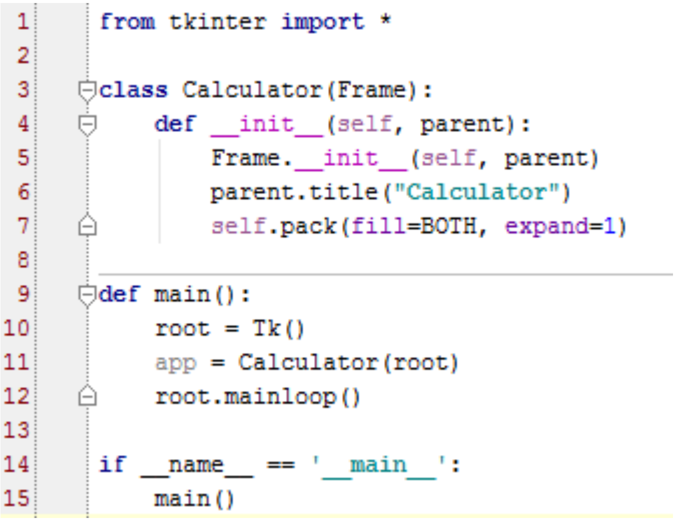
Jordan Woehr
Python Meetup - July 3, 2013

Why tkinter

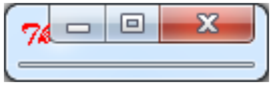
- Alternatives
  - wxPython
  - Qt
  - PyGTK

- Tkinter
  - Easy
  - Built-in
  - Quick

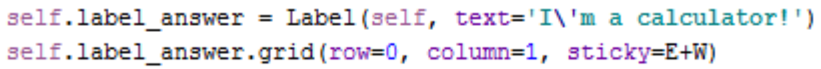
Creating an application



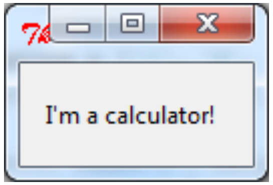
```python
from tkinter import *

class Calculator(Frame):
    def __init__(self, parent):
        Frame.__init__(self, parent)
        parent.title("Calculator")
        self.pack(fill=BOTH, expand=1)

def main():
    root = Tk()
    app = Calculator(root)
    root.mainloop()

if __name__ == '__main__':
    main()
```



Label
Grid versus pack

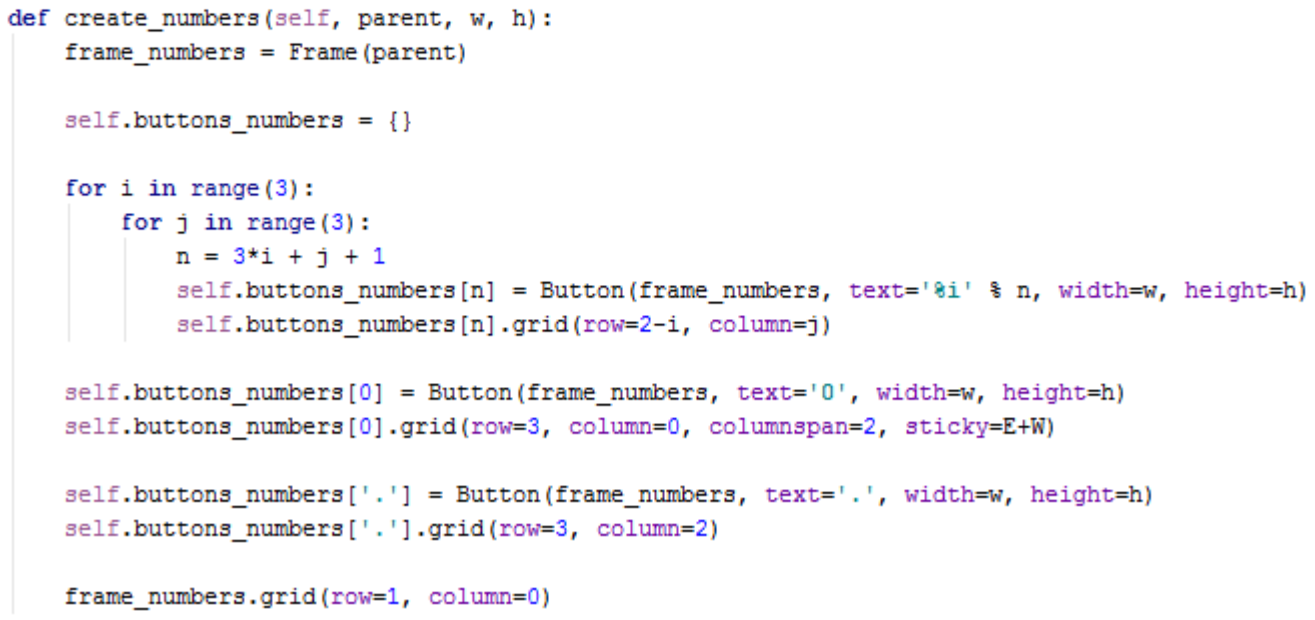


```
self.label_answer = Label(self, text='I\'m a calculator!')
self.label_answer.grid(row=0, column=1, sticky=E+W)
```

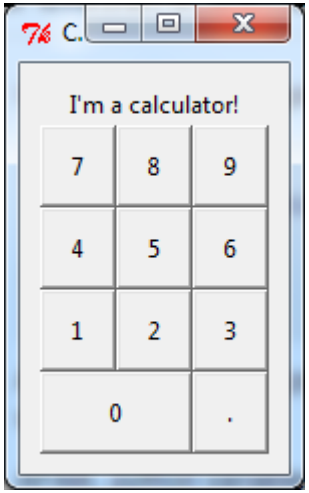


I'm a calculator!

Buttons



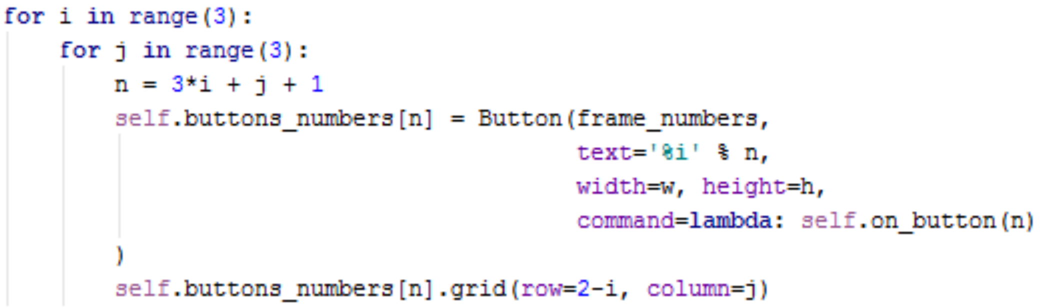
```
def create_numbers(self, parent, w, h):
    frame_numbers = Frame(parent)

    self.buttons_numbers = {}

    for i in range(3):
        for j in range(3):
            n = 3*i + j + 1
            self.buttons_numbers[n] = Button(frame_numbers, text='%i' % n, width=w, height=h)
            self.buttons_numbers[n].grid(row=2-i, column=j)

    self.buttons_numbers[0] = Button(frame_numbers, text='0', width=w, height=h)
    self.buttons_numbers[0].grid(row=3, column=0, columnspan=2, sticky=E+W)

    self.buttons_numbers['.'] = Button(frame_numbers, text='.', width=w, height=h)
    self.buttons_numbers['.'].grid(row=3, column=2)

    frame_numbers.grid(row=1, column=0)
```



Adding commands

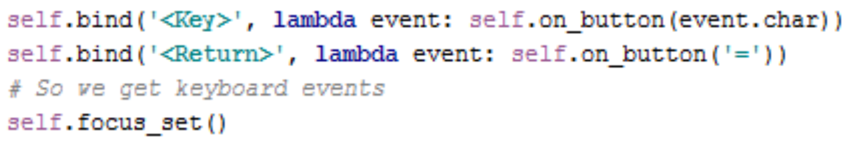
Python test!

Can you spot the subtle defect?



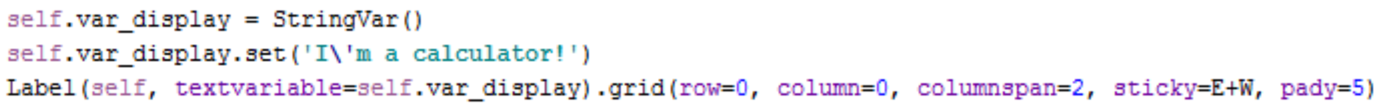
```
for i in range(3):
    for j in range(3):
        n = 3*i + j + 1
        self.buttons_numbers[n] = Button(frame_numbers,
                                         text='%i' % n,
                                         width=w, height=h,
                                         command=lambda: self.on_button(n)
        )
        self.buttons_numbers[n].grid(row=2-i, column=j)
```

Events



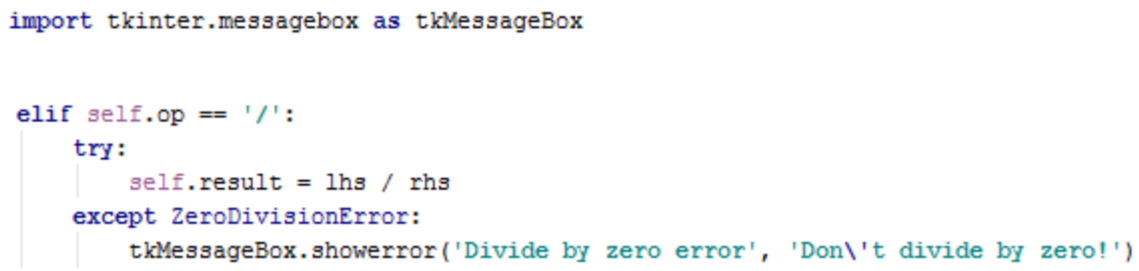
```
self.bind('<Key>', lambda event: self.on_button(event.char))
self.bind('<Return>', lambda event: self.on_button('='))
# So we get keyboard events
self.focus_set()
```

Interacting with GUI elements

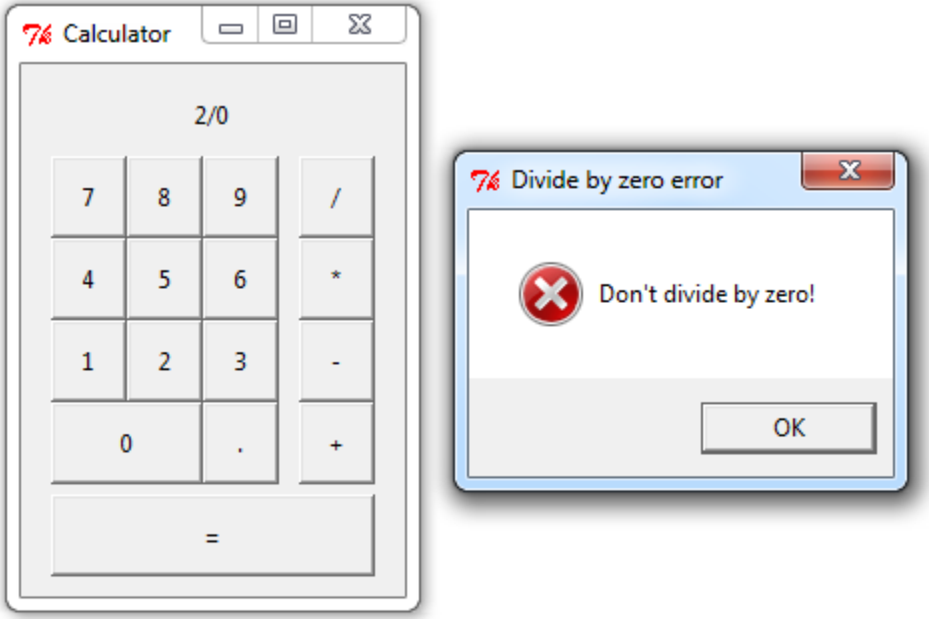


```
self.var_display = StringVar()
self.var_display.set('I\'m a calculator!')
Label(self, textvariable=self.var_display).grid(row=0, column=0, columnspan=2, sticky=E+W, pady=5)
```

Message boxes



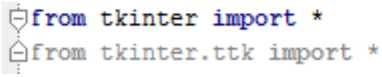
```
import tkinter.messagebox as tkMessageBox


elif self.op == '/':
    try:
        self.result = lhs / rhs
    except ZeroDivisionError:
        tkMessageBox.showerror('Divide by zero error', 'Don\'t divide by zero!')
```



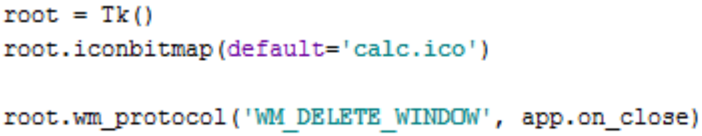
Divide by zero error

Don't divide by zero!

Other handy things



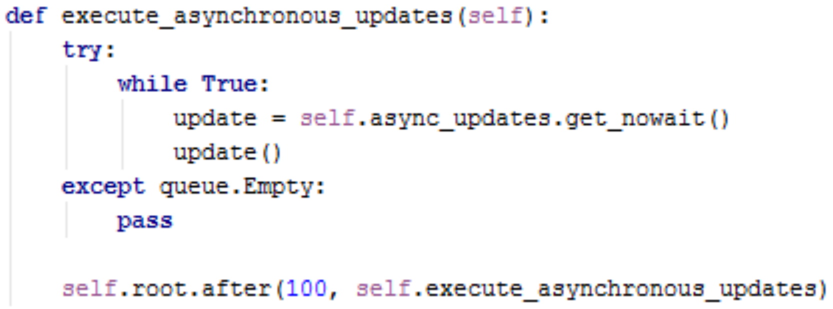
```
from tkinter import *
from tkinter.ttk import *
```



```
root = Tk()
root.iconbitmap(default='calc.ico')

root.wm_protocol('WM_DELETE_WINDOW', app.on_close)
```

A note about threads

- Only the thread that creates the GUI can modify the GUI



```python
def execute_asynchronous_updates(self):
    try:
        while True:
            update = self.async_updates.get_nowait()
            update()
    except queue.Empty:
        pass

    self.root.after(100, self.execute_asynchronous_updates)
```

References

- http://www.tkdocs.com/tutorial/index.html
- http://effbot.org/tkinterbook/