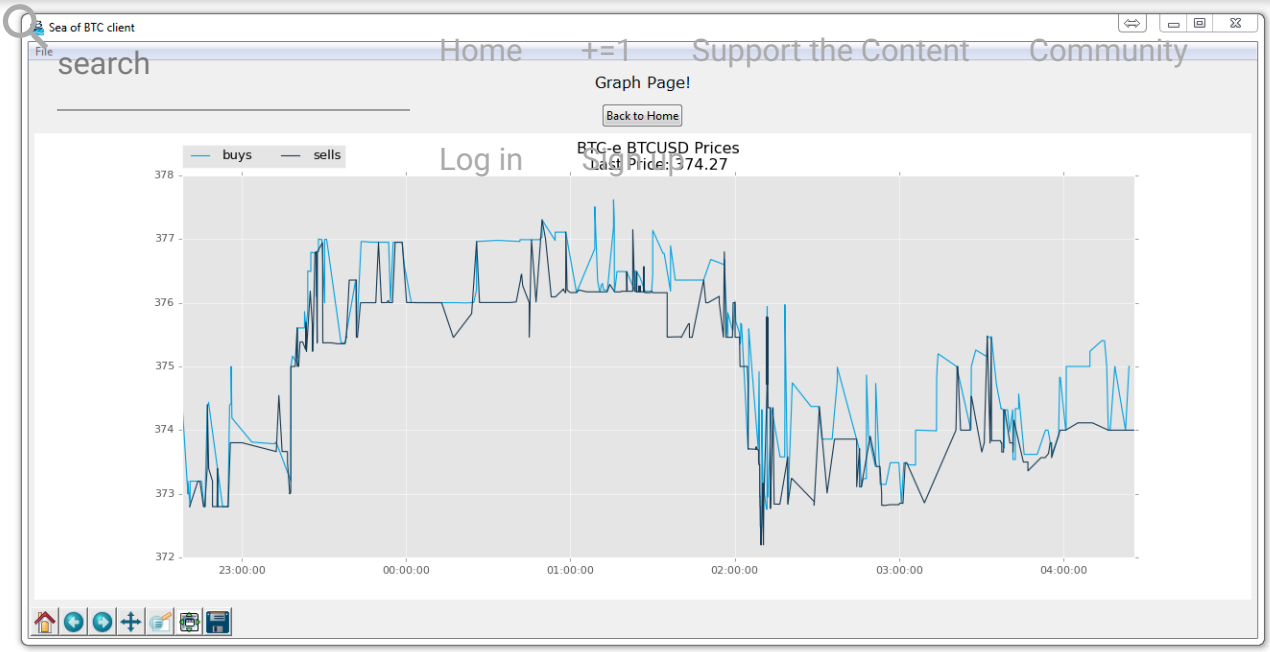
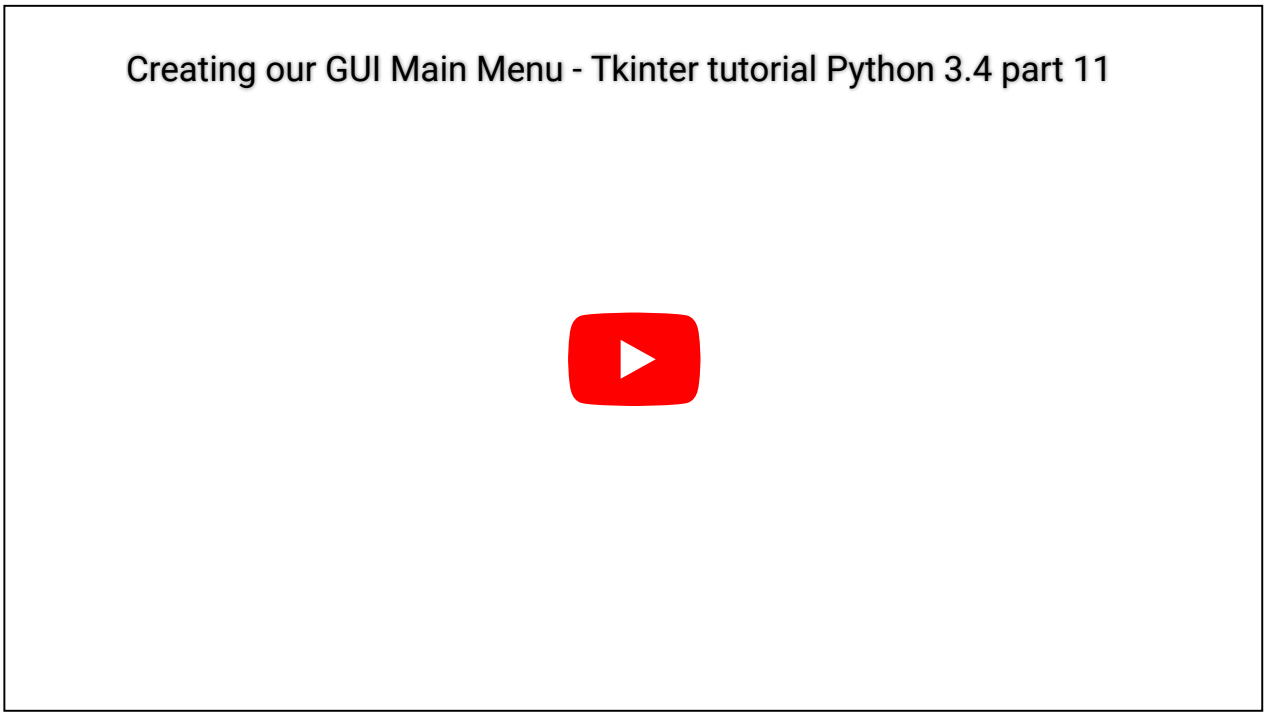


Creating our Main Menu in Tkinter



simple menu bar to act as our main menu.

To start off, however, we want to make a quick modification to how we're displaying our chart.

Now, we want to add another import:

```
from matplotlib import pyplot as plt
```

Then, instead of f=Figure, we want to do:

```
f = plt.figure()
```

Now that we have that out of the way, let's make that menu bar.

Within the SeaofBTCapp class, we want to add the following after our container:

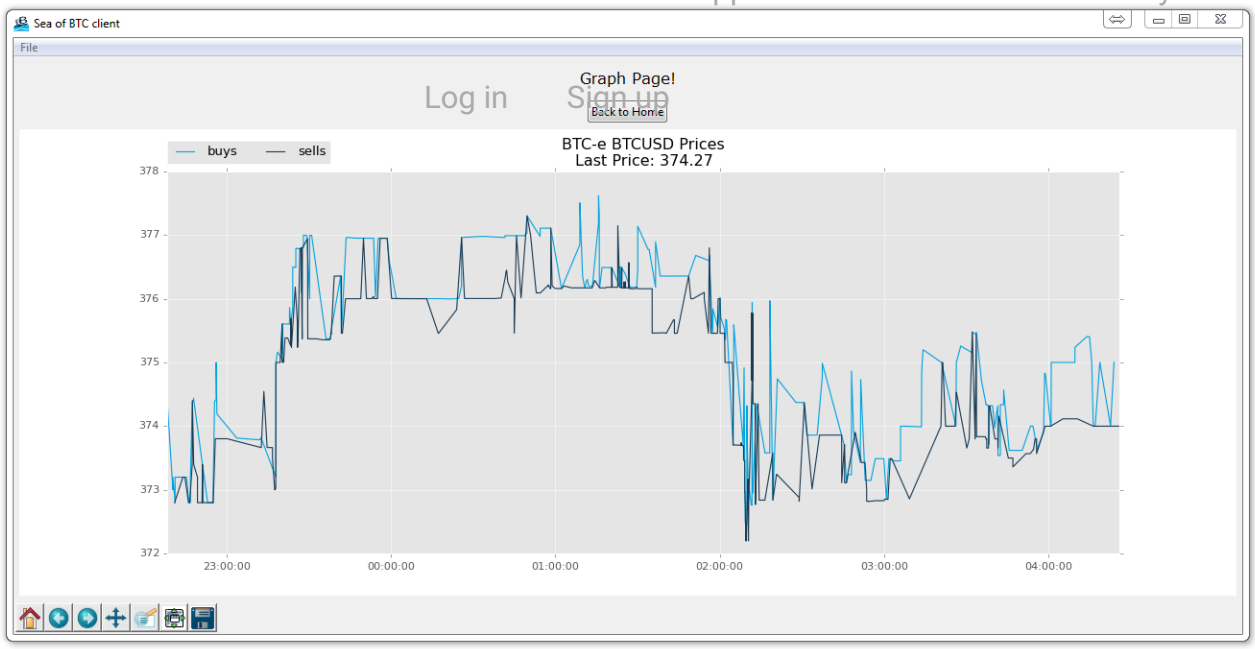
```
        menubar = tk.Menu(container)
        filemenu = tk.Menu(menubar, tearoff=0)
        filemenu.add_command(label="Save settings", command=lambda: popupms
        filemenu.add_separator()
        filemenu.add_command(label="Exit", command=quit)
        menubar.add_cascade(label="File", menu=filemenu)


        tk.Tk.config(self, menu=menubar)
```

Above, we're defining the menu bar, then we're adding a menu item (filemenu). From there, we build the filemenu items, then, when we're all done with all of that, we add this to our GUI.

Finally, at the end of our script, after we've defined app, we want to do:

The end result should be:



With that, your code should look like:

```
# The code for changing pages was derived from: http://stackoverflow.com/qu
# License: http://creativecommons.org/licenses/by-sa/3.0/

import matplotlib
matplotlib.use("TkAgg")
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg, Navigation
from matplotlib.figure import Figure
import matplotlib.animation as animation
from matplotlib import style

import tkinter as tk
from tkinter import ttk

import urllib
import json

import pandas as pd
import numpy as np

from matplotlib import pyplot as plt


LARGE_FONT= ("Verdana", 12)
style.use("ggplot")
```

```
a = f.add_subplot(111)

def animate(i):
    dataLink = 'https://btc-e.com/api/3/trades/btc_usd?limit=2000'
    data = urllib.request.urlopen(dataLink)
    data = data.readall().decode("utf-8")
    data = json.loads(data)


    data = data["btc_usd"]
    data = pd.DataFrame(data)

    buys = data[(data['type']=="bid")]
    buys["datestamp"] = np.array(buys["timestamp"]).astype("datetime64[s]")
    buyDates = (buys["datestamp"]).tolist()


    sells = data[(data['type']=="ask")]
    sells["datestamp"] = np.array(sells["timestamp"]).astype("datetime64[s]
    sellDates = (sells["datestamp"]).tolist()

    a.clear()

    a.plot_date(buyDates, buys["price"], "#00A3E0", label="buys")
    a.plot_date(sellDates, sells["price"], "#183A54", label="sells")

    a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3,
             ncol=2, borderaxespad=0)

    title = "BTC-e BTCUSD Prices\nLast Price: "+str(data["price"][1999])
    a.set_title(title)



class SeaofBTCapp(tk.Tk):

    def __init__(self, *args, **kwargs):

        tk.Tk.__init__(self, *args, **kwargs)

        tk.Tk.iconbitmap(self, default="clienticon.ico")
```


```
        container = tk.Frame(self)
        container.pack(side="top", fill="both", expand = True)
        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)


        menubar = tk.Menu(container)
        filemenu = tk.Menu(menubar, tearoff=0)
        filemenu.add_command(label="Save settings", command = lambda: popup
        filemenu.add_separator()
        filemenu.add_command(label="Exit", command=quit)
        menubar.add_cascade(label="File", menu=filemenu)

        tk.Tk.config(self, menu=menubar)

        self.frames = {}

        for F in (StartPage, BTCe_Page):

            frame = F(container, self)

            self.frames[F] = frame

            frame.grid(row=0, column=0, sticky="nsew")

        self.show_frame(StartPage)

    def show_frame(self, cont):

        frame = self.frames[cont]
        frame.tkraise()


class StartPage(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self,parent)
        label = tk.Label(self, text=("""ALPHA Bitcoin trading application
        use at your own risk. There is no promise
        of warranty."""), font=LARGE_FONT)
        label.pack(pady=10,padx=10)
```


```
                        command=lambda: controller.show_frame(BTCe_Page
        button1.pack()

        button2 = ttk.Button(self, text="Disagree",
                            command=quit)
        button2.pack()



class PageOne(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        label = tk.Label(self, text="Page One!!!", font=LARGE_FONT)
        label.pack(pady=10,padx=10)

        button1 = ttk.Button(self, text="Back to Home",
                            command=lambda: controller.show_frame(StartPage
        button1.pack()




class BTCe_Page(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        label = tk.Label(self, text="Graph Page!", font=LARGE_FONT)
        label.pack(pady=10,padx=10)

        button1 = ttk.Button(self, text="Back to Home",
                            command=lambda: controller.show_frame(StartPage
        button1.pack()







        canvas = FigureCanvasTkAgg(f, self)
        canvas.show()
        canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=Tr
```

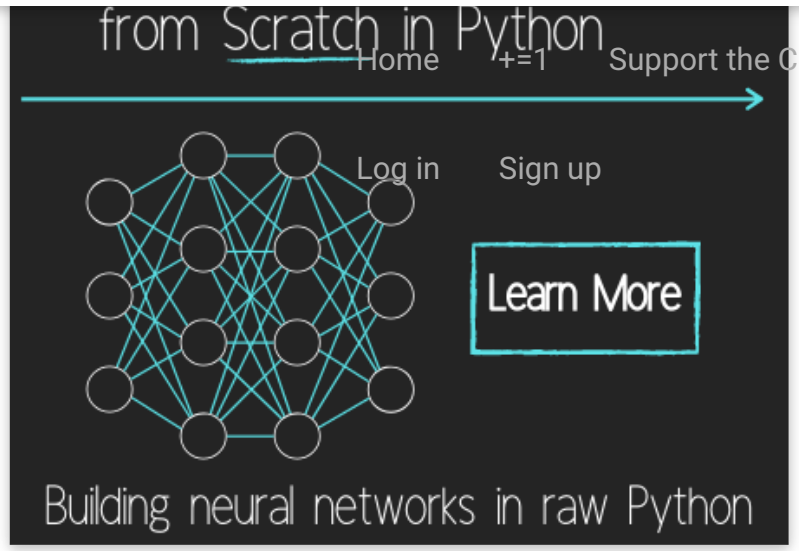
```
        canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
app = SeaofBTCapp()
app.geometry("1280x720")
ani = animation.FuncAnimation(f, animate, interval=5000)
app.mainloop()
```


NEXT →

Trending For You Today

NextArticle

The next tutorial: Building A Pop-Up Message Window



- Programming GUIs and windows with Tkinter and Python Introduction
- Object Oriented Programming Crash Course with Tkinter
- Passing functions with Parameters in Tkinter using Lambda
- How to change and show a new window in Tkinter
- Styling your GUI a bit using TTK
- How to embed a Matplotlib graph to your Tkinter GUI
- How to make the Matplotlib graph live in your application
- Organizing our GUI
- Plotting Live Updating Data in Matplotlib and our Tkinter GUI
- Customizing an embedded Matplotlib Graph in Tkinter

Creating our Main Menu in Tkinter

- Building a pop-up message window
- Exchange Choice Option
- Time-frame and sample size option

You've reached the end!

Contact: Harrison@pythonprogramming.net.

Programming is a superpower.