

Exchange Choice Option

Exchange choice option - Tkinter tutorial Python 3.4 part 13



Now that we have this main menu, let's allow the user to choose a new exchange. We want the user to be able to choose between multiple exchanges.

To start, we need some starting variables, so, at the top after defining our figure and subplot:



Sore Knees? Do This Once Daily

[Home](#)[+=1](#)[Support the Content](#)[Community](#)[Log in](#)[Sign up](#)

It takes less than 30 seconds (and you can do it right at home)...

[Watch Now](#)

```
exchange = "BTC-e"  
DatCounter = 9000  
programName = "btce"
```

We have "exchange," which is the "display name" of the exchange. This will be used in things like titles. Then we have DatCounter. Right now, you can just ignore this, but later we're going to be using this variable to dictate when a refresh of data is required. Finally, we have "programName" which is the name for the exchanges that we'll be using within our program as well as within the Sea of BTC API.

Next, we want to create a changeExchange function:

```
def changeExchange(toWhat,pn):  
    global exchange  
    global DatCounter  
    global programName  
  
    exchange = toWhat  
    programName = pn  
    DatCounter = 9000
```

This function will change the exchange "toWhat" which is the display exchange, and then the "pn" parameter is short for "program name."



The #1 Mistake That Makes Bad Knees Worse

[Home](#)
[Tutorials](#)
[Support the Content](#)
[Community](#)
[Log in](#)
[Sign up](#)

Knees hurt? Stop doing this...

[Learn more](#)

This function just globals these variables (so they can be accessed locally), and then assigns the new values to them.

Next, we want to add the options that use changeExchange to our main SeaofBTCapp class:

```
class SeaofBTCapp(tk.Tk):

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

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

        tk.Tk.iconbitmap(self, default="clienticon.ico")
        tk.Tk.wm_title(self, "Sea of BTC client")

        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)

        exchangeChoice = tk.Menu(menubar, tearoff=1)
        exchangeChoice.add_command(label="BTC-e",
                                   command=lambda: changeExchange("BTC-e", "
        exchangeChoice.add_command(label="Bitfinex",
                                   command=lambda: changeExchange("Bitfinex
```



```
exchangeChoice.add_command(label="Huobi",
                             command=lambda: ChangeExchange("Huobi", "
                             + "BTC", "USD"))

menubar.add_cascade(label="Exchange", menu=exchangeChoice)

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()
```



Sore Knees? Do This Once Daily

It takes...

Watch...

Note the major difference as the exchangeChoice definitions.

Right now, this option just changes the programs variables. Later, we will modify the animate function to take into account variable values to generate the dynamic data.



The full code up to this point for our Tkinter tutorial:

```
# 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)
NORM_FONT= ("Verdana", 10)
SMALL_FONT= ("Verdana", 8)

style.use("ggplot")
```



exchange = "BTC-e"

Home

+=1

Support the Content

Community

DatCounter = 9000

programName = "btce"

Log in

Sign up

```
def changeExchange(toWhat,pn):
```

```
    global exchange
```

```
    global DatCounter
```

```
    global programName
```

```
    exchange = toWhat
```

```
    programName = pn
```

```
    DatCounter = 9000
```

```
def popupmsg(msg):
```

```
    popup = tk.Tk()
```

```
    popup.wm_title("!")
```

```
    label = ttk.Label(popup, text=msg, font=NORM_FONT)
```

```
    label.pack(side="top", fill="x", pady=10)
```

```
    B1 = ttk.Button(popup, text="Okay", command = popup.destroy)
```

```
    B1.pack()
```

```
    popup.mainloop()
```

```
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")]
```


[a.clear\(\)](#)
[Home](#)
[+=1](#)
[Support the Content](#)
[Community](#)

```

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")
```

```
        tk.Tk.wm_title(self, "Sea of BTC client")
```

```
        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)
```



```

exchangeChoice.add_command(label="BTC-e",
                             command=lambda: changeExchange("BTC-e", "
exchangeChoice.add_command(label="Bitfinex",
                             command=lambda: changeExchange("Bitfinex
exchangeChoice.add_command(label="Bitstamp",
                             command=lambda: changeExchange("Bitstamp
exchangeChoice.add_command(label="Huobi",
                             command=lambda: changeExchange("Huobi", "

menubar.add_cascade(label="Exchange", menu=exchangeChoice)

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)

        button1 = ttk.Button(self, text="Agree",
                              command=lambda: controller.show_frame(BTCe_Page

```




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

```
button2.pack()
```

Home Support the Content

Community

Log in Sign up

```
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=True)
```

```
        toolbar = NavigationToolbar2TkAgg(canvas, self)
```

```
        toolbar.update()
```

```
        canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
```



```
app = SeaofBTCapp()
```

[Home](#)[+=1](#)[Support the Content](#)[Community](#)

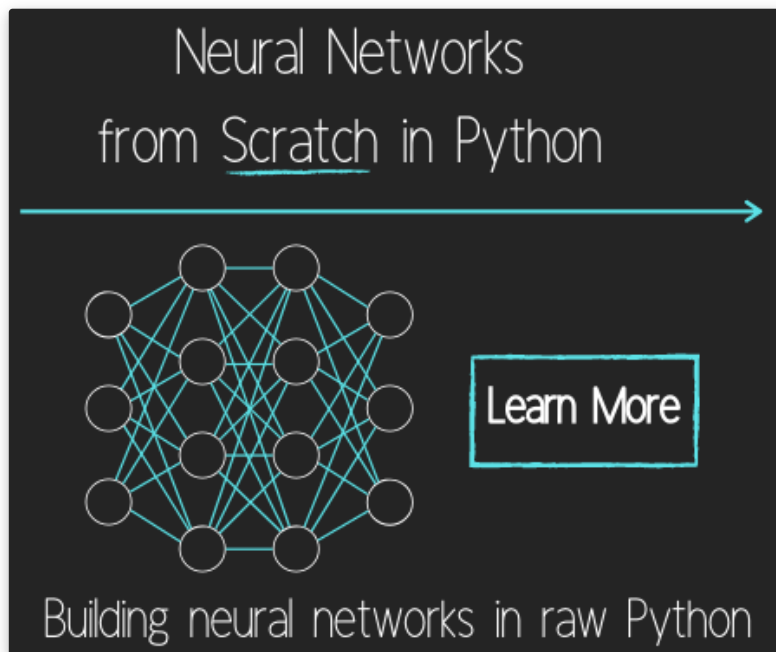
```
app.geometry("1280x720")
```

```
ani = animation.FuncAnimation(f, animate, interval=5000)
```

```
app.mainloop()
```

[Log in](#)[Sign up](#)

The next tutorial:

[Time-Frame And Sample Size Option](#)[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](#)



Organizing our GUI	Home	+ = 1	Support the Content	Community
Plotting Live Updating Data in Matplotlib and our Tkinter GUI	Log in	Sign up		
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				
Adding indicator Menus (3 videos)				
Trading option, start/stop, and help menu options				
Tutorial on adding a tutorial				
Allowing the exchange choice option to affect actual shown exchange				
Adding exchange choice cont'd				
Adding exchange choices part 3				
Indicator Support				
Pulling data from the Sea of BTC API				
Setting up sub plots within our Tkinter GUI				
Graphing an OHLC candlestick graph embedded in our Tkinter GUI				
Acquiring RSI data from Sea of BTC API				
Acquiring MACD data from Sea of BTC API				

Converting Tkinter application to .exe and installer with cx_Freeze

[Home](#)[+=1](#)[Support the Content](#)[Community](#)[Log in](#)[Sign up](#)

New York: Dentists Are Aln Giving Away Implants Now Offers)

**Perform A Simple Search On The Ne
View Dental Implant Prices and Offe**

sponsored by: Dental Implant Ads

[LE](#)

You've reached the end!

Contact: Harrison@pythonprogramming.net.

Support this Website!

Consulting and Contracting

Facebook

Twitter

Instagram

Legal stuff:

[Terms and Conditions](#)

[Privacy Policy](#)

© OVER 9000! [PythonProgramming.net](https://pythonprogramming.net)

Programming is a superpower.