



search

Home

+=1

Support the Content

Community

Log in

Sign up

Organizing our GUI

Organizing our GUI window - Tkinter tutorial Python 3.4 p. 8



Now we're ready to begin moving towards our goal some more, but first we need to do a few things to prepare for this.

First, we're going to add some new imports to be used soon:

Why you're not losing weight

WAKE UP TIME	6 AM	7 AM	8 AM	AGE	18-25	26-35	36-	
DAILY MEALS	1	2	3	4+	DAILY WATER INTAKE			
HOURS OF SLEEP	<input type="range" value="7"/>			YOUR BMI	40+	30+	25-	
AGE	18-25	26-35	36-55	56+	FASTING SCHEDULE	16:8	12:12	14:



[import pandas as pd](#)
[Home](#)
[+=1](#)
[Support the Content](#)
[Community](#)

[import numpy as np](#)

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

This means you're going to need [pandas](#) and [numpy](#) if you do not already have them.

You can also use pip to install using something like: `pip install pandas` in cmd.exe / bash.

If you need help with pip, check out the [Pip Tutorial](#).

Next, we need to change the following in our SeaofBTCapp class:

Why you're not losing weight

WAKE UP TIME	6 AM	7 AM	8 AM	AGE	18-25	26-35	36-
DAILY MEALS	1	2	3	DAILY WATER INTAKE			
HOURS OF SLEEP	<input type="range" value="7"/>			YOUR BMI	40+	30+	25-
AGE	18-25	26-35	36-55	FASTING SCHEDULE	16:8	12:12	14:

```

self.frames = {}

for F in (StartPage, PageOne, PageTwo, PageThree):

    frame = F(container, self)

    self.frames[F] = frame

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

self.show_frame(StartPage)

```

to

```

self.frames = {}

for F in (StartPage, BTCE_Page):

    frame = F(container, self)

```



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

```
self.show_frame(StartPage)
```

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

Then we're going to slightly modify our StartPage class:

Why you're not losing weight

WAKE UP TIME	6 AM	7 AM	8 AM	AGE	18-25	26-35	36+
DAILY MEALS	1	2	3	DAILY WATER INTAKE			
HOURS OF SLEEP	<input type="range" value="7"/>			YOUR BMI	40+	30+	25+
AGE	18-25	26-35	36-55	FASTING SCHEDULE	16:8	12:12	14:

```
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))
        button1.pack()

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



WAKE UP TIME	6 AM	7 AM	8 AM	AGE	18-25	26-35	36-	
DAILY MEALS	1	2	3	DAILY WATER INTAKE				
HOURS OF SLEEP	<input type="text" value="7"/>			YOUR BMI	40+	30+	25-	
AGE	18-25	26-35	36-55	56+	FASTING SCHEDULE	16:8	12:12	14:

Notice mainly the new label, but also the button changes.

Why you're not losing weight

WAKE UP TIME	6 AM	7 AM	8 AM	AGE	18-25	26-35	36-	
DAILY MEALS	1	2	3	4+	DAILY WATER INTAKE			
HOURS OF SLEEP	<input type="text" value="7"/>			YOUR BMI	40+	30+	25-	
AGE	18-25	26-35	36-55	56+	FASTING SCHEDULE	16:8	12:12	14:

Now we're going to convert PageThree a bit, and re-name it to BTCE_Page:

```
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)
```



Home

+=1

Support the Content

Community

After all of this, your full code should be:

*# 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

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

f = Figure(figsize=(5,5), dpi=100)
a = f.add_subplot(111)

def animate(i):
    pullData = open("sampleData.txt","r").read()
    dataList = pullData.split('\n')
    xList = []
    yList = []
    for eachLine in dataList:
        if len(eachLine) > 1:
            x, y = eachLine.split(',')
            xList.append(int(x))
            yList.append(int(y))

    a.clear()
```



class SeaofBTCapp(tk.Tk): Home +=1 Support the Content Community

```
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)

    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() Home +=1 Support the Content Community

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**)

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

toolbar.update()

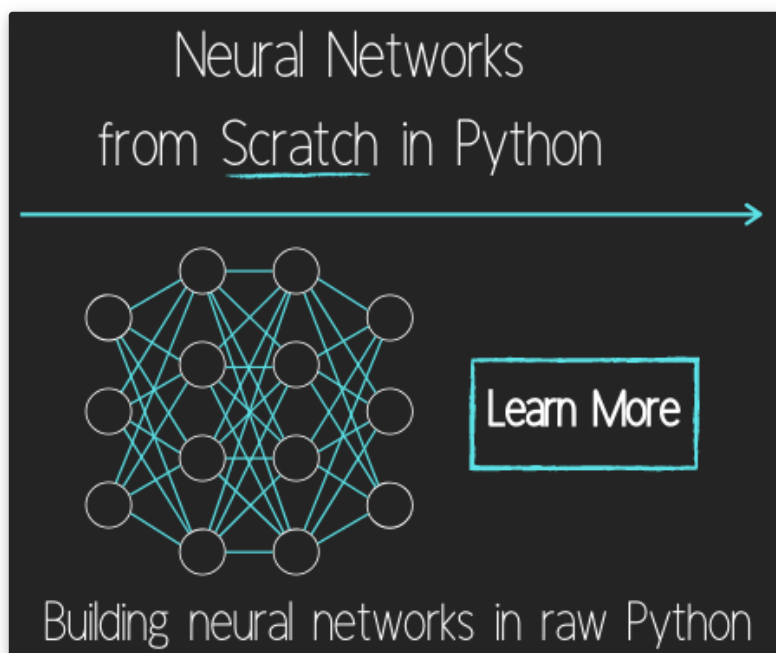
canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=**True**)



```
ani = animation.FuncAnimation(t, animate, interval=1000)  
app.mainloop()
```

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

The next tutorial:

[Plotting Live Updating Data In Matplotlib And Our Tkinter GUI](#)[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](#)



Plotting Live Updating Data in Matplotlib and our Tkinter GUI	Home	+1	Support the Content	Community
Customizing an embedded Matplotlib Graph in Tkinter	Login	Sign-up		
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](#)[Visit The adidas® Store](#)

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.