

Passing functions v Parameters in Tkint using Lambda

Functions with Parameters with Tkinter GUI buttons - Tkinter tutorial Pyt...



If you have been using Tkinter much, you've likely found yourself stumped when it comes time to pass parameters through your functions via the command option. You can very easily pass a function without parameters, but adding parameters seems impossible.

Fear not, I have a solution for you!



def qf(quickPrint):
 print(quickPrint)

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

So this is just a super simple function that will print whatever we pass through it. So it's a print function that prints using the print function!

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

Next, here's our new StartPage, with an added button:

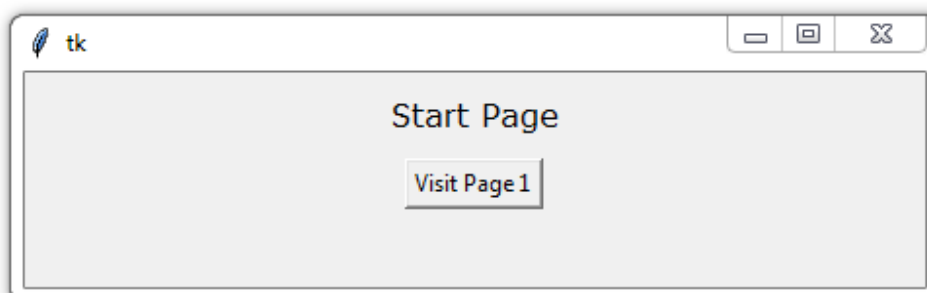


```
class StartPage(tk.Frame):

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

        button = tk.Button(self, text="Visit Page 1",
                           command=lambda: qf("Check me out, I'm passing v
        button.pack()
```

Command is where you can pass functions, but here we're going to pass a lambda function instead, which creates a quick throwaway function, using the parameters we've set. Here's what you should get:





Home +=1 Support the Content Community

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:

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

```
import tkinter as tk
```

```
LARGE_FONT= ("Verdana", 12)
```

```
class SeaofBTCapp(tk.Tk):
```

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

```
        tk.Tk.__init__(self, *args, **kwargs)
        container = tk.Frame(self)
```

```
        container.pack(side="top", fill="both", expand = True)
```

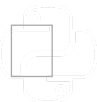
```
        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)
```

```
        self.frames = {}
```

```
        frame = StartPage(container, self)
```

```
        self.frames[StartPage] = frame
```

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



```
def show_frame(self, cont):
```

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

```
    frame = self.frames[cont]
```

```
    frame.tkraise()
```

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

```
def qf(quickPrint):
```

```
    print(quickPrint)
```

```
class StartPage(tk.Frame):
```

```
    def __init__(self, parent, controller):
```

```
        tk.Frame.__init__(self, parent)
```

```
        label = tk.Label(self, text="Start Page", font=LARGE_FONT)
```

```
        label.pack(pady=10, padx=10)
```

```
        button = tk.Button(self, text="Visit Page 1",
```

```
                           command=lambda: qf("Check me out, I'm passing v
```

```
        button.pack()
```

```
app = SeaofBTCapp()
```

```
app.mainloop()
```



The downside? This is a throwaway function. As we'll see later on, this can be a bit of a pain. For example, if we pass some data through this function like maybe the button's relationship to some other data, like an ID for the button or something, it can just go missing and not act as we had hoped. This is problematic at times, but it can also be resolved. There's always a way!

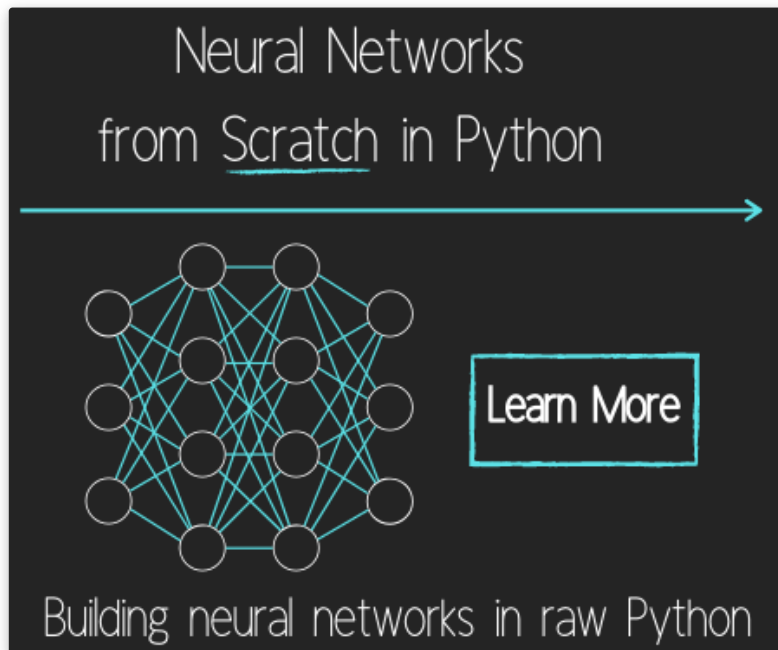


raised in our application.

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

The next tutorial:

[How To Change And Show A New Window In Tkinter](#)

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

[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](#)



Creating our Main Menu in Tkinter

Home

+=1

Support the Content

Community

Building a pop-up message window

Log in

Sign up

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](#)

Find Directions & Visit Today

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](#)

