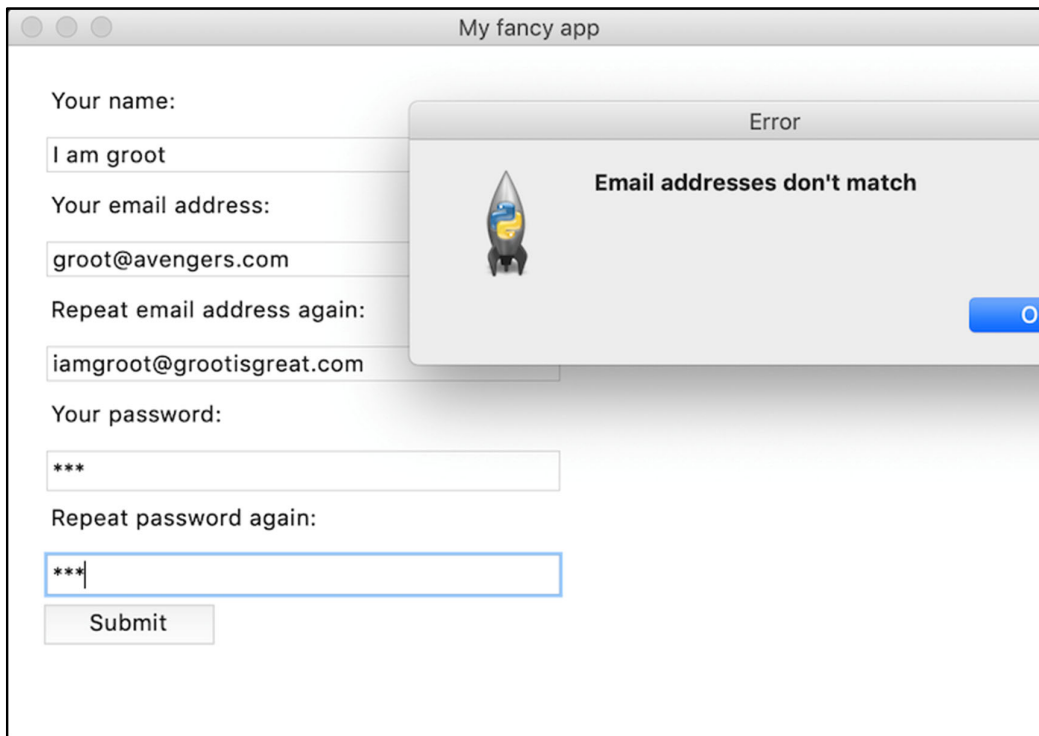


Example email and password verification tool

This example uses the Tkinter library which allows you to create graphical user interfaces. Writing Python GUI's are not part of the course syllabus but they are useful and likely something you are interested in knowing how to do.

If it is something you wish learn more about, I do have some notes on my Python website, called "GUI with TKinter". <https://pbaumgarten.com/python/> - I'm happy to create additional resources for this if you are interested please let me know.



I've written this demo in line with best practices for Tkinter. You will see a lot of different ways of using it online. It is worth learning how to do it the right way if it is something you are keen to use. Two key points to draw your attention to:

- Protecting your namespace

```
import tkinter as tk
```

You will see many tutorials online use `from tkinter import *`

This has the effect of "polluting" your "namespace". By that, I mean it imports all the tkinter variables and functions into your Python code increasing the risk you will accidentally overwrite a tkinter function. By importing "as tk" means you use the "tk." prefix to use any tk functionality but it also protects it from accidents and ensures you understand exactly when you are depending on tkinter.

- Using an object orientated approach

While it is possible to build Tkinter apps without classes, it is generally only suitable for very simple/basic programs. As soon as you want to build anything more complex, especially a program with multiple windows, objects become key. Without the use of objects you would need a lot of global variables which is generally considered to be poor programming practice. Take the time to learn object orientated programming if you are going to build GUIs. It is not optional in any other GUI language system (eg: Java for Android, Swift for iOS).

```

import tkinter as tk
from tkinter import messagebox

class VerificationApp():
    # The __init__() function is automatically executed when VerificationApp() is created
    def __init__(self, parent):
        # Save a variable that keeps track of our containing object
        self.parent = parent
        # Create a window
        self.window = tk.Toplevel()
        self.window.title("Verification app")
        self.window.geometry("600x400")
        # Create the labels and text entry boxes
        self.name_label = tk.Label(self.window, text="Your name:")
        self.name_label.place(x=20, y=20)
        self.name_text = tk.Entry(self.window)
        self.name_text.place(x=20, y=50, width=300, height=25)
        self.name_text.focus()
        self.email1_label = tk.Label(self.window, text="Your email address:")
        self.email1_label.place(x=20, y=80)
        self.email1_text = tk.Entry(self.window)
        self.email1_text.place(x=20, y=110, width=300, height=25)
        self.email2_label = tk.Label(self.window, text="Repeat email address again:")
        self.email2_label.place(x=20, y=140)
        self.email2_text = tk.Entry(self.window, text="")
        self.email2_text.place(x=20, y=170, width=300, height=25)
        self.password1_label = tk.Label(self.window, text="Your password:")
        self.password1_label.place(x=20, y=200)
        self.password1_text = tk.Entry(self.window, text="", show="*")
        self.password1_text.place(x=20, y=230, width=300)
        self.password2_label = tk.Label(self.window, text="Repeat password again:")
        self.password2_label.place(x=20, y=260)
        self.password2_text = tk.Entry(self.window, text="", show="*")
        self.password2_text.place(x=20, y=290, width=300)
        # Create a button to execute the self.clicked() function when clicked
        self.submit_button = tk.Button(self.window, text="Submit", command=self.clicked )
        self.submit_button.place(x=20, y=320, width=100, height=25)

    def clicked(self):
        ok = True
        # Get text strings entered in the text boxes
        email1 = self.email1_text.get()
        email2 = self.email2_text.get()
        pw1 = self.password1_text.get()
        pw2 = self.password2_text.get()
        # Run checks on the email addresses provided
        if "@" not in email1:
            self.email1_text["bg"] = "#ffa8a8"
            messagebox.showerror("Error", "That is not an email address")
            ok = False
        elif email1 != email2:
            self.email1_text["bg"] = "#ffa8a8"
            self.email2_text["bg"] = "#ffa8a8"
            messagebox.showerror("Error", "Email addresses don't match")

```

```

        ok = False
    else:
        self.email1_text["bg"] = "white"
        self.email2_text["bg"] = "white"
    # Run checks on the passwords provided
    if len(pw1) == 0:
        self.password1_text["bg"] = "#ffa8a8"
        messagebox.showerror("Error", "You must enter a password")
        ok = False
    elif pw1 != pw2:
        self.password1_text["bg"] = "#ffa8a8"
        self.password2_text["bg"] = "#ffa8a8"
        messagebox.showerror("Error", "Passwords don't match")
        ok = False
    else:
        self.password1_text["bg"] = "white"
        self.password2_text["bg"] = "white"
    # Exit program if all checks have passed
    if ok:
        messagebox.showinfo("Good job", "Well done, bye for now")
        self.parent.quit() # Terminate the program
    # End of clicked()
# End of VerificationApp()

# *** MAIN PROGRAM STARTS HERE ***

root = tk.Tk() # Create the Tk environment
root.withdraw() # Remove the default window
app = VerificationApp(root) # Create an instance of our verification app
root.mainloop() # Execute

```