

tkinter

links

Basics: <https://likegeeks.com/python-gui-examples-tkinter-tutorial/>

Layout: <https://zetcode.com/tkinter/layout/>

grid: <https://blog.tecladocode.com/tkinters-grid-geometry-manager/>

Mixed layout: <https://stackoverflow.com/a/38154650/15030988>

Radiobutton Gruppen: <https://riptutorial.com/tkinter/example/30286/create-a-group-of-radiobuttons>
<https://stackoverflow.com/a/47162318/15030988>

Buttons mit gemeinsamer click-Funktion: <https://stackoverflow.com/a/6921225/15030988>

MessageBox: <https://docs.python.org/3/library/tkinter.messagebox.html>

Neues Fenster:
<https://www.delftstack.com/de/howto/python-tkinter/how-to-create-a-new-window-with-a-button-in-tkinter/>

Bildschirmgröße

- `winfo_screenwidth()`
- `winfo_screenheight()`

Fenster in Bildschirmmitte ausrichten: <https://m.youtube.com/watch?v=gjU3Lx8XMS8>

- $X = (\text{screenwidth} - \text{windowwidth}) / 2$
- $Y = (\text{screenheight} - \text{windowheight}) / 2$

die Position kann wie die Größe mit der geometry Methode gesetzt werden

```
root = Tk()  
root.geometry("300x200+300+300") #WxH+X+Y
```

es kann auch nur die position verändert werden:

```
root.geometry("+10+10")
```

Fenster als Klasse

```
import tkinter as tk
```

```
class App:
    def __init__(self, window):
        window.geometry('350x200')
        frame = tk.Frame(window)
        frame.pack()
        self.button = tk.Button(frame,
                                text="QUIT", fg="red",
                                command=frame.quit)

        self.button.pack(side=tk.LEFT)
        self.slogan = tk.Button(frame,
                                text="Hello",
                                command=self.write_slogan)
        self.slogan.pack(side=tk.LEFT)
    def write_slogan(self):
        print("Tkinter is easy to use!")

if __name__ == "__main__":
    root = tk.Tk()
    app = App(root)
    root.mainloop()
```

alternativ kann man auch direkt von der tk.Tk-Klasse ableiten, somit muss das Fenster nicht extern erzeugt werden

```
class App(tk.Tk):
    def __init__(self):
        tk.Tk.__init__(self)
        self.geometry('350x200')
        frame = tk.Frame(self)
        ...
```

Das sollte aber nur bei Anwendungen gemacht werden die nur 1 Fenster nutzen. Bei mehreren Fenstern sollte die Fenstererstellung außerhalb der Klasse und nur das Füllen des Fensters innerhalb ablaufen.

Texteingabe

hier wird ein neuer Frame erzeugt um die Scrollbars an der Textbox auszurichten

ggf. mainframe (Eltern-Element) und txt_frm.grid anpassen (ggf. pack o.ä.)

```
import tkinter as tk

class scrolltext(tk.Frame):
    def __init__(self, parent, width, height):
        tk.Frame.__init__(self, parent, width=width, height=height)
        # ensure a consistent GUI size
        self.grid_propagate(False)
        # implement stretchability
```

```

self.grid_rowconfigure(0, weight=1)
self.grid_columnconfigure(0, weight=1)

self.txt = tk.Text(self, borderwidth=3, relief="sunken")
self.txt.config(undo=True, wrap='none')#word
self.txt.grid(row=0, column=0, sticky="nsew", padx=2, pady=2)

#create vertical scrollbar
self.vscrollb = tk.Scrollbar(self, command=self.txt.yview)
self.vscrollb.grid(row=0, column=1, sticky='nsew')
self.txt['yscrollcommand'] = self.vscrollb.set

#create horizontal scrollbar
self.hscrollb = tk.Scrollbar(self,
orient='horizontal',command=self.txt.xview)
self.hscrollb.grid(row=1, column=0, sticky='nsew')
self.txt['xscrollcommand'] = self.hscrollb.set

def focus(self):
    self.txt.focus()
def getText(self):
    return self.txt.get("1.0", tk.END)
def setText(self, text):
    self.txt.delete(1.0, "end")
    self.txt.insert(1.0, text)

if __name__ == "__main__":
    def clicked():
        print(txt.getText().splitlines())
        txt.setText("this is a special text")

    window = tk.Tk()
    window.geometry('350x200')
    window.title("Welcome to LikeGeeks app")

    txt=scrolltext(window, width=250,height=200)
    txt.pack(side = tk.LEFT,fill=tk.BOTH,expand=1)

    btn = tk.Button(window, text="Click Me", command=clicked)
    btn.pack(side=tk.RIGHT)

    txt.focus()

    window.mainloop()

```

<https://stackoverflow.com/a/9662139/15030988>

Beispiel mit yscrollcommand und zeilenweisem auslesen:

<https://stackoverflow.com/a/17747230/15030988> Horizontale scrollbar:

<https://stackoverflow.com/a/47954818/15030988>

treeview

<https://riptutorial.com/tkinter/example/31880/treeview--basic-example>

<https://www.askpython.com/python-modules/tkinter/tkinter-treeview-widget>

PanedWindow

ein PanedWindow ist ein Container für mindestens 2 Widgets mit einer verschiebbaren Begrenzung.

```
m1 = tk.PanedWindow(window)
m1.pack(fill = tk.BOTH, expand = 1)

button = tk.Button(m1, text="QUIT", fg="red")
m1.add(button, stretch="always") #always richtet automatisch aus, danach sind alle Felder gleich groß

button2 = tk.Button(m1, text="Hello")
m1.add(button2, stretch="always")

#setzen des Schiebereglers (optional, falls always nicht richtig ist)
m1.update()
m1.sash_place(0, 400, 0) #x-position 400 des ersten Widgets
```

<https://stackoverflow.com/a/62032595/15030988>

<https://stackoverflow.com/a/27389064/15030988>

Notebook

```
tc = ttk.Notebook(master)
t1 = ttk.Frame(tc)
t2 = ttk.Frame(tc)
tc.add(t1, text='Notebook tab1')
tc.add(t2, text='Notebook tab2')
tc.pack(expand = 1, fill="both")
```

<https://www.educba.com/tkinter-notebook/>

Beispiel-Gui

```
from tkinter import Tk, RIGHT, BOTH, RAISED, X, Y, IntVar, BooleanVar, StringVar
from tkinter import messagebox
from tkinter.ttk import Label, Frame, Button, Entry,
```

Checkbutton, Radiobutton, Notebook, Treeview, Style

```
class Example():

    def __init__(self, window):
        self.window = window
        self.initUI(window)

    def initUI(self, window):
        self.tc = Notebook(window)
        self.t1 = Frame(self.tc)
        self.t2 = Frame(self.tc)
        self.t3 = Frame(self.tc)
        self.tc.add(self.t1, text='Tab1')
        self.tc.add(self.t2, text='Tab2')
        self.tc.add(self.t3, text='Tab3')
        self.tc.pack(fill=BOTH, expand=True)

        closeButton = Button(window, text="Close", command=window.destroy)
        closeButton.pack(side=RIGHT, padx=5, pady=5)
        okButton = Button(window,
text="OK", command=lambda: messagebox.showinfo("OK pressed", "Hello, you have
successful pressed the OK-Button!\n\nEntry: %s CHK:%d RD:%s" %
(self.e1.get(), self.chkvar1.get(), self.radvar1.get())))
        okButton.pack(side=RIGHT)
        mvButton =
Button(window, text="move", command=lambda: window.geometry("+10+10"))
        mvButton.pack(side=RIGHT)

        self.filltabs()

    def filltabs(self):
        l1 = Label(self.t1, text="Entry:")
        l1.grid(row=0, column=0)
        self.e1 = Entry(self.t1)
        self.e1.grid(row=0, column=1)
        l2 = Label(self.t1, text="Checkbutton:")
        l2.grid(row=1, column=0)
        self.chkvar1 = IntVar()
        self.chkvar1.set(1)
        c1 = Checkbutton(self.t1, text='Choose', variable=self.chkvar1) #,
onvalue=1, offvalue=0)
        c1.grid(row=1, column=1, sticky="W") #left justify

        l3 = Label(self.t2, text="Radiobuttons")
        l3.grid(row=0, column=0, sticky="N")
        rf = Frame(self.t2)
        rf.grid(row=0, column=1)
        self.radvar1 = StringVar()
        r1 = Radiobutton(rf, text="Python", variable=self.radvar1,
value="python")
```

```
r1.pack(anchor="w") #untereinander linksbündig
r2=Radiobutton(rf, text="PHP", variable=self.radvar1, value="php")
r2.pack(anchor="w")

self.tree1=Treeview(self.t3, columns=("Name", "Data")) # causes
WIndow be bigger and hides buttonframe
self.tree1.pack(fill=BOTH, expand=True)

self.tree1.insert(parent='', index='end', iid=0, text="root",
values=("First Col", "Second Col"))
self.tree1.insert(parent=0, index='end', iid=1, text="child",
values=("Hello", "Again"))

def main():

    root = Tk()
    root.geometry("400x300+300+300")
    root.title("testwindow")
    root.style = Style().theme_use("alt") # to show checkmarks in
checkboxbutton and round radiobuttons
    app = Example(root)
    root.mainloop()

if __name__ == '__main__':
    main()
```

From:

<http://www.fw-web.de/dokuwiki/> - **FW-WEB Wiki**

Permanent link:

<http://www.fw-web.de/dokuwiki/doku.php?id=programming:python:tkinter>

Last update: **2023/06/08 17:06**

