

# guizero

Marco Bakera ([bakera@tbs1.de](mailto:bakera@tbs1.de))

guizero<sup>1</sup> ermöglicht die einfache Gestaltung von graphischen Benutzeroberflächen. Es kann mit pip installiert werden.

Auf der Webseite wird eine alternative Möglichkeit der Installation vorgestellt, die mit einem Download auskommt.

Die folgende Beispielanwendungen zeigen verschiedene GUI-Elemente in Aktion. Der Quelltext muss ausgeführt werden, um das Fenster zu sehen.

Zunächst ein leeres Fenster ohne Inhalt.

---

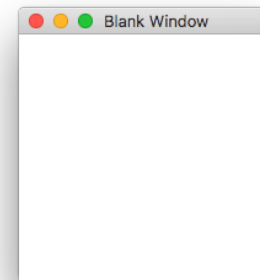
```
1 import guizero
2
3 class BlankWindow:
4     def __init__(self, title="Blank Window"):
5         self.root = guizero.App(title=title)
6
7     def run(self):
8         self.root.display()
9
10 w = BlankWindow()
11 w.run()
```

---

<sup>1</sup> <https://lawsie.github.io/guizero>

Windows: pip install guizero

Linux/Mac: pip3 install guizero

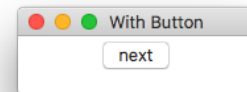


Nun ein Fenster mit einem Button. Die Methode `click` wird bei einem Klick auf den Button ausgeführt.

---

```
1 class WindowWithButton(BlankWindow):
2     def __init__(self):
3         super().__init__("With Button")
4         # add button, invoking method 'click' when clicked.
5         btn = guizero.PushButton(self.root, text="next",
6                                   command=self.click)
7
8     def click(self):
9         print("Button clicked!")
10
11
12 w = WindowWithButton()
13 w.run()
```

---



Ausgabe:

Button clicked!

Button clicked!

Zwei Buttons können in einem Grid-Layout<sup>2</sup> angeordnet werden.

<sup>2</sup> <https://lawsie.github.io/guizero/layout/>

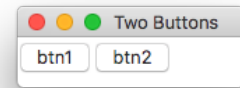
---

```

1 class WindowWithTwoButtons:
2     def __init__(self):
3         self.root = guizero.App(title='Two Buttons',
4                                   layout='grid')
5         btn1 = guizero.PushButton(self.root, text="btn1",
6                                   grid=[0,0])
7         btn2 = guizero.PushButton(self.root, text="btn2",
8                                   grid=[1,0])
9     def run(self):
10        self.root.display()
11
12 w = WindowWithTwoButtons()
13 w.run()

```

---



Die Klasse `Picture` kann genutzt werden, um Bilder anzuzeigen.<sup>3</sup>  
Hierfür wird ein Bild mit einem Ball verwendet.

<sup>3</sup> <https://lawsie.github.io/guizero/images/>

---

```

1 class WindowWithLabel(BlankWindow):
2     def __init__(self):
3         super().__init__("With Label")
4         # add image in a Picture-Object
5         self.bild = guizero.Picture(self.root,
6                                     image="ball.gif")
7
8 w = WindowWithLabel()
9 w.run()

```

---



Nun ein komplexeres Beispiel mit einem Menü, Button und Label.

---

```

1 class GUI:
2     def __init__(self):
3         self.counter = 0
4
5         root = guizero.App(title='TKinter Demo', layout='grid')
6
7         guizero.MenuBar(root,
8                         toplevel=['Tools'],
9                         options=[
10                            [ ['Next', self.click] ]
11                        ])
12         # add image
13         bild = guizero.Picture(root, image="ball.gif",
14                                grid=[0,0])
15
16         # add button, invoking method 'click' when clicked.
17         btn = guizero.PushButton(root, text="next",
18                                   command=self.click,
19                                   grid=[0,1])
20
21         # add label with counter value
22         self.lbl = guizero.Text(root,
23                                  text="Label für Counter",
24                                  grid=[1,1])
25
26         # add entry field
27         self.ent = guizero.TextBox(root, grid=[2,1])
28
29         # entering main event loop
30         root.display()
31
32     def click(self):
33         # update label and counter
34         self.lbl.value = "Counter %s" % self.counter
35         self.counter += 1
36
37
38 GUI()

```

---

