

<https://www.halvorsen.blog>



Python GUI Development using guizero

Hans-Petter Halvorsen

Table of Contents

- [Introduction to GUI Applications in Python](#)
- [Introduction to guizero](#)
- [guizero Widgets Examples](#)
 - [App](#) (the foundation for all guizero applications)
 - [Text](#) widget
 - [TextBox](#) widget
 - [PushButton](#) widget
 - [MessageBox](#)
- [Event Handling](#)
- [Timers](#) (repeat)
- [Plotting Examples](#)
 - Here we use guizero in combination with the matplotlib



Introduction

GUI Applications in Python

Hans-Petter Halvorsen

[Table of Contents](#)

Introduction

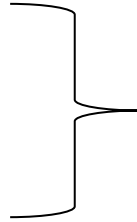
- GUI – Graphical User Interface
- In Programming Environment like Visual Studio, LabVIEW, etc. you can easily create GUI Applications because it is built into the IDE (Integrated Development Environment), and you can use the built-in Graphical Designer Tools
- There are no built-in functionality for creating GUI Applications in Python
- You need to install and use an external GUI Python Library, and there are typically no Designer, you need to create the GUI entirely in Code
- This Tutorial will focus on **guizero**

GUI Applications in Python

- No built-in features for creating GUI
- No Graphical GUI Designers available – you need to create GUI in code
- There exists many external libraries for creating GUI Applications
- So, the the main problem is – which Library shall you use?
- There are pros and cons with all these libraries

GUI Python Libraries

- **PyQt5**
- **Tkinter**
- ...



Probably the 2 most used GUI libraries for Python

- **guizero** → This Tutorial will use guizero
- ... lots of others

Which are best?

There are pros and cons with all these libraries



guizero

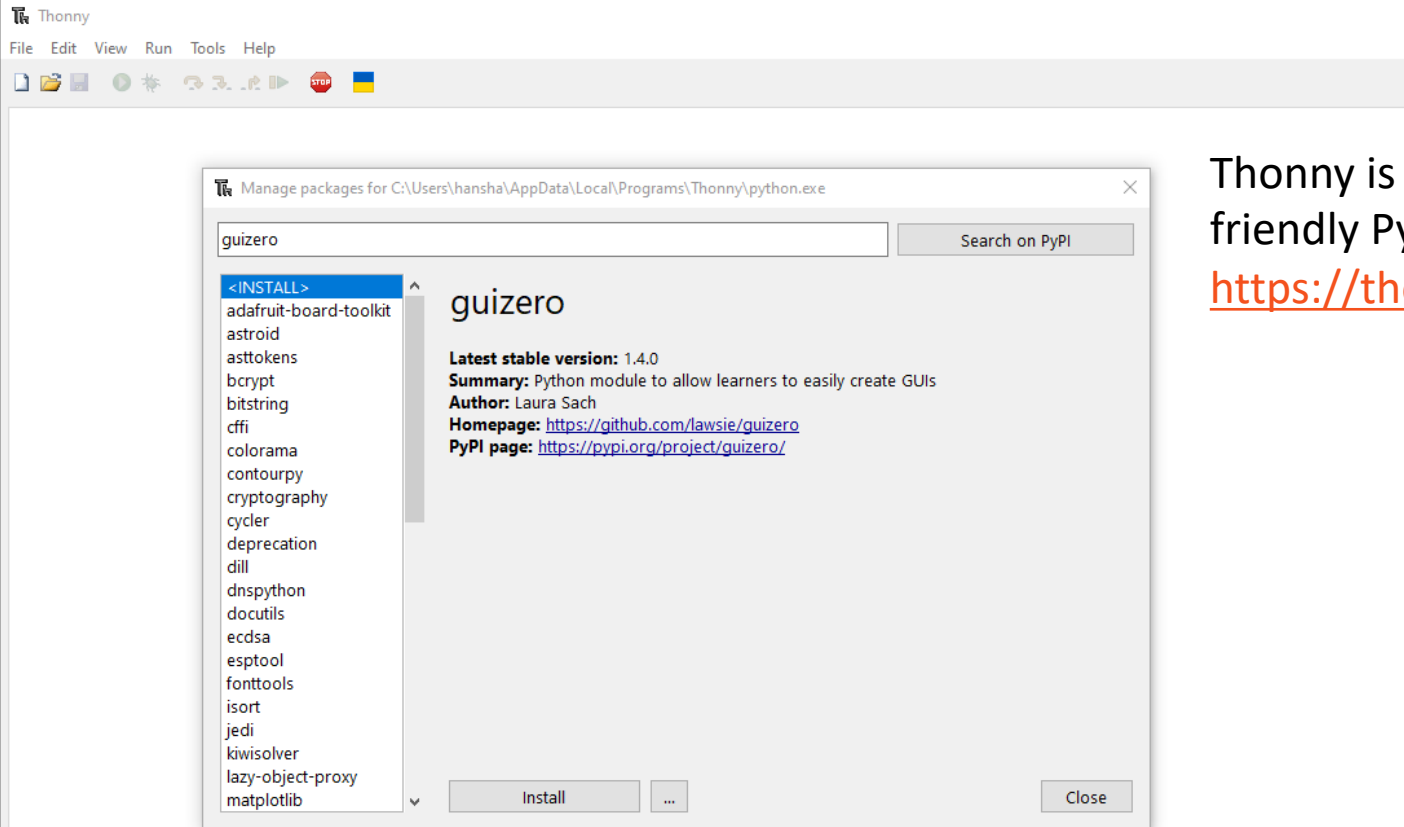
guizero

- guizero is a Python 3 library for creating simple GUIs
- It is easy to start with
- You simply install it by using pip:
`pip install guizero`
- Homepage: <https://lawsie.github.io/guizero/>

guizero References

- Getting started with GUIs:
<https://projects.raspberrypi.org/en/projects/getting-started-with-guis/1>
- guizero in PyPi: <https://pypi.org/project/guizero/>
- guizero Homepage: <https://lawsie.github.io/guizero/>
- GitHub: <https://github.com/lawsie/guizero>
- From Zero to GUI Hero:
https://dev.to/teach_wizbits/getting-started-with-graphical-user-interfaces-with-python-using-guizero-library-13o4

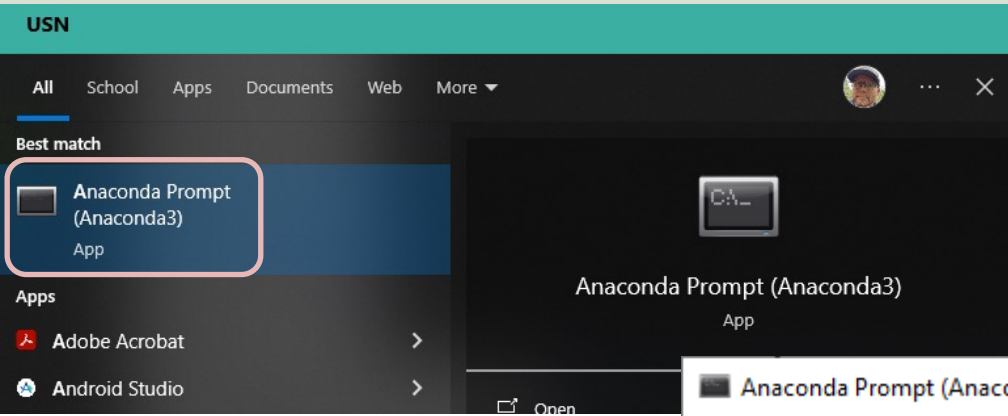
Install guizero using Thonny



Thonny is a simple and user-friendly Python Editor.

<https://thonny.org>

Install guizero using Anaconda



A screenshot of the Anaconda Prompt terminal window. The window title is "Anaconda Prompt (Anaconda3)". The terminal output shows the following commands and results:

```
(base) C:\Users\hansha>pip install guizero
Collecting guizero
  Downloading guizero-1.5.0-py3-none-any.whl (44 kB)
  | 44 kB 861 kB/s
Installing collected packages: guizero
Successfully installed guizero-1.5.0

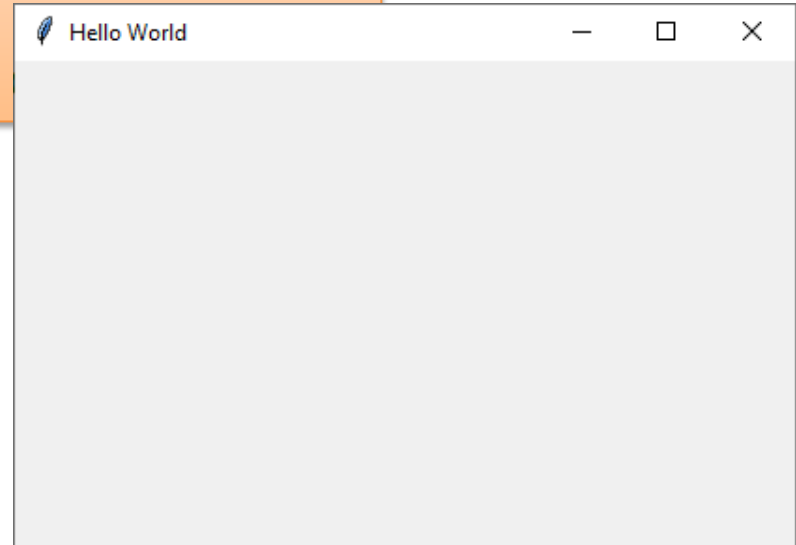
(base) C:\Users\hansha>
```

Hello World Example

```
from guizero import App
```

```
app = App(title="Hello World")
```

```
app.display()
```





Widgets Examples

guizero Widgets

- To create GUI objects with guizero, you use so-called widgets like App, Text, TextBox, PushButton, ListBox, CheckBox, Slider, Picture, etc.
- <https://lawsie.github.io/guizero/>

Examples

Some basic Python examples will be given:

- **Window**
- Using the **Text** widget
- Using the **TextBox** widget
- Using the **PushButton** widget
- **Plotting** Data (using matplotlib library and Picture widget)
- More examples can be found on the Homepage:
<https://lawsie.github.io/guizero/>



App

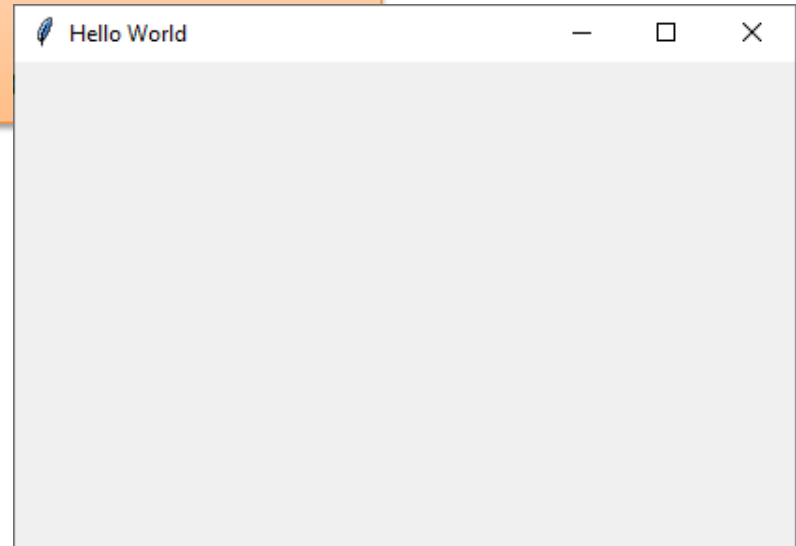
App

```
from guizero import App
```

```
app = App(title="Hello World")
```

```
app.display()
```

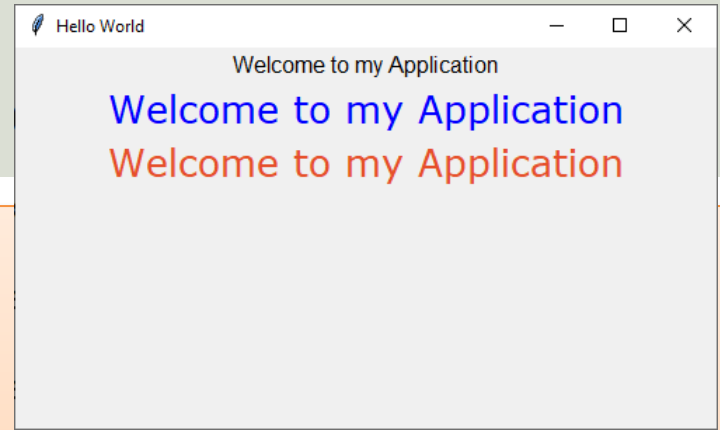
The App is the foundation for all guizero Applications. It creates an empty Window.





Text widget

Text Example



```
from guizero import App, Text
```

```
app = App(title="Hello World")
```

```
message = Text(app, text="Welcome to my Application")
```

```
message = Text(app, text="Welcome to my Application", size=20,  
font="Verdana", color="blue")
```

```
message = Text(app, text="Welcome to my Application", size=20,  
font="Verdana", color="#E64F2A")
```

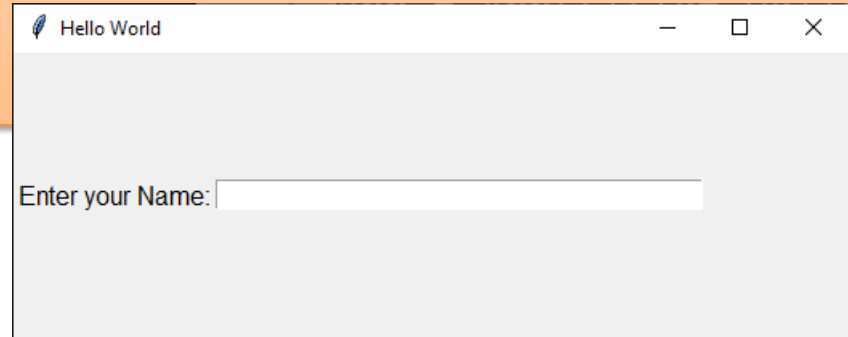
```
app.display()
```



TextBox widget

TextBox Example

```
from guizero import App, Text, TextBox  
  
app = App(title="Hello World")  
  
message = Text(app, text="Enter your Name:", align="left")  
  
textinput = TextBox(app, width=50, align="left")  
  
app.display()
```





PushButton widget

PushButton Example



```
from guizero import App, Text, PushButton
```

```
def button_click():  
    message.value = "You pressed the button!"
```

```
app = App(title="Hello World")
```

```
message = Text(app, text="Welcome to the Hello world app!")
```

```
button = PushButton(app, text="Press me", command=button_click)
```

```
app.display()
```

TextBox + PushButton Example

```
from guizero import App, Text, TextBox, PushButton
```

```
def button_click():  
    name = textinput.value  
    message.value = "Your name is: " + name
```

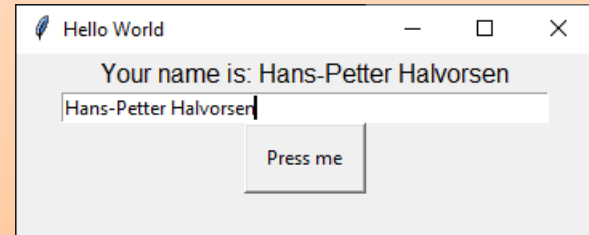
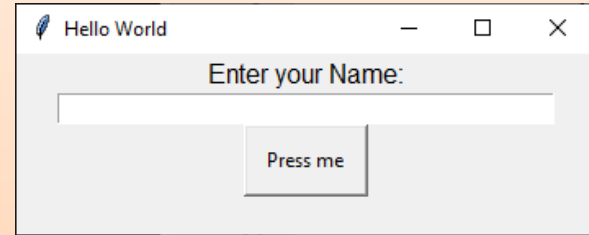
```
app = App(title="Hello World")
```

```
message = Text(app, text="Enter your Name:")
```

```
textinput = TextBox(app, width=50)
```

```
button = PushButton(app, text="Press me", command=button_click)
```

```
app.display()
```





MessageBox

Popup Window

MessageBox Example

Here we will demonstrate how to create a MessageBox or a Popup Window

```
from guizero import App, Text, TextBox, PushButton

def button_click():
    name = textinput.value
    message = "Your name is: " + name
    app.info("Name", message)

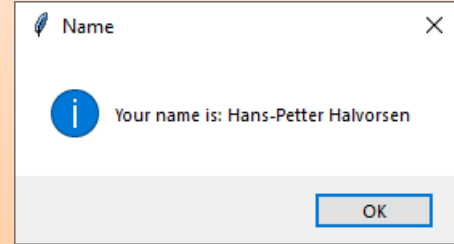
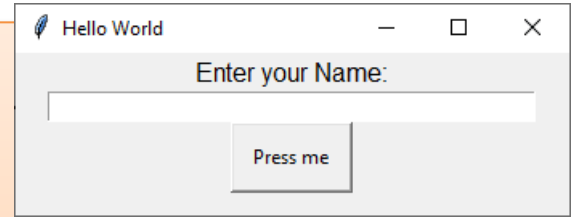
app = App(title="Hello World")

message = Text(app, text="Enter your Name:")

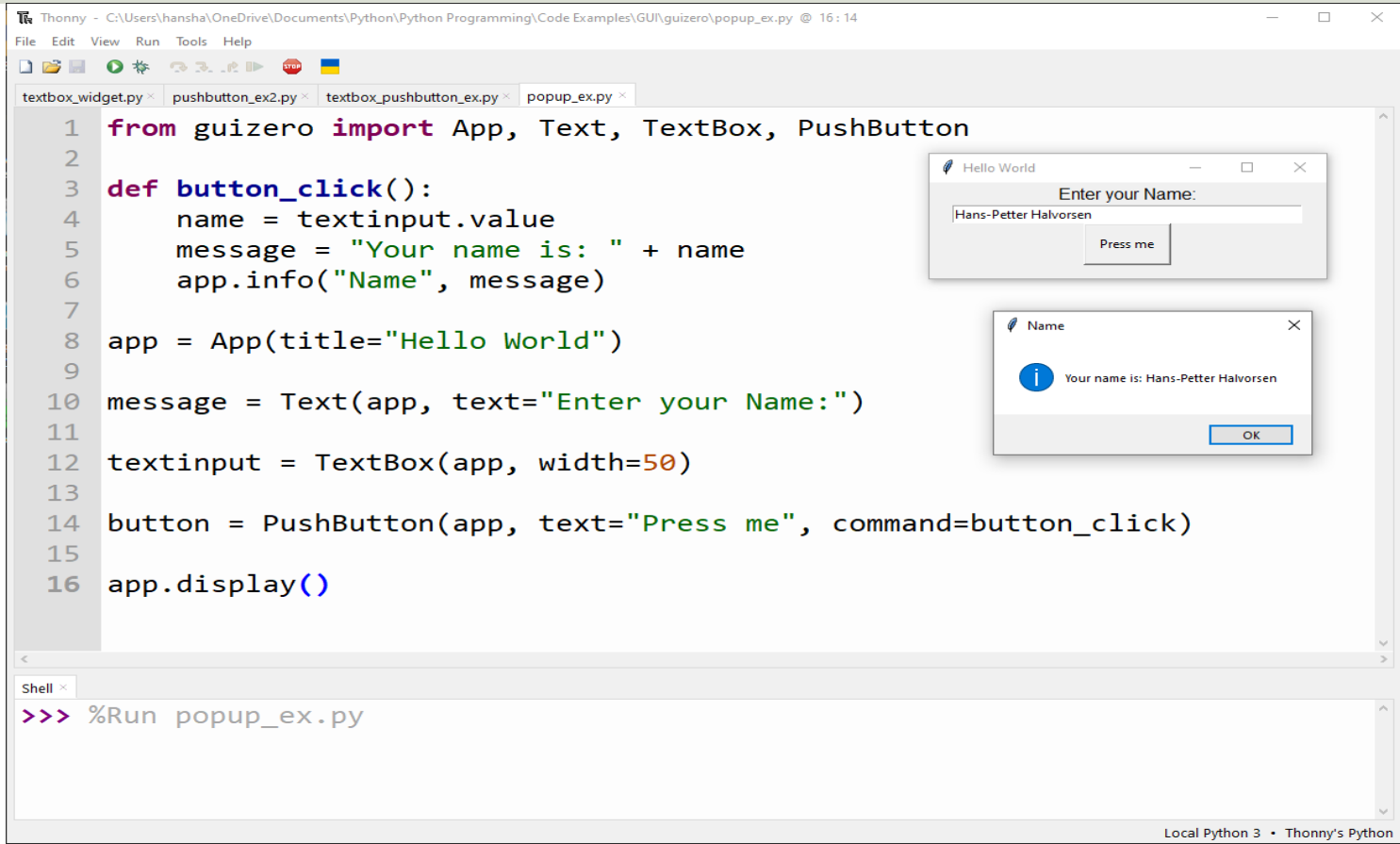
textinput = TextBox(app, width=50)

button = PushButton(app, text="Press me", command=button_click)

app.display()
```



MessageBox Example



The screenshot shows the Thonny IDE interface. The main editor displays a Python script named `popup_ex.py` with the following code:

```
1 from guizero import App, Text, TextBox, PushButton
2
3 def button_click():
4     name = textinput.value
5     message = "Your name is: " + name
6     app.info("Name", message)
7
8 app = App(title="Hello World")
9
10 message = Text(app, text="Enter your Name:")
11
12 textinput = TextBox(app, width=50)
13
14 button = PushButton(app, text="Press me", command=button_click)
15
16 app.display()
```

Below the code editor is a Shell window with the command `>>> %Run popup_ex.py` entered.

Two windows are shown on the right side of the IDE:

- The first window, titled "Hello World", contains a text input field with the text "Hans-Petter Halvorsen" and a "Press me" button.
- The second window, titled "Name", contains an information icon and the text "Your name is: Hans-Petter Halvorsen" with an "OK" button.

The status bar at the bottom right indicates "Local Python 3 • Thonny's Python".



Event Handling

Event Handling

We have the following Events:

- `when_clicked`
- `when_double_clicked`
- `when_left_button_pressed`
- `when_left_button_released`
- `when_right_button_pressed`
- `when_right_button_released`
- `when_key_pressed`
- `when_key_released`
- `when_mouse_enters`
- `when_mouse_leaves`
- `when_mouse_dragged`
- `when_resized`

<https://lawsie.github.io/guizero/events/>

Event Handling

```
from guizero import App, Text, TextBox
```

```
def counter1() :
```

```
    global i
```

```
    textinput.value = i
```

```
    i = i + 1
```

```
def counter2() :
```

```
    global i
```

```
    textinput.value = i
```

```
    i = i - 1
```

```
i = 1
```

```
app = App("Event Handler Example")
```

```
message = Text(app, text="Counter")
```

```
textinput = TextBox(app)
```

```
textinput.disable()
```

```
# Set Event Handling
```

```
app.when_clicked = counter1
```

```
app.when_right_button_pressed = counter2
```

```
app.display()
```



Timer

Timer Example

```
from guizero import App, Text, TextBox
```

```
def counter():  
    global i  
    textinput.value = i  
    i = i + 1
```

```
i = 1
```

```
app = App(title="Hello World")
```

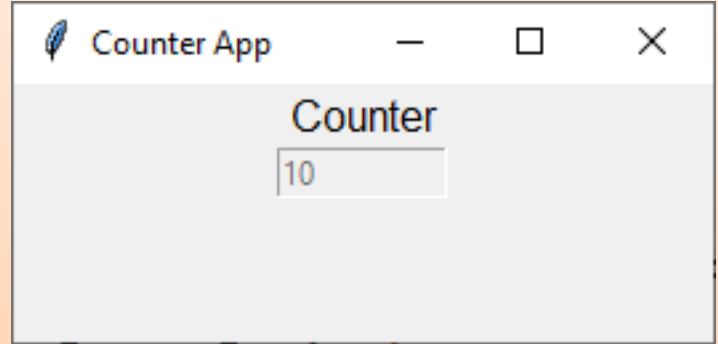
```
message = Text(app, text="Counter")  
textinput = TextBox(app)  
textinput.disable()
```

```
#Timer that updates the application periodically
```

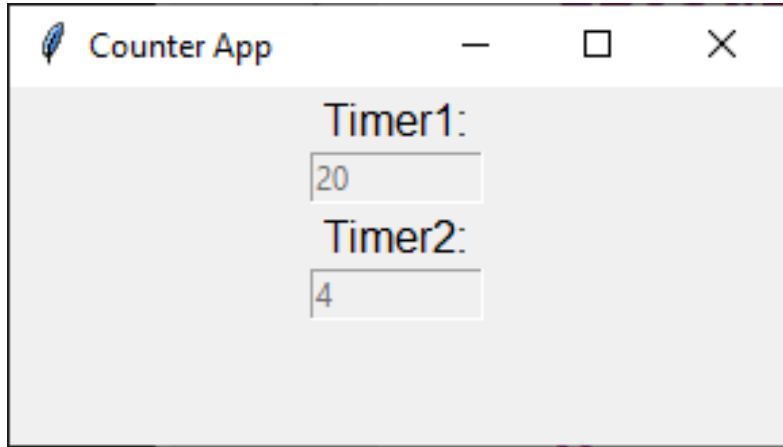
```
timer = 1000 #milliseconds
```

```
textinput.repeat(timer, counter)
```

```
app.display()
```



Multiple Timers



```
from guizero import App, Text, TextBox
```

```
def counter1():  
    global i  
    textinput1.value = i  
    i = i + 1
```

```
def counter2():  
    global j  
    textinput2.value = j  
    j = j + 1
```

```
i = 1  
j = 1
```

```
app = App(title="Counter App")
```

```
message = Text(app, text="Timer1:")  
textinput1 = TextBox(app)  
textinput1.disable()
```

```
message2 = Text(app, text="Timer2:")  
textinput2 = TextBox(app)  
textinput2.disable()
```

```
#Timer that updates the application periodically  
wait = 1000 #milliseconds  
textinput1.repeat(wait, counter1)
```

```
wait = 5000 #milliseconds  
textinput2.repeat(wait, counter2)
```

```
app.display()
```



Plotting Data

Plotting Data

- **Matplotlib** is a Python library that allows you to create a variety of charts and graphs.
- To use Matplotlib with guizero, you can embed the Matplotlib API directly into your guizero code
- You create a Matplotlib graph and saves it as a PNG file named, e.g., “graph.png”
- The graph can then be displayed in the GUI using the Picture widget in guizero

Basic Plotting Example

```
import matplotlib.pyplot as plt
from guizero import App, Text, Picture

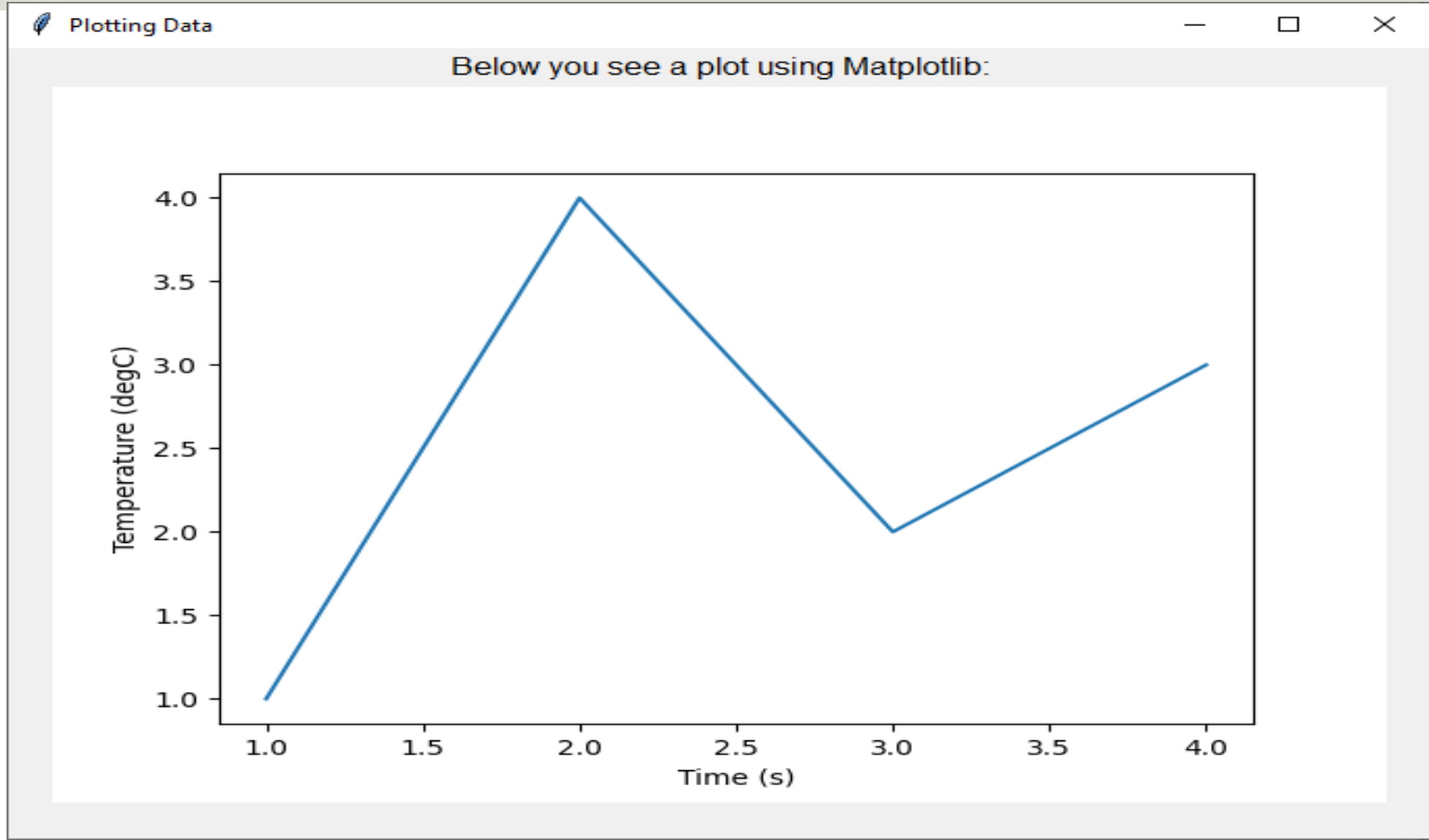
app = App(title="Plotting Data")
text = Text(app, text="Below you see a plot using Matplotlib:")

fig, ax = plt.subplots()
ax.plot([1, 2, 3, 4], [1, 4, 2, 3])
plt.xlabel('Time (s)')
plt.ylabel('Temperature (degC)')

fig.savefig("graph.png")
picture = Picture(app, image="graph.png")

app.display()
```

Plotting Example



Button + Plotting Example

```
from guizero import App, PushButton, Picture
import matplotlib.pyplot as plt
```

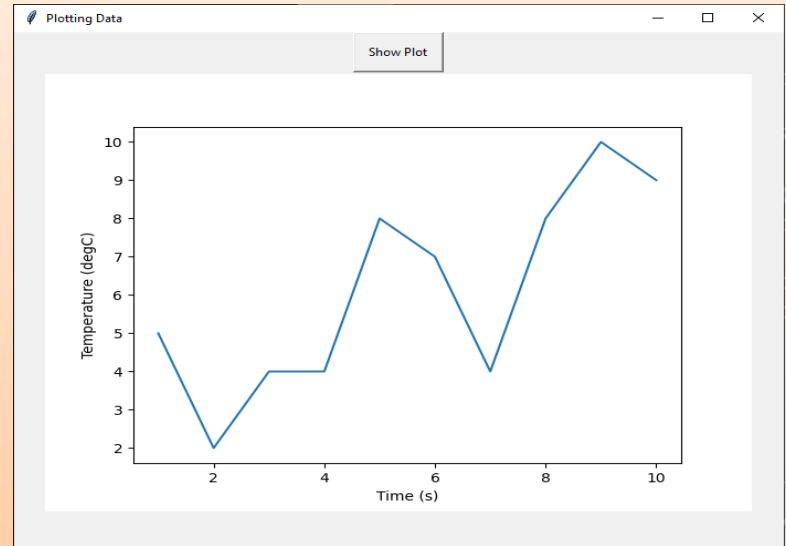
```
def plotdata():
    fig, ax = plt.subplots()
    x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    y = [5, 2, 4, 4, 8, 7, 4, 8, 10, 9]
    ax.plot(x, y)
    plt.xlabel('Time (s)')
    plt.ylabel('Temperature (degC)')

    fig.savefig("graph.png")
    picture = Picture(app, image="graph.png")
```

```
def button_click():
    plotdata()
```

```
# Main Program
app = App(title="Plotting Data")
button = PushButton(app, text="Show Plot", command=button_click)
```

```
app.display()
```



Refresh Plot Example

```
from guizero import App, PushButton, Picture
import matplotlib.pyplot as plt
import random
```

```
def plotdata():
```

```
    min = 20
    max = 30
    N = 10
    x = []
    y = []

    for i in range(N):
        x.append(i)
        newvalue = random.randint(min, max)
        y.append(newvalue)
```

```
fig, ax = plt.subplots()
ax.plot(x, y)
```

```
plt.xlabel('Time (s)')
plt.ylabel('Temperature (degC)')
```

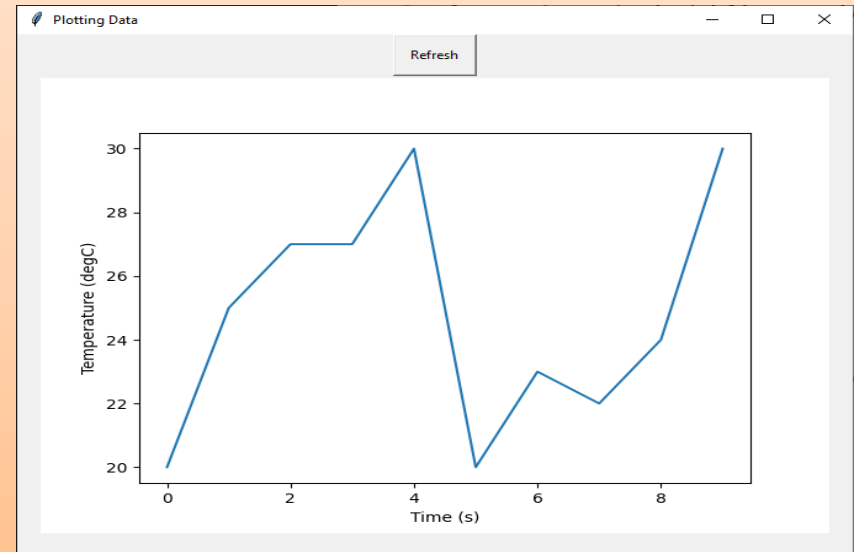
```
fig.savefig("graph.png")
picture.value="graph.png"
```

```
def button_click():
    plotdata()
```

```
# Main Program
app = App(title="Plotting Data")
button = PushButton(app, text="Refresh", command=button_click)
picture = Picture(app)
```

```
app.display()
```

Every time we hit the "Refresh" button, we get a new set of random data in the plot



Plotting Dynamic Data

Every time we hit the “Refresh” button we add a new value at the end of the plot

```
from guizero import App, PushButton, Picture
import matplotlib.pyplot as plt
import random
```

```
def plotdata():
    global i
    x.append(i)
    i = i + 1
    min = 20; max = 30
    newvalue = random.randint(min, max)
    y.append(newvalue)

    fig, ax = plt.subplots()
    ax.plot(x, y, '-o')

    plt.xlabel('Time (s)')
    plt.ylabel('Temperature (degC)')

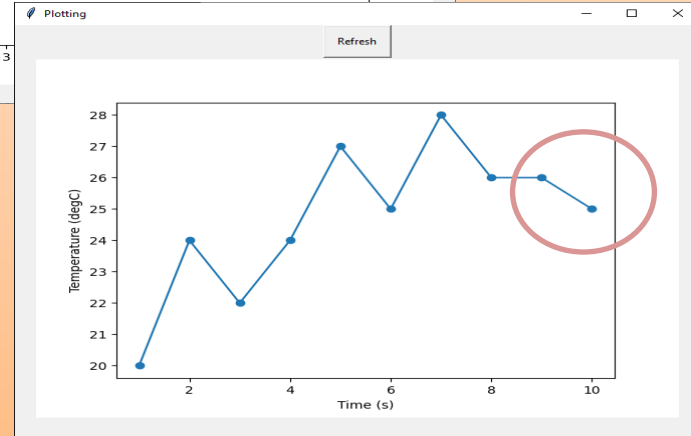
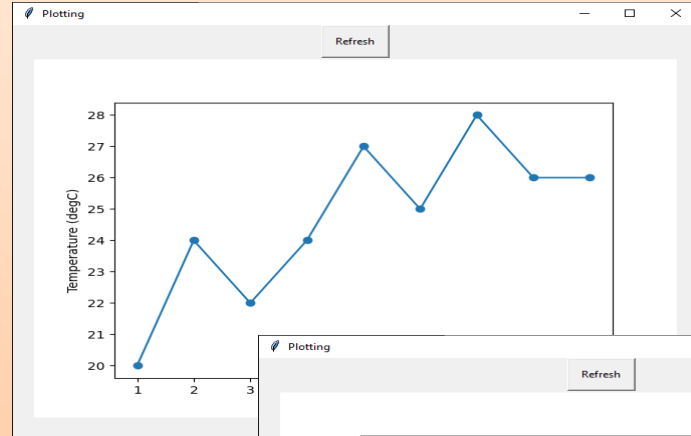
    fig.savefig("graph.png")
    picture.value="graph.png"
    plt.close()
```

```
def button_click():
    plotdata()
```

```
# Main Program
i = 1
x = []
y = []
```

```
app = App(title="Plotting")
button = PushButton(app, text="Refresh", command=button_click)
picture = Picture(app)
```

```
app.display()
```



Plotting Dynamic Data using a Timer

```
from guizero import App, Picture
import matplotlib.pyplot as plt
import random

def plotdata():
    global i
    x.append(i)
    i = i + 1
    min = 20; max = 30
    newvalue = random.randint(min, max)
    y.append(newvalue)

    fig, ax = plt.subplots()
    ax.plot(x, y, '-o')

    plt.xlabel('Time (s)')
    plt.ylabel('Temperature (degC)')

    fig.savefig("graph.png")
    picture.value="graph.png"
    plt.close()

# Main Program
i = 1
x = []; y = []

app = App(title="Plotting")
picture = Picture(app)

#Timer that updates the application periodically
timer = 1000 #milliseconds
picture.repeat(timer, plotdata)

app.display()
```

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

