

GUI's maken met Python: Meme-generator

Ontwerp een GUI-applicatie die memes maakt.



Laura Sach

Laura leidt het A Level-team bij de Raspberry Pi Foundation, waar ze middelen maakt voor leerlingen om te leren over computerwetenschappen.

@CodeBoom



Martin O'Hanlon

Martin werkt in het leerteam van de Raspberry Pi Foundation, waar hij online cursussen, projecten en leermiddelen creëert.

@martinohanlon

Laten we de lessen die je geleerd hebt uit de vorige afleveringen gebruiken om een GUI te ontwerpen die memes maakt. Je voert de tekst en de naam van de afbeelding in en je GUI zal ze combineren tot je eigen meme met behulp van de Drawing widget.

Begin met het maken van een simpele GUI met twee tekstboxen voor de bovenste en onderste tekst. Dit is waar je de tekst zult invoeren die over je afbeelding heen zal worden gezet om je meme te maken. Voeg deze regel toe om de benodigde widgets te importeren.

```
from guizero import App, TextBox, Drawing
```

Voeg dan deze code toe voor de app:

```
app = App("meme")

top_text = TextBox(app, "top text")
bottom_text = TextBox(app, "bottom text")

app.display()
```

De meme wordt gemaakt op een Drawing-widget die de afbeelding en de tekst zal bevatten.

Maak een meme

Voeg hem toe aan de GUI door het invoegen van onderstaande code net boven de regel `app.display()`. De hoogte en breedte van de Drawing-widget moeten worden ingesteld op 'fill' om de rest van de GUI te vullen.

```
meme = Drawing(app, width="fill",
height="fill")
```

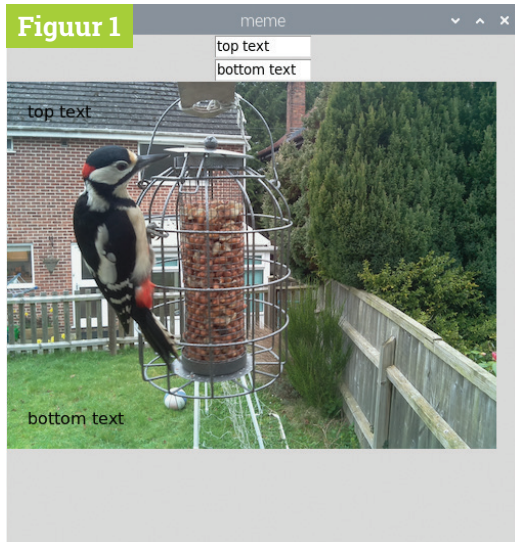
De meme zal worden gemaakt wanneer de tekst in de bovenste en onderste tekstvakken verandert. Hiervoor moeten we een functie maken die de meme samenstelt.

De functie moet de tekening leegmaken, een afbeelding maken (wij gebruiken een foto van een specht, maar je kunt elke foto gebruiken die je wilt) en de tekst boven en onder de afbeelding invoegen. Weet je nog dat je `name.value` gebruikte om de waarde van de Text-widget in te stellen met de naam van de spion in Deel 2 van deze serie? Je kunt ook de eigenschap `value` gebruiken om de waarde van een Text widget op te halen, dus in dit geval betekent `top_text.value` 'haal de waarde op die is getypt in de box `top_text`'.

```
def draw_meme():
    meme.clear()
    meme.image(0, 0, "woodpecker.png")
    meme.text(20, 20, top_text.value)
    meme.text(20, 320, bottom_text.value)
```

De eerste twee getallen in `meme.image(0, 0)` en `meme.text(20, 20)` zijn de x, y coördinaten van waar de afbeelding en de tekst getekend moeten worden. De afbeelding wordt getekend op positie 0, 0, dat is de linkerbovenhoek, zodat de afbeelding de hele tekening bedekt.

Roep tenslotte je `draw_meme` functie op vlak voordat je de app weergeeft. Voeg deze code in net



▲ **Figuur 1** Meme met niet opgemaakte tekst.

voor de regel `app.display()`:

```
draw_meme()
```

Je code zou er nu uit moeten zien als **meme1.py**. Als je je app uitvoert (**Figuur 1**) en probeert de tekst boven en onder te veranderen, zul je merken dat deze niet wordt bijgewerkt in de meme. Om dit werkend te krijgen, moet je je programma veranderen om de `draw_meme` functie aan te roepen wanneer de tekst verandert, door een commando toe te voegen aan de twee `TextBox` widgets in de app.

“ Werk je meme bij door de tekst boven en onder te veranderen ”

```
top_text = TextBox(app, "top text",
command=draw_meme)
bottom_text = TextBox(app, "bottom text",
command=draw_meme)
```

Je code zou er nu zo uit moeten zien in **meme2.py**. Voer het uit en pas je meme aan door de bovenste en onderste tekst te veranderen.

Je kunt het uiterlijk van je meme veranderen door de parameters `color`, `size` en `font` van de tekst te veranderen. Bijvoorbeeld:

meme1.py

➤ Taal: Python 3

**DOWNLOAD
DE VOLLEDIGE CODE:**



magpi.cc/guizero-code

```
001. # Imports -----
002.
003. from guizero import App, TextBox, Drawing
004.
005.
006. # Functions -----
007.
008. def draw_meme():
009.     meme.clear()
010.     meme.image(0, 0, "woodpecker.png")
011.     meme.text(20, 20, top_text.value)
012.     meme.text(20, 320, bottom_text.value)
013.
014.
015. # App -----
016.
017. app = App("meme")
018.
019. top_text = TextBox(app, "top text")
020. bottom_text = TextBox(app, "bottom text")
021.
022. meme = Drawing(app, width="fill", height="fill")
023.
024. draw_meme()
025.
026. app.display()
```

meme2.py

➤ Taal: Python 3

```
001. # Imports -----
002.
003. from guizero import App, TextBox, Drawing
004.
005.
006. # Functions -----
007.
008. def draw_meme():
009.     meme.clear()
010.     meme.image(0, 0, "woodpecker.png")
011.     meme.text(20, 20, top_text.value)
012.     meme.text(20, 320, bottom_text.value)
013.
014.
015. # App -----
016.
017. app = App("meme")
018.
019. top_text = TextBox(app, "top text", command=draw_meme)
020. bottom_text = TextBox(app, "bottom text", command=draw_meme)
021.
022. meme = Drawing(app, width="fill", height="fill")
023.
024. draw_meme()
025.
026. app.display()
```

Tip



Deze regels code begonnen erg lang te worden, dus hebben we ze opgesplitst over een aantal regels om de leesbaarheid te vergroten. Het heeft geen invloed op wat het programma doet, alleen hoe het eruit ziet.

```
meme.text(
    20, 20, top_text.value,
    color="orange",
    size=40,
    font="courier")
meme.text(
    20, 320, bottom_text.value,
    color="blue",
    size=28,
    font="times new roman",
)
```

Je code zou er nu uit moeten zien als **meme3.py**. Probeer verschillende stijlen tot je iets vindt dat je bevalt (**Figuur 2**).

meme3.py

► Taal: **Python 3**

```
001. # Imports -----
002.
003. from guizero import App, TextBox, Drawing
004.
005.
006. # Functions -----
007.
008. def draw_meme():
009.     meme.clear()
010.     meme.image(0, 0, "woodpecker.png")
011.     meme.text(
012.         20, 20, top_text.value,
013.         color="orange",
014.         size=40,
015.         font="courier")
016.     meme.text(
017.         20, 320, bottom_text.value,
018.         color="blue",
019.         size=28,
020.         font="times new roman",
021.     )
022.
023.
024. # App -----
025.
026. app = App("meme")
027.
028. top_text = TextBox(app, "top text", command=draw_meme)
029. bottom_text = TextBox(app, "bottom text", command=draw_meme)
030.
031. meme = Drawing(app, width="fill", height="fill")
032.
033. draw_meme()
034.
035. app.display()
```

Pas je meme-generator aan

Voor een echt interactieve meme-generator moet de gebruiker het lettertype, de grootte en de kleur zelf kunnen instellen. Je kunt extra widgets op de GUI zetten om hen in staat te stellen dit te doen.

Het aantal beschikbare opties voor de kleur en het lettertype zijn beperkt, dus je zou hiervoor een drop-down lijst, ook bekend als een Combo box, kunnen gebruiken. De grootte kun je instellen met een Slider widget.

Pas eerst je import statement aan om de Combo en Slider widgets op te nemen.

```
from guizero import App, TextBox, Drawing,
Combo, Slider
```

Nadat je je TextBox widgets hebt gemaakt voor de bovenste en onderste tekst, maak je een nieuwe Combo widget zodat de gebruiker een kleur kan kiezen.

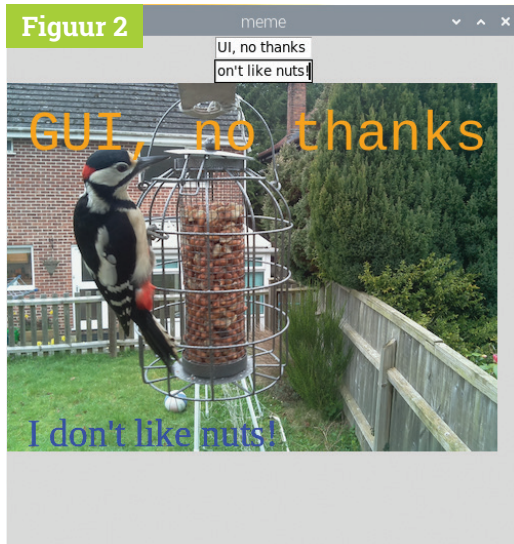
```
bottom_text = TextBox(app, "bottom text",
command=draw_meme)
color = Combo(app,
    options=["black", "white", "red",
"green", "blue", "orange"],
    command=draw_meme)
```

“ De opties worden getoond in de volgorde waarin je ze in de lijst zet ”

De parameter **options** bepaalt welke kleuren de gebruiker kan kiezen uit de Combo. Elke kleur is een element in een lijst. Je kunt alle andere kleuren die je wilt aan de lijst toevoegen.

De opties worden getoond in de volgorde waarin je ze in de lijst zet. De eerste optie is de standaard, die als eerste wordt weergegeven. Als je een andere optie als standaard wilt hebben, kun je dat doen met de parameter **selected**, bijvoorbeeld **"blue"**.

```
color = Combo(app,
    options=["black", "white", "red",
"green", "blue", "orange"],
    command=draw_meme,
    selected="blue")
```



▲ **Figuur 2** Lettertypen en kleuren wijzigen

Nu kan je gebruiker een kleur kiezen. Vervolgens moet je de `draw_meme` functie zo aanpassen dat de waarde van Combo wordt gebruikt bij het maken van de tekst in je meme. Bijvoorbeeld:

```
meme.text(
    20, 20, top_text.value,
    color=color.value,
    size=40,
    font="courier")
```

Doe hetzelfde voor het blok code voor de onderste tekst. Je programma zou nu moeten lijken op **meme4.py**.

Voeg volgens de bovenstaande stappen een tweede Combo toe aan je applicatie zodat de gebruiker een lettertype kan kiezen uit deze lijst met opties: ["times new roman", "verdana", "courier", "impact"]. Vergeet niet om de `draw_meme` functie aan te passen zodat de waarde van `font` gebruikt wordt bij het toevoegen van de tekst.

Maak een nieuwe Slider widget om de grootte van de tekst in te stellen die je gebruiker wil.

```
size = Slider(app, start=20, end=40,
    command=draw_meme)
```

Het bereik van de schuif wordt ingesteld met de parameters `start` en `end`. In dit voorbeeld zal de kleinste beschikbare tekst dus 20 zijn en de grootste 40.

meme4.py

► Taal: **Python 3**

```
001. # Imports -----
002.
003. from guizero import App, TextBox, Drawing, Combo, Slider
004.
005.
006. # Functions -----
007.
008. def draw_meme():
009.     meme.clear()
010.     meme.image(0, 0, "woodpecker.png")
011.     meme.text(
012.         20, 20, top_text.value,
013.         color=color.value,
014.         size=40,
015.         font="courier")
016.     meme.text(
017.         20, 320, bottom_text.value,
018.         color=color.value,
019.         size=28,
020.         font="times new roman",
021.         )
022.
023.
024. # App -----
025.
026. app = App("meme")
027.
028. top_text = TextBox(app, "top text", command=draw_meme)
029. bottom_text = TextBox(app, "bottom text", command=draw_meme)
030.
031. color = Combo(app,
032.     options=["black", "white", "red", "green",
033.             "blue", "orange"],
034.     command=draw_meme, selected="blue")
035.
036. meme = Drawing(app, width="fill", height="fill")
037.
038. draw_meme()
039. app.display()
```

Drawing widget

De Drawing widget is zeer veelzijdig en kan worden gebruikt om veel verschillende vormen, patronen en afbeeldingen weer te geven. Voor meer informatie over de Drawing widget, zie Appendix C van het boek (magpi.cc/pythongui), of kijk eens naar de online documentatie: lawsie.github.io/guizero/drawing.

Python 3 Programmeren en GUI's

Dit is de tweede editie van een boek gericht op ingenieurs, wetenschappers en hobbyisten die PC's willen interfacen met hardware projecten door gebruik te maken van grafische user interfaces. Desktop en web gebaseerde toepassingen worden behandeld. De gebruikte programmeertaal is Python 3, een van de populairste talen op de markt: snelheid van programmeren is een belangrijk kenmerk. Het boek is herzien en bijgewerkt met de nadruk op de gebruiker om praktische ontwerpen met gemak te produceren - een teksteditor is alles wat nodig is om Python-programma's te produceren. www.elektor.nl/python-3-programming-and-guis

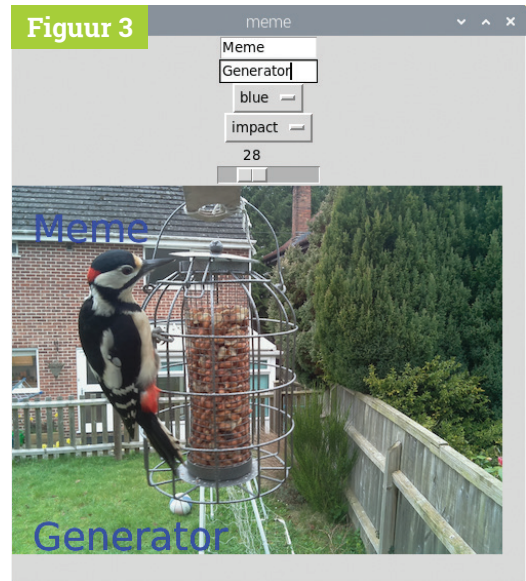


04-meme-generator.py

► Taal: Python 3

```
001. # Imports -----
002.
003. from guizero import App, TextBox, Drawing, Combo, Slider
004.
005.
006. # Functions -----
007.
008. def draw_meme():
009.     meme.clear()
010.     meme.image(0, 0, "woodpecker.png")
011.     meme.text(
012.         20, 20, top_text.value,
013.         color=color.value,
014.         size=size.value,
015.         font=font.value)
016.     meme.text(
017.         20, 320, bottom_text.value,
018.         color=color.value,
019.         size=size.value,
020.         font=font.value,
021.     )
022.
023.
024. # App -----
025.
026. app = App("meme")
027.
028. top_text = TextBox(app, "top text", command=draw_meme)
029. bottom_text = TextBox(app, "bottom text", command=draw_meme)
030.
031. color = Combo(app,
032.     options=["black", "white", "red", "green",
033.             "blue", "orange"],
034.     command=draw_meme, selected="blue")
035.
036. font = Combo(app,
037.     options=["times new roman", "verdana", "courier",
038.             "impact"],
039.     command=draw_meme)
040.
041. size = Slider(app, start=20, end=50, command=draw_meme)
042.
043. meme = Drawing(app, width="fill", height="fill")
044. draw_meme()
045. app.display()
```

Figuur 3



▲ Figuur 3 De voltooide meme-generator.

Pas de `draw_meme` functie aan om de waarde van je schuif te gebruiken bij het maken van de tekst van de meme.

```
meme.text(
    20, 20, top_text.value,
    color=color.value,
    size=size.value,
    font=font.value)
```

Je code zou nu moeten lijken op die in **04-meme-generator.py**. Probeer het uit te voeren en je zou iets als **Figuur 3** moeten zien.

Kun je de GUI zo veranderen dat de naam van het afbeeldingsbestand in een TextBox kan worden ingevoerd of misschien uit een lijst in een Combo kan worden gekozen? Hiermee zou je applicatie ook in staat zijn om memes met verschillende afbeeldingen te genereren. [\[1\]](#)