

GUI's maken met Python: Spionnennaamkiezer

Maak een interactieve GUI-toepassing.



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 maakt.

@martinohanlon

Tot nu toe heb je in deze serie geleerd hoe je je GUI kunt aanpassen met een verscheidenheid aan verschillende opties.

Nu is het tijd om echt interactief te worden en een GUI-applicatie te maken die echt reageert op input van de gebruiker. Wie wil nou niet op een grote rode knop drukken om een supergeheime spionnennaam te genereren?

Aangezien je al weet hoe je een app moet maken, maak dan eens een basisvenster en voeg wat tekst toe als je wilt. Hier is wat code om je op weg te helpen, en deze code bevat ook wat commentaar (de regels die beginnen met een #) om je te helpen je programma te structureren:

```
# Imports -----
from guizero import App, Text

# Functions -----

# App -----
app = App("TOP SECRET")

# Widgets -----
title = Text(app, "Push the red button to find out your spy name")

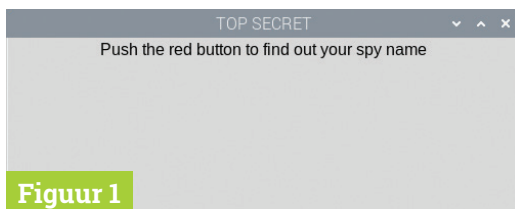
# Display -----
app.display()
```

Voer deze code uit en je zou een venster moeten zien met de tekst (**Figuur 1**).

Voeg een knop toe

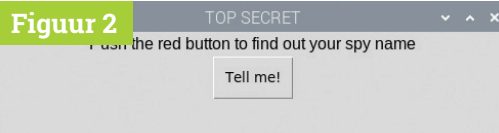
Laten we nu een knop toevoegen aan de GUI. Voeg `PushButton` toe aan je lijst van imports zodat je knoppen kunt gebruiken. (Let op dat je een hoofdletter B gebruikt!)

Onder de `Text`-widget, maar voordat de app wordt weergegeven, voeg je een regel code toe om een



Figuur 1

► **Figuur 1** De tekst in een venster weergeven.



Figuur 2

► **Figuur 2** Je hebt nu een knop. knop te maken.

```
button = PushButton(app, choose_name,
text="Tell me!")
```

Je code zou er nu uit moeten zien als **spy1.py**. Voer hem uit en er zal geen knop verschijnen, maar je zult een fout zien in het Shell-venster:

```
NameError: name 'choose_name' is not defined
```

Dit komt omdat `choose_name` de naam is van een commando dat wordt uitgevoerd als de knop wordt ingedrukt. Aan de meeste GUI-componenten kan een commando gekoppeld worden. Voor een knop betekent het toevoegen van een commando "als de knop wordt ingedrukt, voer dan dit commando uit". Een GUI-programma werkt anders dan andere Python programma's die je misschien hebt geschreven omdat de volgorde waarin de commando's in het programma worden uitgevoerd volledig afhangt van de volgorde waarin de gebruiker op de knoppen drukt, de schuifregelaars beweegt, de vakjes aanvinkt, of interactie heeft met welke andere widgets je ook maar gebruikt. Het eigenlijke commando is bijna altijd de naam van een functie die moet worden uitgevoerd.

Maak een functie

Laten we de functie `choose_name` schrijven zodat je knop iets kan doen als hij wordt ingedrukt.

Zoek in je programma de sectie `Functions` op. Dit is waar alle functies moeten komen die aan GUI-widgets worden gekoppeld, om ze gescheiden te houden van de code voor het tonen van de widget. Voeg deze code toe in de sectie `Functions`:

```
def choose_name():
    print("Button was pressed")
```

Je code zou er nu uit moeten zien als **spy2.py**. De knop zal nu verschijnen (**Figuur 2**). Als je op de knop drukt, lijkt het alsof er niets gebeurt, maar als je in

je Shell- of uitvoervenster kijkt, zul je zien dat daar wat tekst is verschenen (Figuur 3).

Het is slim om je functie eerst wat testtekst te laten weergeven, zodat je zeker weet dat de knop zijn opdrachtfunctie correct uitvoert wanneer hij wordt ingedrukt. Vervolgens kun je de `print` statement vervangen door de eigenlijke code voor de taak die je de knop wilt laten uitvoeren.

Typ in je `choose_name` functie een hekje (`#`) voor de regel code die 'Button was pressed' print. Programmeurs noemen dit 'uitcommentariëren', hiermee heb je de computer verteld om deze regel code te behandelen alsof het een commentaar is, of anders gezegd, je hebt de computer geïnstrueerd om het te

spy1.py

> Taal: Python 3

DOWNLOAD
DE VOLLEDIGE CODE:

 magpi.cc/guizerocode

```
001. # Imports -----
002. from guizero import App, Text, PushButton
003.
004. # Functions -----
005.
006. # App -----
007. app = App("TOP SECRET")
008.
009. # Widgets -----
010. title = Text(app, "Push the red button to find out your spy name")
011. button = PushButton(app, choose_name, text="Tell me!")
012.
013. # Display -----
014. app.display()
```

spy2.py

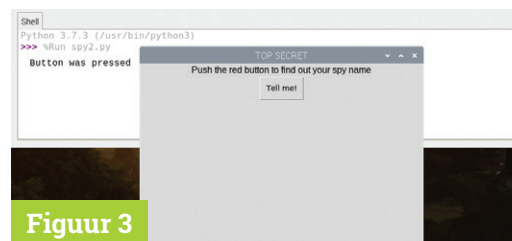
> Taal: Python 3

```
001. # Imports -----
002. from guizero import App, Text, PushButton
003.
004. # Functions -----
005.
006. # App -----
007. app = App("TOP SECRET")
008.
009. # Widgets -----
010. title = Text(app, "Push the red button to find out your spy name")
011. button = PushButton(app, choose_name, text="Tell me!")
012.
013. # Display -----
014. app.display()
```

Grote rode knop

Op dit moment is je knop niet groot of rood! Je hebt in deel één van deze tutorialserie eigenschappen gebruikt om het uiterlijk van je tekst op de 'Wanted' poster te veranderen. Kun je nu de eigenschappen van de PushButton widget gebruiken om de achtergrondkleur en de tekstgrootte te veranderen?

Let op: op macOS is het misschien niet mogelijk is om de kleur van een knop te wijzigen, omdat sommige versies van het besturingssysteem dit niet toestaan, maar je zou nog wel de tekstgrootte moeten kunnen wijzigen.



▲ **Figuur 3** Tekst wordt uitgevoerd naar het Shell-venster.

negeren. Het voordeel van het uitcommentariëren van een regel code in plaats van die gewoon te verwijderen, is dat als je die code ooit weer wilt gebruiken, je hem gemakkelijk weer deel kunt laten uitmaken van je programma door het `#`-symbool te verwijderen.

Namen toevoegen

Typ op een nieuwe regel een lijst met voornamen. Je kunt zelf bepalen welke namen er in je lijst komen en het mogen er zoveel zijn als je wilt, maar zorg ervoor dat elke naam tussen aanhalingstekens staat, en dat de namen van elkaar gescheiden zijn door een komma. Een verzameling letters, cijfers en/of leestekens tussen aanhalingstekens wordt een *string* genoemd, dus we zeggen dat elke naam een string moet zijn.

```
first_names = ["Barbara", "Woody",
               "Tiberius", "Smokey", "Jennifer", "Ruby"]
```

Voeg nu ook een lijst van achternamen toe:

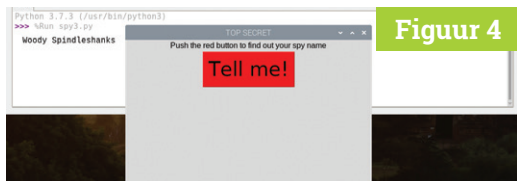
```
last_names = ["Spindleshanks", "Mysterioso",
              "Dungeon", "Catseye", "Darkmeyer",
              "Flamingobreath"]
```

Nu moet je een manier toevoegen om een willekeurige naam te kiezen uit elke lijst om je spionennaam te vormen. Je eerste taak is het toevoegen van een nieuwe import-regel in je sectie imports:

```
from random import choice
```

Dit vertelt het programma dat je een functie wilt gebruiken, `choice` genaamd, die een willekeurig item uit een lijst kiest. Iemand anders heeft de code geschreven die dit voor je doet, en het is meegeleverd met Python zodat je het kunt gebruiken.

In je code voor de `choose_name` functie, net onder je lijsten met namen, voeg je een regel code toe om de voornaam van je spion te kiezen, en daar vervolgens een achternaam aan te hangen, met een spatie ertussen. Dit wordt 'concatenation' (aaneenschakeling) genoemd, dat betekent 'voeg twee strings samen' en het symbool in Python voor concatenation is een plus (+).



▲ **Figuur 4** Het uitvoeren van een spionennaam.

```
spy_name = choice(first_names) + " " +
choice(last_names)
print(spy_name)
```

Je code zou nu moeten lijken op **spy3.py**. Sla hem op en voer hem uit. Wanneer je op de knop drukt, zou er een willekeurig gegenereerde spionennaam moeten verschijnen in je console of Shell, op dezelfde plaats waar eerst de melding 'Button was pressed' verscheen (**Figuur 4**).

Zet de naam in de GUI

Dat is goed, maar zou het niet mooier zijn als de spionennaam in de GUI zou verschijnen? Laten we een andere Text-widget maken en die gebruiken om de naam van de spion weer te geven.

Voeg in de sectie widgets een nieuwe Tekst-widget toe die de spionennaam zal weergeven:

```
name = Text(app, text="")
```

Wanneer je de widget maakt, wil je niet dat die iets van tekst laat zien omdat de persoon nog niet op de knop heeft gedrukt, dus kun je de tekst op "" zetten, wat een 'lege string' wordt genoemd en niets weergeeft. In je `choose_name` functie, zet je een hekje vóór de regel code waarin je de spionennaam print.

Voeg nu een nieuwe regel code toe aan het einde van de functie om de waarde van de Text-widget name in te stellen op de `spy_name` die je zojuist hebt gemaakt. Dit zal ervoor zorgen dat de Text-widget zichzelf update en de naam weergeeft.

```
name.value = spy_name
```

Je uiteindelijke code zou moeten zijn zoals in **03-spy-name-chooser.py**. Voer die uit en druk op de knop om je spionennaam met trots te zien verschijnen op de GUI (**Figuur 5**).

Je kunt nogmaals op de knop drukken als je de naam die je krijgt niet leuk vindt, en het programma zal een willekeurige andere naam voor je genereren. 🐍



Figuur 5

▲ **Figuur 5** De voltooide spionennaamkiezer.

spy3.py

► Taal: Python 3

```
001. # Imports -----
002. from guizero import App, Text, PushButton
003. from random import choice
004.
005. # Functions -----
006. def choose_name():
007.     #print("Button was pressed")
008.     first_names = ["Barbara", "Woody", "Tiberius", "Smokey",
009.     "Jennifer", "Ruby"]
010.     last_names = ["Spindleshanks", "Mysterioso", "Dungeon",
011.     "Catseye", "Darkmeyer", "Flamingobreath"]
012.     spy_name = choice(first_names) + " " + choice(last_names)
013.     print(spy_name)
014.
015. # App -----
016. app = App("TOP SECRET")
017.
018. # Widgets -----
019. title = Text(app, "Push the red button to find out your spy name")
020. button = PushButton(app, choose_name, text="Tell me!")
021. button.bg = "red"
022. button.text_size = 30
023.
024. # Display -----
025. app.display()
```

03-spy-name-chooser.py

► Taal: Python 3

```
001. # Imports -----
002. from guizero import App, Text, PushButton
003. from random import choice
004.
005. # Functions -----
006. def choose_name():
007.     #print("Button was pressed")
008.     first_names = ["Barbara", "Woody", "Tiberius", "Smokey",
009.     "Jennifer", "Ruby"]
010.     last_names = ["Spindleshanks", "Mysterioso", "Dungeon",
011.     "Catseye", "Darkmeyer", "Flamingobreath"]
012.     spy_name = choice(first_names) + " " + choice(last_names)
013.     #print(spy_name)
014.     name.value = spy_name
015.
016. # App -----
017. app = App("TOP SECRET")
018.
019. # Widgets -----
020. title = Text(app, "Push the red button to find out your spy name")
021. button = PushButton(app, choose_name, text="Tell me!")
022. button.bg = "red"
023. button.text_size = 30
024. name = Text(app, text="")
025.
026. # Display -----
027. app.display()
```