

How do I pass parameters to a function from a guizero object?

Asked 4 years, 11 months ago Modified 4 years, 11 months ago Viewed 2k times



I am unable to pass any parameter to a function called by a guizero widget either with the attribute "command" at initialization or by invoking an event.

2

This works as expected (no parameters passed):



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

```
def go():  
    print (10)
```

```
app = App(title="Test")  
button = PushButton(app, text = "Click Here", command = go)  
  
app.display()
```

but the following prints the number 10 **once, before** the button is clicked and then when I click, nothing happens

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

```
def go(n):  
    print (n)
```

```
app = App(title="Test")  
button = PushButton(app, text = "Click Here", command = go(10))  
  
app.display()
```

The same result I get with this:

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

```
def go(n):  
    print (n)
```

```
app = App(title="Test")  
button = PushButton(app, text = "Click Here")  
button.when_clicked = go(10)  
  
app.display()
```

What am I missing?

Thanks in advance!

Share Improve this question Follow

edited May 12, 2019 at 0:09

asked May 12, 2019 at 0:01



[lurker](#)

57.5k

9

71

107



[Thot](#)

45

8

[Report this ad](#)

3 Answers

Sorted by: Highest score (default)



4



```
from guizero import App, PushButton
def go(n):
    print (n)
app = App(title="Test")
button = PushButton(app, text = "Click Here", command = lambda: go(10))
app.display()
```



Whenever you write `go(10)` anywhere, you are invoking the function `go`. You might think you are passing `go` with arguments, but you aren't because the parentheses next to `go()` invoke the function right then and there. If you want to pass the function `go` to another function, and `go` should also be passed with some arguments, then you need to wrap the function `go` and pass the wrapped function as the argument "command". Using a lambda function to wrap `go(10)` is one such way of doing this.

The reason this works is that the lambda function is NOT invoked right then and there. You are saying that that `command()` should invoke the **declared** anonymous lambda function eventually, and when that lambda function is called it will itself call `go(10)`. You are **declaring** an anonymous lambda function, NOT invoking it. The lambda function will be invoked later on as `command()`.

Share Improve this answer Follow

edited May 12, 2019 at 0:41

answered May 12, 2019 at 0:19



In all GUI frameworks `command=` (or similar) expects `callback` - it means function's name without `()` and without parameters.

1



If you use `go(10)` then it does



```
result = go(10)
PushButton( ..., command=result)
```



so it executes `go(10)` before you even click button - and it assign result to `command=`. Because `go()` returns `None` so you get `PushButton(..., command=None)` and when you click button then you get nothing.

You have to create new function which you can run without `()`

```
def go_10():
    go(10)

PushButton( ..., command=go_10)
```

Or you can use `lambda` to create function directly in `command=`

```
PushButton( ..., command=lambda:go(10))
```

Share Improve this answer Follow

edited May 12, 2019 at 0:25

answered May 12, 2019 at 0:19



furas

138k

12

111

157



I found the solution!

1



Using a **lambda** function is a clever way to work around the callback limitation but I found a more proper way to pass arguments to a function by using the widget properties (pun not intended). As it turns out I missed a very brief reference to the property *args* that can be set at instantiation and is used to pass a list to the function called by *command*



So here is my solution:



```
from guizero import App, PushButton

def go(n):
    print (n)
```

```
app = App(title="Test")
button = PushButton(app, text = "Click Here", args = [10], command = go)

app.display()
```

As expected, it prints '10' every time you click the button and not before.

Thanks for all the other answers!

[Share](#) [Improve this answer](#) [Follow](#)

answered May 21, 2019 at 15:00



[Thot](#)

45 8
