

[< VOLTAR](#)

Programação em GUI com Tkinter

Apresentar o recurso Tkinter do Python para programação com interface gráfica.

NESTE TÓPICO

> ALGUMAS
CARACTERÍSTICAS DA
PROGRAMAÇÃO GUI EM
PYTHON:

> ALGUNS PARÂMETROS
▼ Marcar
tópico



Olá alunos,

No início, na década de 60 e 70, estávamos na era da interface por linha de comando: CLI (Command Line Interface), onde o principal dispositivo deste tipo de interface é o teclado: toda a “navegação” pela tela era realizada pelas teclas ou por pressionar uma combinação de duas ou mais teclas. A partir da década de 80, aparece a GUI (Graphical User Interface), interface gráfica de usuário, onde o principal dispositivo, além do teclado, é o mouse para navegarmos pela tela do monitor. É o tipo de interface mais utilizada atualmente, principalmente para aplicações em desktop. E temos também hoje a NUI (Natural User Interface), interface natural do usuário, para interagir com o sistema não será necessário teclado ou mouse, basta realizar movimentos com as mãos ou com outras partes do corpo. Um exemplo deste tipo de interface é o Kinect® da Microsoft® que executam aplicativos que reconhecem os movimentos do corpo humano.

Com a linguagem Python também podemos trabalhar no ambiente de desenvolvimento do MS Visual Studio® e controlar o dispositivo Kinect®. No entanto, Python vem com bibliotecas para trabalharmos em GUI. O padrão incluso é o Tcl/Tk que possui o módulo **Tkinter (Tk)** que é uma biblioteca GUI escrita em linguagem C para o Tcl (o interpretador do Tk), é leve, portátil, simples e robusta. Vem nativa na instalação do Python e podemos utilizar na interface IDLE. Para utilizá-la basta invocar a biblioteca:

```
from tkinter import *
```

Existem outros tipos de bibliotecas gráficas para Python, tais como: PyQt, Gtk, wxWidgets, PySide, etc.

Para programarmos em Tk, por exemplo, abra o IDLE e crie o código em script.

Na hora de salvar, poderemos fazê-lo de duas formas: colocando o nome do script com a extensão **.py** (*nome.py*) ou com a extensão **.pyw** (*nome.pyw*). A diferença é que ao executarmos com a extensão **.py**, será aberto a tela do interpretador junto com a aplicação, enquanto na extensão **.pyw**, vai abrir somente a tela da aplicação.

ALGUMAS CARACTERÍSTICAS DA PROGRAMAÇÃO GUI EM PYTHON:

Para que possamos programar em GUI, é necessário utilizarmos as técnicas da Orientação a Objetos.

class Classe: encapsula objetos, métodos e atributos.

def Cria um método ou função. Nomes iniciados por **'_'** significa que o método é privado.

self Objeto principal (base) e padrão do Python.

container Recipiente, depósito, caixa: armazena vários objetos (widgets).

widget Objetos de uma tela, tais como: button, label, frame, icon, etc.

frame Área retangular que pode conter outros **widgets**.

ALGUNS PARÂMETROS DA PROGRAMAÇÃO GUI EM PYTHON:

font Define o tipo de fonte do texto: (*fontname, fontsize, fonttype*), onde *fonttype* pode ser *bold, italic* ou *normal*.

background (bg) Cor de fundo.

foreground (fg) Cor do texto.

side Define em qual lado o objeto vai aparecer (RIGHT,LEFT,TOP,BOTTOM).

command Realiza uma operação.

padx Posiciona o widget horizontalmente tendo como referência a borda da direita e da esquerda em pixels.

pady Posiciona o widget verticalmente tendo como referência o topo e o inferior da borda em pixels.

pack() Exibe os widgets na tela.

Vamos ver um exemplo:

```
1. from tkinter import *
2.
3. tela = Tk() #instancia a classe 'Tk()' através da variável 'tela'
4. tela.title("TELA INICIAL")
5. tela["bg"] = "light yellow"
6. tela.geometry("200x200") #define o tamanho da tela em width (largura) x height (altura)
```

Na linha 1, chamamos a biblioteca **tkinter** importando todos os recursos dessa biblioteca.

Na linha 3, criamos uma variável, como um objeto, para receber a classe **TK()**.

Nas linhas 4, 5 e 6, definimos um título para a tela (“TELA INICIAL”), uma cor de fundo e o tamanho da tela.

Vamos continuar incluindo mais alguns objetos:

```
1. from tkinter import *
2.
3. tela = Tk() #instancia a classe 'Tk()' através da variável 'tela'
4. tela.title("TELA INICIAL")
5. tela["bg"] = "light yellow"
6. tela.geometry("200x200") #define o tamanho da tela em width (largura) x height (altura)
7.
8. texto = Text(height=1, width=15)
9. texto.pack()
10.
11. btOK = Button()
12. btOK["text"] = "OK"
13. btOK["font"] = ("Calibri", "10", "bold")
14. btOK["width"] = 12
15. btOK["bg"] = "blue"
16. btOK.pack()
17.
18. btsair = Button()
19. btsair["text"] = "SAIR"
20. btsair["font"] = ("Calibri", "10", "bold")
21. btsair["width"] = 5
22. btsair["command"] = quit
23. btsair.pack()
24.
25. tela.mainloop()
```

Na linha 8, criamos um campo de texto, definindo a altura e largura.

A partir da linha 11, criamos um botão, definindo um nome (“OK”), com a fonte e tamanho da fonte. Definimos também a largura [**“width”**] = **12** e a cor do botão: [**“bg”**] = **“blue”**.

A partir da linha 18, criamos um botão SAIR, com a fonte e tamanho da fonte. Passamos um comando de sair para o botão, com a função **command**: [**“command”**] = **quit**, ou seja, quando o usuário clicar no botão SAIR, a tela será fechada.

A linha 25, tem um loop principal que faz com que toda a aplicação fique em execução até o usuário encerra-la.

Agora vamos incluir mais um objeto, um container:

```

1. from tkinter import *
2.
3. tela = Tk() #instancia a classe 'Tk()' através da variável 'tela'
4. tela.title("TELA INICIAL")
5. tela["bg"] = "light yellow"
6. tela.geometry("200x200") #define o tamanho da tela em width (largura) x height (altu
   ra)
7.
8. Container = Frame() #definimos o container com um frame
9. Container["pady"] = 10 #posiciona o container na tela a partir do top e bottom
10. Container["bg"] = "light yellow" # define uma cor de fundo
11. Container.pack() #para exibir os widgets do container
12.
13. texto = Text(Container)
14. texto["height"] = 1
15. texto["width"] = 15
16. texto.pack()
17.
18. btOK = Button(Container)
19. btOK["text"] = "OK"
20. btOK["font"] = ("Calibri", "10", "bold")
21. btOK["width"] = 12
22. btOK["bg"] = "blue"
23. btOK.pack()
24.
25. btsair = Button()
26. btsair["text"] = "SAIR"
27. btsair["font"] = ("Calibri", "10", "bold")
28. btsair["width"] = 5
29. btsair["command"] = quit
30. btsair.pack()
31.
32. tela.mainloop()

```

Na linha 8, criamos um **container** e agora colocamos os widgets: caixa de texto e o botão “OK” neste **container** (linhas 13 e 18). Já o botão SAIR ficou fora do container.

Ao executar o código acima, a tela deve ficar como na imagem abaixo:



Neste tópico podemos observar que a interface gráfica parte da interface por linha de comando (CLI), ou seja, utilizamos linhas de comandos para gerar a interface gráfica.

SAIBA MAIS...

Dê uma olhada nos links abaixo para saber mais sobre a linguagem Python:

<https://www.python.org/doc/> (<https://www.python.org/doc/>)

<https://wiki.python.org/moin/PythonBooks>
(<https://wiki.python.org/moin/PythonBooks>)

Neste tópico vimos como o Python pode trabalhar no modo gráfico (GUI) através do seu módulo nativo: o Tkinter.

Vimos também algumas funções para produzirmos uma tela gráfica que apresenta alguns objetos como botões e caixa de texto.

Quiz

Exercício Final

Programação em GUI com Tkinter

INICIAR ➤

Referências

SUMMERFIELD, M. *Programação em Python 3: Uma introdução completa à linguagem Python*. Rio de Janeiro Alta Books, 2012. 495 p.

MENEZES, N. N. C. *Introdução à programação com Python: algoritmos e lógica de programação para iniciantes*. 2. ed. São Paulo: Novatec, 2014. 328 p.

SWEIGART, AL. *Automatize tarefas maçantes com Python*: programação prática para verdadeiros iniciantes. São Paulo: Novatec, 2015. 568 p.

PYTHON, doc. Disponível em: <<https://www.python.org/doc/>>. Acesso em: Junho/2018.

PYTHON, books. Disponível em: <<https://wiki.python.org/moin/PythonBooks>>. Acesso em: Junho/2018.



Avalie este tópico



ANTERIOR

Multi-threading



Índice

Biblioteca

([https://www.uninove.br/conheca-](https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site

Criando uma tela de login recebendo dados do usuário

© Todos os direitos reservados

Ajuda?

PRÓXIMO
([https://ava.un](https://ava.uninove.br/seu/AVA/topico/topico.php)

de dados do usuário)

usuário

