

8-Noções Básicas de Tkinter

April 20, 2021

1 NOÇÕES BÁSICAS DE TKINTER

O Tkinter, apesar de não ser necessariamente uma biblioteca fundamental para o desenvolvimento dos simuladores, se destaca como um framework capaz de proporcionar o desenvolvimento de interfaces gráficas de usuário (GUI – Graphic User Interface) no python de forma prática, uma vez que é de fácil usabilidade e contém múltiplos recursos. Assim, é possível criar janelas capazes de receber os dados inseridos pelo usuário, tornando os simuladores mais atraentes para os estudantes, já que estão adaptados a esse tipo de configuração proporcionada pelos programas atuais.

2 Importando a Biblioteca Tkinter

```
[ ]: from tkinter import * #Sempre importar para utilizar os elementos do Tkinter
```

3 Construindo a Janela

Para o desenvolvimento da GUI, é preciso criar a janela onde os elementos vão ser inseridos. Para tanto, basta atribuir a função principal do Tkinter Tk() em uma variável responsável por orientá-la, e, a partir dela, definir parâmetros de título title, dimensão e posição dessa janela na tela geometry(larguraXaltura+x+y). Além disso, para que ela fique visível permanentemente, é preciso inserir um loop ao final mainloop.

```
[ ]: janela = Tk()  
janela.title('Título da Janela Principal') #Título da janela  
janela.geometry('270x120+500+200') #Posição e tamanho da janela  
janela.mainloop() #loop
```

Reinicie o kernel e limpe as saídas (Kernel -> Restart & Clear Output, se estiver no Jupyter Notebook;) e comece a execução dos códigos a partir da célula abaixo para uma melhor facilidade na visualização

3.1 Definindo Elementos na Janela

Com a janela pronta, a segunda etapa é definir os elementos que aparecerão nela, podendo ser rótulos (label) para definir comandos de saída (escritos em seu parâmetro text), caixas (entry) para receber os dados inseridos pelo usuário e botões (button) para executar eventos após serem clicados. Nesses elementos, a variável associada a janela **sempre deverá ser colocada como um parâmetro**. Além disso, podem ser utilizados outros parâmetros, como a altura do elemento na tela (height) e a fonte de seu texto (font = estilo tamanho)

(Aqui estamos *definindo os elementos*, eles só aparecerão na tela após padronizarmos seu layout)

```
[ ]: from tkinter import * #Sempre importar para utilizar os elementos do Tkinter

#Criando a janela
janela = Tk()
janela.title('Título da Janela Principal') #Titulo da janela
janela.geometry('270x120+500+200') #Posição e tamanho da janela

#Definindo os elementos da janela
lb1 = Label(janela, text='Insira seu nome: ', font='times 14 ', height=3)
ed1 = Entry(janela)
bt = Button(janela, width = 20, text='Registrar')

janela.mainloop() #loop
```

Reinicie o kernel e limpe as saídas (Kernel -> Restart & Clear Output, se estiver no Jupyter Notebook;) e comece a execução dos códigos a partir da célula abaixo para uma melhor facilidade na visualização

3.2 Inserindo Elementos na Janela - Gerenciador de Layout

Após essa etapa, esses elementos devem ser distribuídos na interface através do gerenciador de layout: ferramenta responsável por mapear os elementos na interface. Dentre os principais gerenciadores, se destacam o **place** com o sistema de coordenadas cartesianas, **pack** com o sistema de orientação nas quatro direções e **grid** com o sistema de células, podendo ser escolhidos de forma opcional.

Nesses simuladores, optamos pelo **grid**, tendo em vista que é amplamente utilizado no python para a criação de GUI. De forma geral, os elementos são distribuídos na janela de forma similar à distribuição de células em uma tabela, tendo a coluna **column** e linha **row** inicial no valor de 0 (zero). Além disso, outro parâmetro importante é o **columnspan** para mesclar colunas e **rowspan** para mesclar linhas.

```
[ ]: from tkinter import *

janela = Tk()
janela.title('Título da Janela Principal') #Titulo da janela
janela.geometry('270x120+500+200') #Posição e tamanho da janela

#Definindo os elementos da janela
lb1 = Label(janela, text='Insira seu nome: ', font='times 14 ', height=3)
ed1 = Entry(janela)
bt = Button(janela, width = 20, text='Registrar')

#Inserindo os elementos da janela
lb1.grid(row=0, column=0)
ed1.grid(row=0, column=1)
bt.grid(row=2, column=0, columnspan = 2)
```

```
janela.mainloop()
```

Reinicie o kernel e limpe as saídas (Kernel -> Restart & Clear Output, se estiver no Jupyter Notebook;) e comece a execução dos códigos a partir da célula abaixo para uma melhor facilidade na visualização

4 Capturando os Valores Iseridos pelo Usuário

O ultimo passo é capturar os dados inseridos pelo usuário a partir das caixas de entrada. Para isso, é preciso definir um evento/função em que, ao clicar no botão, os dados sejam enviados para uma variável por meio de uma função capaz capturá-los. Nesse caso, o evento é associado ao parâmetros `command`, que definirá a função para o tratamento dos dados (aqui a função foi nomeada de `bt_clic`). Por fim, os dados são capturados por meio da função `get()`- cabe lembrar que a função `get()` captura dados do tipo carácter, sendo necessário covertê-los para real (`float()`) ou inteiro (`int()`) caso deseje realizar operações com esses números (ver tutorial 3-Noções Básicas de Python).

```
[ ]: from tkinter import *

def bt_clic():
    nome=ed1.get()
    print('Olá, ', nome, '!\\nNome Registrado!')

janela = Tk()
janela.title('Título da Janela Principal') #Título da janela
janela.geometry('270x120+500+200') #Posição e tamanho da janela

lb1 = Label(janela, text='Insira seu nome: ', font='times 14 ', height=3)
ed1 = Entry(janela)
bt = Button(janela, width = 20, text='Registrar', command = bt_clic)

lb1.grid(row=0, column=0)
ed1.grid(row=0, column=1)
bt.grid(row=2, column=0, columnspan = 2)

janela.mainloop()
```

Esses dados podem ser tratados da forma desejada, podendo, a partir do VPython, construir os simuladores com uma GUI amigável.

Para mais informações sobre o TKinter, ver: <https://docs.python.org/3/library/tkinter.html>

```
[ ]:
```