



# **Curso completo de Python!**





**Instrutor: Vitor Mazuco**

**<http://facebook.com/vitormazuco>**

**Email: [vitor.mazuco@gmail.com](mailto:vitor.mazuco@gmail.com)**

**WebSite:**

**<http://vmzsolutions.com.br>**



## Adicionando vários widgets

Podemos trabalhar com mais de um **widget** em nosso tkinter.

Este nosso arquivo cria um widget Frame (outra classe tkinter) e anexa três outros objetos widget a ele, um Label e dois Botões, passando o Frame como seu primeiro argumento. No tkinter, podemos dizer que o Frame se torna um “pai” para os outros três widgets.

\* Veja mais no arquivo gui4.py



## Adicionando vários widgets

Ambos os botões neste display acionam callbacks:

- Pressionar o botão 'Olá' aciona a função de saudação definida neste arquivo, que imprime para *stdout* denovo.
- Ao apertar o botão 'Sair' chama o método de encerramento padrão do tkinter, herdado da classe Frame ('Frame.quit' tem o mesmo efeito que o 'Tk.quit' usado antes).



## Adicionando vários widgets

Podemos alterar a ordem de onde aparecem os botões, para isso é só comentar as dimensões e mudar essas linhas abaixo:

```
Label(win, text='Olá container').pack(side=RIGHT)
```

```
Button(win, text='Sair', command=win.quit).pack(side=TOP)
```



## Adicionando vários widgets

Podemos deixar todo os nossos botões ocupados pelo espaço, para isso altera para as seguintes linhas:

```
Button(win, text='Olá', command=greeting).pack(side=LEFT, fill=Y)
```

```
Label(win, text='Olá container').pack(side=TOP)
```

```
Button(win, text='Sair', command=win.quit).pack(side=RIGHT,  
expand=YES,fill=X)
```



## Adicionando vários widgets

Um outro recurso bacana é o uso do âncora que aceita as constantes do tkinter identificando todos os oito pontos da bússola (N, NE, NW, S, etc.) e CENTER como seu valor (por exemplo, âncora = NW)

```
Button(win, text='Olá', command=greeting).pack(side=LEFT, anchor=N)
```