Curso de TkInter - Interface Gráfica para Python 3

-

Indice:

12.1 -- Recuperar valor de uma combobox

13 -- Adicionar um CheckButton a janela (Tkinter checkbox)

13.1 -- Recuperar Estado do CheckButton

14 -- RadioButton - (Botão de escolha redondo)

15 -- texto com scroll lateral - text area

15.1 -- Inserir texto na textbox

15.2 -- Limpar TextBox / Input

16 -- Criar um alerta - MessageBox

16.1 -- Outros tipos de alertas

16.2 -- messageBox com perguntas

17 - SpinBox (Selecionador de números)

17.1- Definir um valor padrão para o Spinbox

18-Barra de Progresso

18.1-Mudar cor da barra de progresso

19- Abrir Arquivos com tkinter - File Dialog

19.1- Definir tipos de arquivos para abrir

20-Adicionar uma barra superior

21-Adicionar separador por Abas

21.1 Adicionar widgets as abas

22 - Espaçamento padding

23 - Prevenir o redimensionamento da janela

24 - Adicionar borda a um frame

25 - Inserir imagens na label

.

Neste tutorial você ira aprender a como desenvolver interfaces gráficas para usuário com a linguagem python.
O pacote TKinter vem instalado junto com o python em suas versões atualizadas e não necessita ser instalado separadamente.

Se você esta utilizando a ide que vem junto da instalação padrão do python o IDLE saiba que ele foi feito utilizando TKinter! então da pra ter uma ideia de como o pacote TKinter é poderoso para criação de interfaces gráficas.

É recomendado que você já saiba o básico em linguagem python antes de prosseguir com este tutorial.

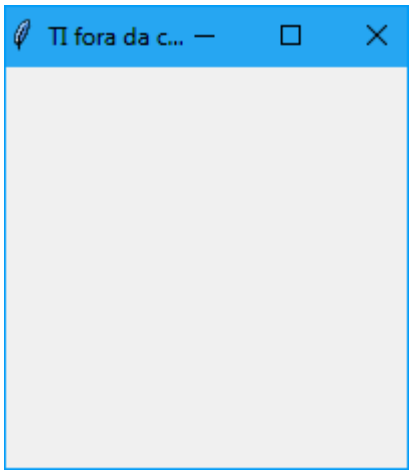
1 - Nossa Primeira Janela...

Vamos começar importando o pacote TKinter, criando uma janela e definindo o seu titulo.

Exemplo:

```
from tkinter import * #importamos o pacote tkinter

window = Tk()#instanciamos a classe tk

#a variavel window agora é uma instancia da classe tk

#o nome da variavel não fara diferença!

#definimos o titulo da janela

window.title("TI fora da caixa")

#colocamos a janela em loop para ela permanecer aber
to.

window.mainloop()
```

O resultado sera algo semelhante a isso:



Parabéns! Sua primeira janela em tkinter esta funcionando!

Na ultima linha do codigo existe o seguinte comando:

`window.mainloop()`

Este comando diz para a window entrar em um loop e que a janela permaneça aberta enquanto o mainloop() estiver ativo / enquanto o usuario não a fechar.
Este comando `window.mainloop()` deve ser o ultimo a ser executado no codigo da janela!

2 - Criando uma Label - Widget Label

Para este exemplo podemos reultilizar o codigo da janela do exemplo anterior. basta apenas adicionar o seguinte codigo a baixo da linha

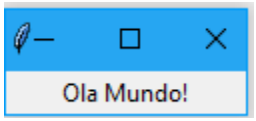
Código: `Texto = Label(window, text="ola Mundo)`

Exemplo:

```
from tkinter import *



window = Tk()

window.title("TI fora da caixa")

texto = Label(window, text="Ola Mundo!")

texto.pack()#adicionamos o widget a janela



#Label é o nome do widget e texto é a variavel que guarda a label

#a label esta recebendo dois parametros

#o primeiro(window) é o nome da janela na qual a label sera inserida

#o segundo (text="Ola mundo!") é a string que queremos exibir na label

#na linha texto.pack()estamos posicionando o widget na janela

#veja mais sobre isso na sessão de posicionamentos!
```

```
window.mainloop()
```

Ao executar salvar e executar esse código o resultado sera o seguinte:




```
##########################################################
##########
```

3 - Alterando Widgets.

Cada Widget do tkinter tem um formato padrão
ao passarmos parâmetros ao instanciarmos em variáveis estamos apenas alterando esse padrão
o padrão dos Widgets esta escrito na forma de dicionários
por isso se quisermos alterar o padrão podemos tanto na instanciação modificá-los, por ex:

```
texto = Label(window, text="Ola Mundo!")
```

ou então alterá-lo como uma chave de array associativo
`texto["text"] = "este é o novo texto"`
Exemplo no código:

```
from tkinter import *


window = Tk()

window.title("TI fora da caixa")

texto = Label(window, text="Ola Mundo!")

#alteramos o texto

texto["text"] = "este texto sera exibido"

#adicionamos o widget a janela

texto.pack()


window.mainloop()
```

##
##########

4 - Posicionamentos:

O tkinter possui 3 diferentes gerenciadores de layout
eles servem para posicionar os widgets de nossa janela e distribui-los.
veja mais sobre eles nas seguintes postagens:

1-Gerenciador de layout place
2-Gerenciador de layout pack
3-Gerenciador de layout grid

##
#########

5 - Mudar fonte da Label:

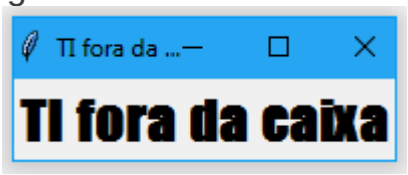
Você pode modificar a fonte da label de formas bem simples
para definir a fonte ao criar a label você pode passar o
seguinte parâmetro:

```
font="nomeDaFonte TamanhoDaFonte EfeitoDaFonte"

ex: font="impact 20 bold"
```

Ex de uso:

```
from tkinter import *



window = Tk()

window.title("TI fora da caixa")

#ao criarmos a label passamos o parâmetro font

texto = Label(window, text="TI fora da caixa", font="impact
20 bold")

texto.pack()

window.mainloop()
```

O resultado desse código sera esse:



##
##########

6-Modificar Tamanho da Janela.

O elemento que guarda o tamanho da janela é o window.geometry
Para alterar o tamanho da janela basta passar como parametro uma string com o novo tamanho por ex:

```
window.geometry('800x600')
```

```
window.geometry('alturaXlargura')
```

ao adicionar esse codigo ao nosso programa ele fará com que o programa fique com 800 pixels de largura e 600 de altura

podemos ainda definir a posição na qual o programa sera exibido ao executar basta apenas adicionar ao parametro apos os tamanhos os valores para somar aos lados x e y das bordas do programa

por exemplo:

```
window.geometry('800x600+100+50')
```

```
window.geometry('alturaXlargura+MargemEsquerda+MargemTopo')
```

##
##########

7-Adicionar Botão a tela.

o codigo que utilizaremos para adicionar um botão a janela é o seguinte:

```
botao = Button(window, text="Clique Aqui")
```

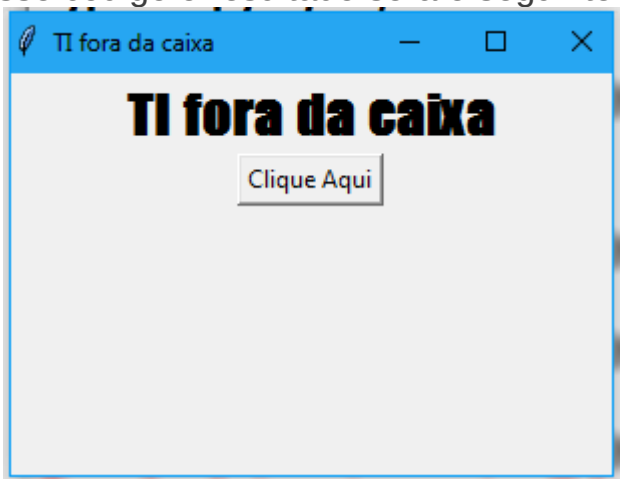
```
botao.pack()
```

Exemplo de uso:

```
from tkinter import *
window = Tk()
window.geometry("300x200+200+100")
```

```
window.title("TI fora da caixa")
texto = Label(window, text="TI fora da caixa", font="impact 20 bold")
texto.pack()
botao = Button(window, text="Clique Aqui")
botao.pack()
window.mainloop()
```

E ao executar esse codigo o resultado sera o seguinte:



##
##########

7.1-Modificar cor de fundo e da fonte do botão.

Para fazer isso basta alterar os parametros bg e fg do botão

ex:

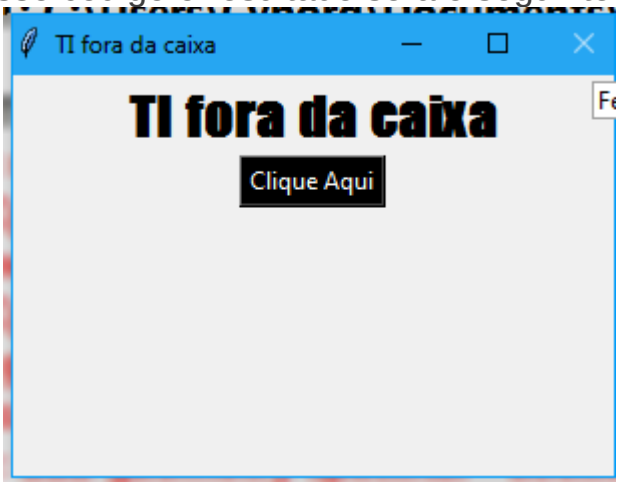
```
botao = Button(window,
text="Clique!",bg="black",fg="white")
```

exemplo no codigo:

```
from tkinter import *

window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")

texto = Label(window, text="TI fora da caixa", font="impact
20 bold")

texto.pack()

botao = Button(window, text="Clique Aqui", bg="black", fg="
white")
```

```
botao.pack()

window.mainloop()
```

E ao executar esse código o resultado sera o seguinte:



##
##########

8 - Executar função ao clicar em botão.

Para executar uma função ao clicar em um botão
o botão tem que receber como parametro o nome da função que ele deve executar
o nome da função deve ser atribuido ao parametro **command** do botão

O NOME DEVE SER SEM OS PARENTESES!!

```
botao = Button(window,
text="Clique!",command=aoClicar)
```

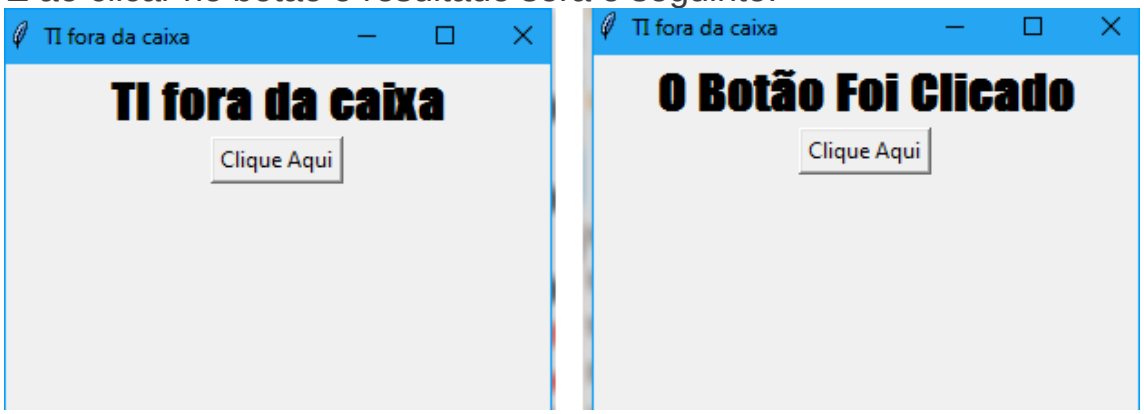
observe o seguinte codigo

```
from tkinter import *

def aoClicar():

    mensagem["text"]="O Botão Foi Clicado"



window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")
```

```
mensagem = Label(window, text="TI fora da caixa", font="imp
act 20 bold")

mensagem.pack()

botao = Button(window, text="Clique Aqui", command=aoClicar
)

botao.pack()

window.mainloop()
```

ao clicar no botão a função aoClicar() é executada
e essa função altera o texto da label mensagem

para passar uma função como parametro ela deve ter sido criada antes do botão!
E ao clicar no botão o resultado sera o seguinte:



antes e depois de clicar

```
##################################################
##########
```

9 - Entrada de informações com Entry widget.

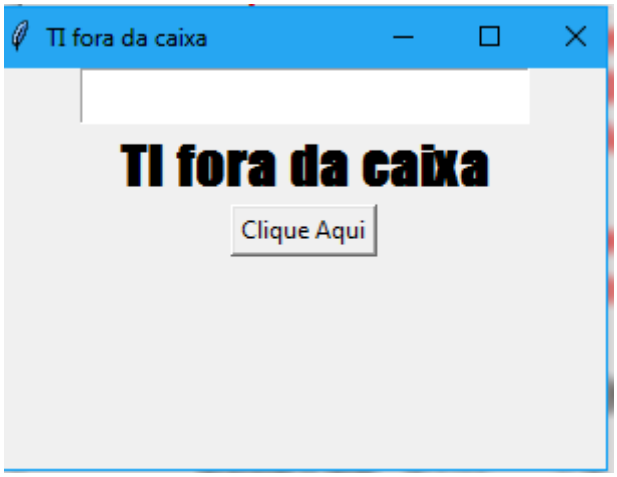
o widget entry pode receber como parametro a fonte do texto que vai ser inserido!

Exemplo de janela com entrada de texto:

```
from tkinter import *


def aoClicar():

    mensagem["text"]="texto: "+entrada.get()
```

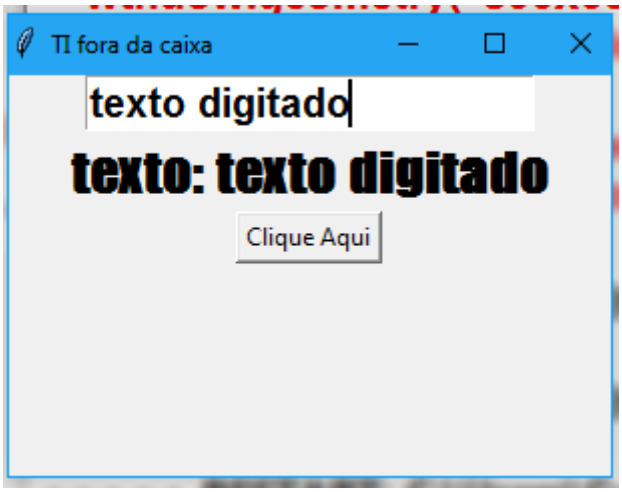
```
window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")

entrada = Entry(window, font="arial 15 bold")

entrada.pack()

mensagem = Label(window, text="TI fora da caixa", font="imp
act 20 bold")

mensagem.pack()

botao = Button(window, text="Clique Aqui", command=aoClicar
)

botao.pack()

window.mainloop()
```

Ao executarmos esse código será aberta a seguinte janela:



e caso você digite um texto e clique no botão a função aoClicar() sera executada

nesse código essa função alem de alterar o texto da label mensagem
também recupera o texto da Entry entrada e o concatena no texto que substitui o da Label mensagem
ficando com o seguinte formato:



o texto digitado só substitui o da label após clicar no botão!

##
##########

10-Focar o cursor em um widget ao executar o programa.

Isso é bem simples basta adicionar antes da linha `window.mainloop()`

o comando widgetAlvoDoFoco.focus()

por exemplo:

```
from tkinter import *


window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")

entrada = Entry(window, font="arial 15 bold")

entrada.pack()

entrada.focus()

window.mainloop()
```

Ao executar esse codigo o programa ja vai ser iniciado com o foco na Entry entrada

```
##################################################################
##########
```

11-Desabilitar uma entrada de texto.

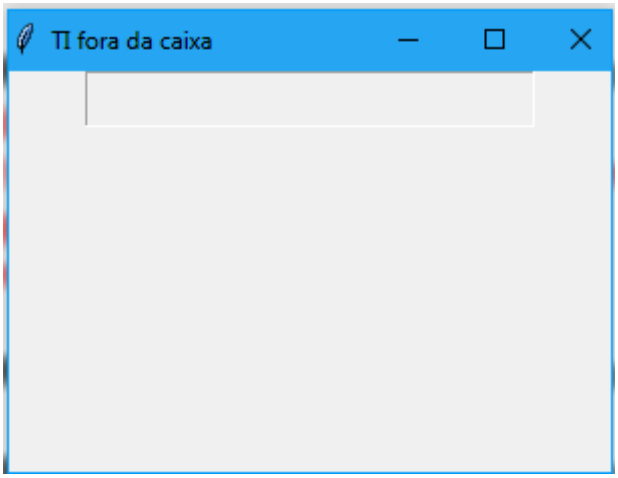
basta modificar o parametro da entrada state para desabled
ex: `entrada["state"]="disabled"`
ou ao criar a label:

```
entrada = Entry(window, font="arial 15 bold", state="disabled")
```

Exemplo de uso:

```
from tkinter import *


window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")

entrada = Entry(window, font="arial 15 bold")

entrada.pack()

entrada["state"]="disabled"

window.mainloop()
```

e o resultado sera esse:




```
##################################################################
##########
```

12 criar uma combobox

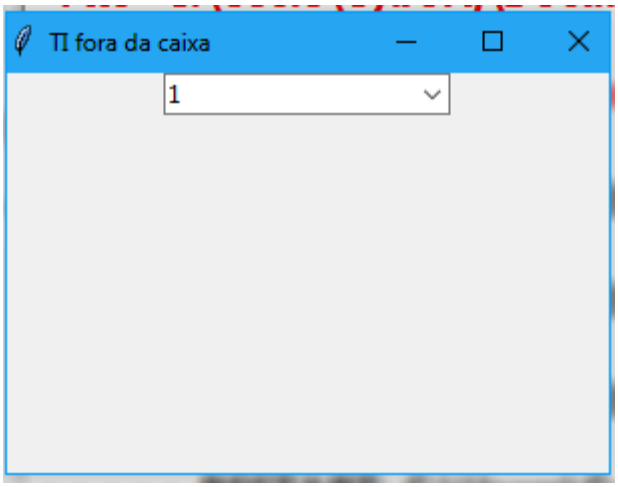
uma combobox é uma lista de parâmetros pré-definidos no qual podemos selecionar um.
(funciona igual a um input type select do html)

Para ultilizar a combobox precisamos importar tudo do modulo tkinter.ttk
com o comando: from tkinter.ttk import *
e instanciamos com o comando: combo = Combobox(window)

Exemplo de uso:

```
from tkinter import *
from tkinter.ttk import *


window = Tk()
window.geometry("300x200+200+100")
window.title("TI fora da caixa")


combo = Combobox(window)
combo['values']= (1, 2, 3, 4, 5, "Text")
combo.current(0) #definimos o valor padrão par ser exibido!
combo.pack()
window.mainloop()
```

E o resultado sera esse:



##
##########

12.1 Recuperar valor de uma combobox

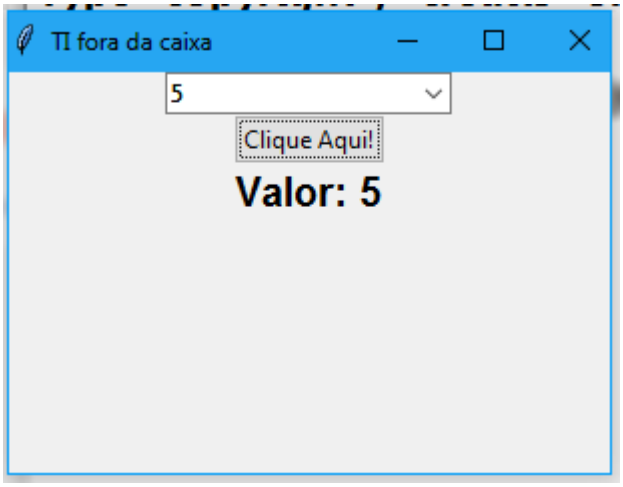
para recuperar um valor selecionado em um combobox basta usar a função get()
por exemplo: selecionado = combo.get()

exemplo de uso:

```
from tkinter import *
from tkinter.ttk import *


def exibeValor():
    mensagem["text"]="Valor: "+combo.get()


window = Tk()
window.geometry("300x200+200+100")
window.title("TI fora da caixa")
combo = Combobox(window)
combo['values']= (1, 2, 3, 4, 5, "Text")
combo.current(0) #definimos o valor padrão!
combo.pack()
```

```
botao = Button(window, text="Clique Aqui!", command=
exibeValor)

botao.pack()

mensagem = Label(window, text="Valor: ", font="arial
15 bold")

mensagem.pack()

window.mainloop()
```

Resultado:



##
##########

13-Adicionar Checkbutton na janela(Tkinter checkbox)

Para criar um checkbutton no tkinter, você pode usar a classe Checkbutton dessa forma:

```
botaoMarcavel = Checkbutton(window, text='Marcar: ')
```

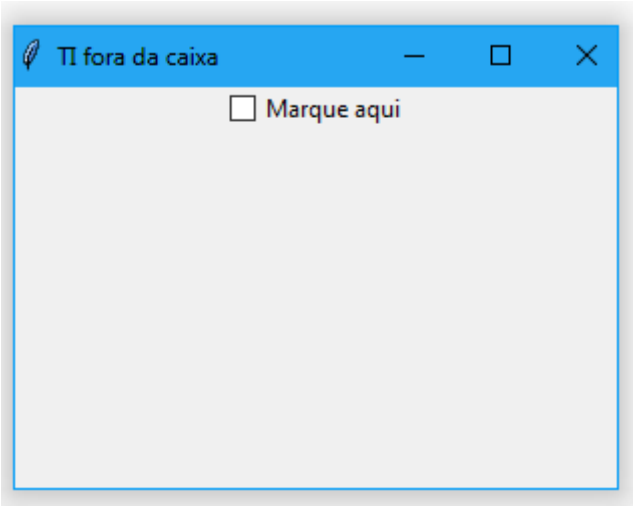
O checkbutton retorna um valor booleano, ao marcar o checkbutton o seu valor fica True, ao desmarcar seu valor muda para False.

```
from tkinter import *
```

```
from tkinter.ttk import *



window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")



botaoMarcavelStatus = BooleanVar()

botaoMarcavelStatus.set(False) #Definimos um status
padrã#o para o botão

botaoMarcavel = Checkbutton(window, text='Marque aqu
i', var=botaoMarcavelStatus)

botaoMarcavel.pack()

window.mainloop()
```

A Variável botaoMarcavelStatus guarda o estado booleano do checkbutton
caso marcado seu valor ser True e desmarcado sera False.

Oresultado sera esse:



##
##########

13.1-Recuperar Estado do CheckButton

Podemos recuperar o estado do estado do checkbutton ultilizando o metodo get() na variavel botaoMarcavelStatus o código sera:

```
estado = botaoMarcavelStatus.get()
```

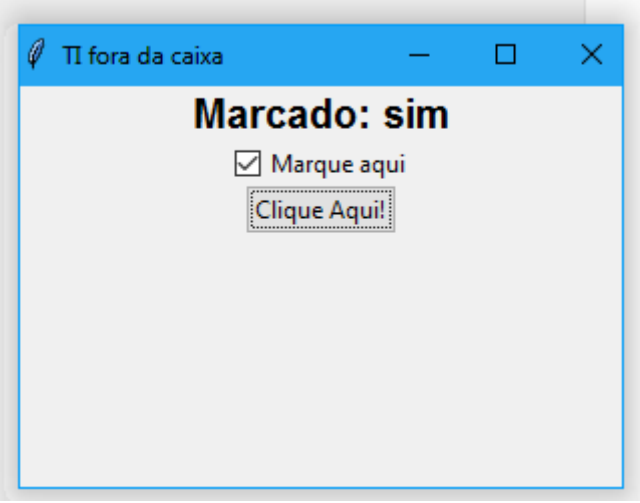
Exemplo de uso:

```
from tkinter import *

from tkinter.ttk import *



def recuperarEstado():

    if botaoMarcavelStatus.get() ==True:
```

```
        valor = "sim"

    else:

        valor = "não"

    mensagem["text"]="Valor: "+str(valor)



window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")



mensagem = Label(window, text="Marcado: ", font="ari
al 15 bold")

mensagem.pack()



botaoMarcavelStatus = BooleanVar()

botaoMarcavelStatus.set(False) #Definimos um status
padrão para o botão
```

```
botaoMarcavel = Checkbutton(window, text='Marque aqui', var=botaoMarcavelStatus)


botaoMarcavel.pack()



botao = Button(window, text="Clique Aqui!", command=recuperarEstado)


botao.pack()



window.mainloop()
```

o resultado sera esse:




```
##################################################################
##########
```

14-RadioButton - Botão Selecionavel

Para criar um radio button basta instanciar a clsse `Radiobutton` da seguinte forma:

```
escolha1 = Radiobutton(window,text='PrimeiraOp', value=1)
```

Podemos guardar o valor do botão escolhido em uma variavel passando um parametro variable na criação do RadioButton,
Ex:

```
escolha = StringVar()#essa variavel guarda o valor definido
```

```
escolha1 = Radiobutton(window,text='PrimeiraOp', value="Primeira", variable = escolha)
```

```
#a variável que guarda o valor do botão escolhido é passada também como parâmetro
```

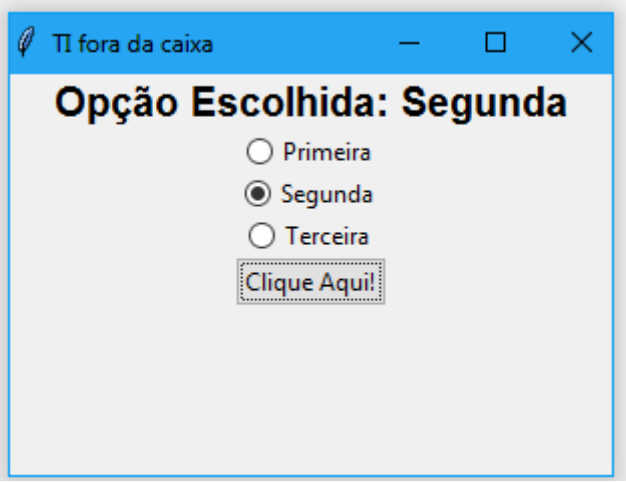
Exemplo de uso:

```
from tkinter import *

from tkinter.ttk import *


def exibeValor():

    mensagem["text"]="Opção Escolhida: "+str(escolha.get())
```

```
window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")



mensagem = Label(window, text="Opção Escolhida: Nenh
uma", font="arial 15 bold")

mensagem.pack()

escolha = StringVar()#Guarda o valor escolhido

#radioButtons:

escolha1 = Radiobutton(window,text='Primeira', value
='Primeira', variable = escolha)

escolha2= Radiobutton(window,text='Segunda', value='
Segunda', variable = escolha)

escolha3 = Radiobutton(window,text='Terceira', value
='Terceira', variable = escolha)



escolha1.pack()

escolha2.pack()
```

```
escolha3.pack()



botao = Button(window, text="Clique Aqui!", command=
exibeValor)


botao.pack()


window.mainloop()
```

E o resultado desse código sera o seguinte:



Os radiobuttons também podem ter um parâmetro command e executar uma função ao ser clicado, como por exemplo a função exibeValor()

```
##################################################
##########
```

15- texto com scroll lateral - text area

Para adicionar uma caixa de texto com scroll, você pode usar a classe ScrolledText dessa forma:

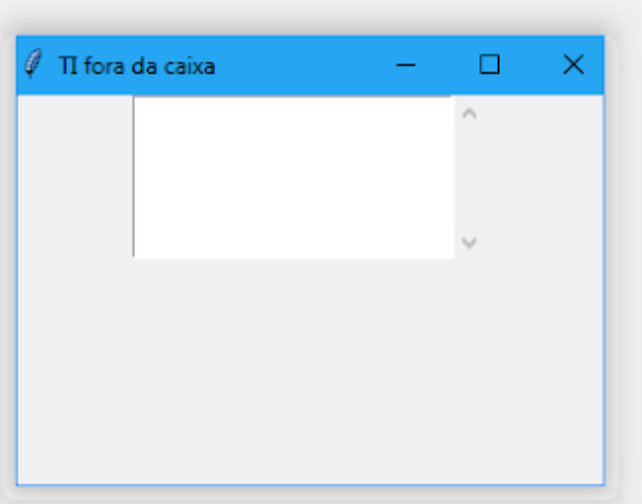
```
#importamos a classe:
```

```
from tkinter import scrolledtext

#e a instanciamos

texto=scrolledtext.ScrolledText(window,width=20,heig
ht=5)
```

Exeplo de uso:

```
from tkinter import *

from tkinter import scrolledtext

from tkinter.ttk import *



window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")



texto = scrolledtext.ScrolledText(window,width=20,he
ight=5)

texto.pack()



window.mainloop()
```

E o resultado sera esse:



##
##########

15.1-Inserir texto na textbox / Input:

Para adicionar um texto em uma textbox podemos usar a função insert:

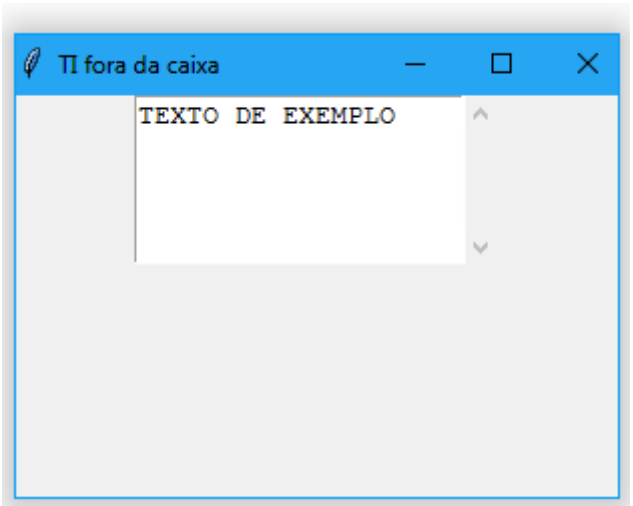
```
texto.insert(INSERT, "TEXTO DE EXEMPLO")
```

Exemplo de uso:

```
from tkinter import *

from tkinter import scrolledtext
```

```
from tkinter.ttk import *



window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")

texto = scrolledtext.ScrolledText(window,width=20,he
ight=5)

texto.insert(INSERT, "TEXTO DE EXEMPLO")



texto.pack()

window.mainloop()
```

E o resultado sera esse:



##
##########

15.2-Limpar TextBox / Input:

Para excluir todo o texto de uma textbox podemos usar a função delete(), exemplo de uso:

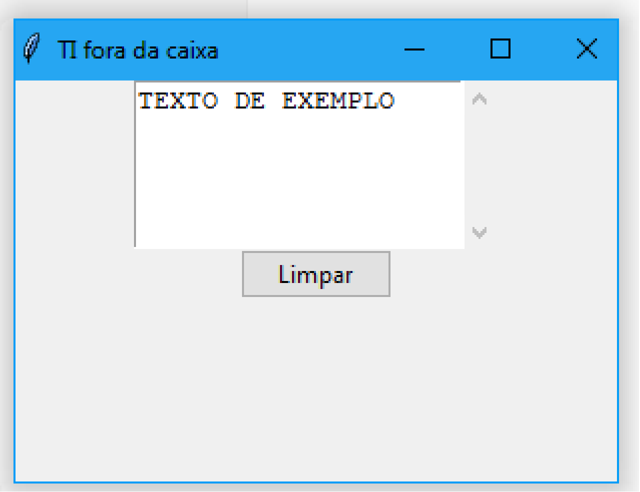
```
texto.delete(1.0, END)
```

Exemplo de uso:

```
from tkinter import *

from tkinter import scrolledtext

from tkinter.ttk import *


def limparTextbox():

    texto.delete(1.0,END)


window = Tk()

window.geometry("300x200+200+100")
```

```
window.title("TI fora da caixa")

texto = scrolledtext.ScrolledText(window,width=20,height=5)

texto.pack()

texto.insert(INSERT, "TEXTO DE EXEMPLO")



bt=Button(window, text="Limpar", command=limparTextbox)

bt.pack()



window.mainloop()
```

E ao executar esse código obtemos o seguinte resultado:




```
##################################################
```

16-Criar um alerta - MessageBox:

Para utilizar a messagebox precisaremos importar diretamente do pacote tkinter e utiliza**-la da seguinte forma:**

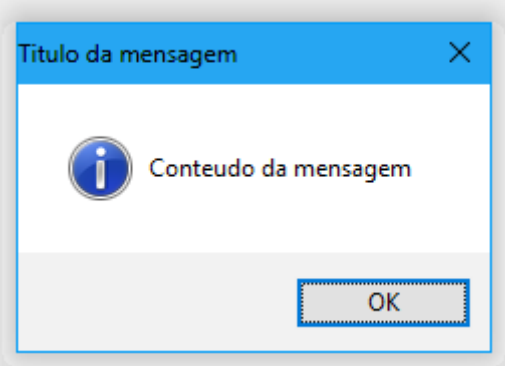
```
from tkinter import messagebox

messagebox.showinfo('Titulo da mensagem','Conteudo d
a mensagem')
```

Exemplo de uso:

```
from tkinter import *

from tkinter import messagebox



window = Tk()

window.geometry("300x200+200+100")

window.title("TI fora da caixa")



messagebox.showinfo('Titulo da mensagem','Conteudo d
a mensagem')
```

```
window.mainloop()
```

E ao executar obtemos o seguinte resultado:




```
##################################################
##########
```

16.1 - Outros tipos de alertas

Exemplos de mensagens:

```
from tkinter import messagebox

messagebox.showwarning('Message title', 'Message con
tent')  #Mostra um alerta


messagebox.showerror('Message title', 'Message conte
nt')    #mostra um erro

##################################################
##########
```

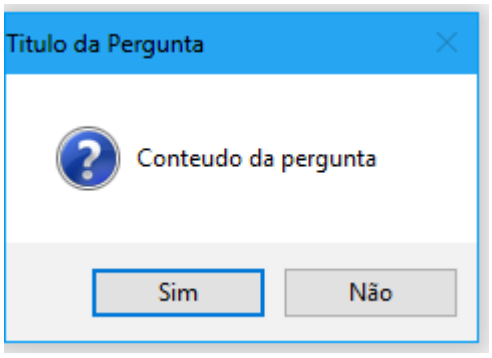
16.2 - messageBox com perguntas:

Exemplos:

```
from tkinter import messagebox
```

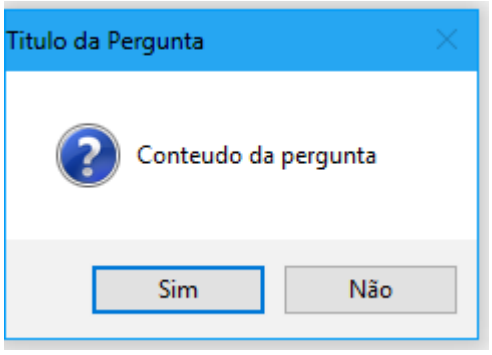
```
questao = messagebox.askquestion('Titulo da Pergunta
','Conteudo da pergunta')#pergunta se sim ou não
```

Ex:

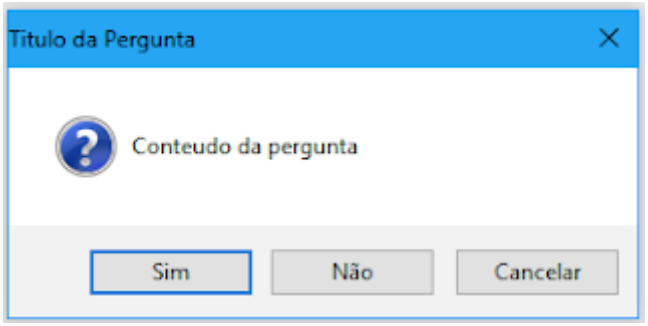



```
questao = messagebox.askyesno('Titulo da Pergunta','
Conteudo da pergunta')#pergunta se sim ou não
```

Ex:

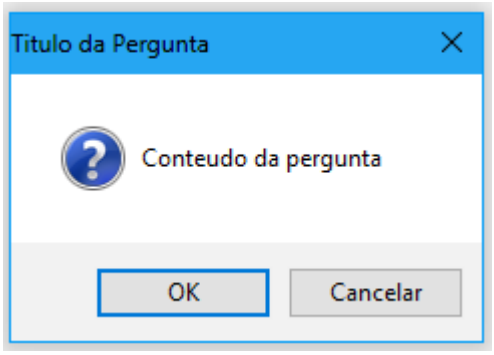


```
questao = messagebox.askyesnocancel('Titulo da Pergu
nta','Conteudo da pergunta')#pergunta se sim ou não
ou cancelar
```




```
questao = messagebox.askokcancel('Titulo da Pergunta
','Conteudo da pergunta')#pergunta se sim ou cancela
r
```

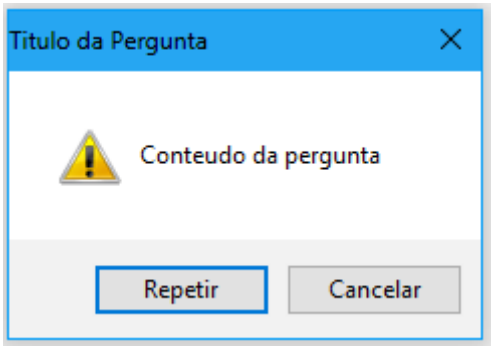
Ex:




```
questao = messagebox.askretrycancel('Titulo da Pergu
nta','Conteudo da pergunta')#pergunta se tentar ou c
```

ancelar

Ex:



##
##########

17 - SpinBox (Selecionador de números):

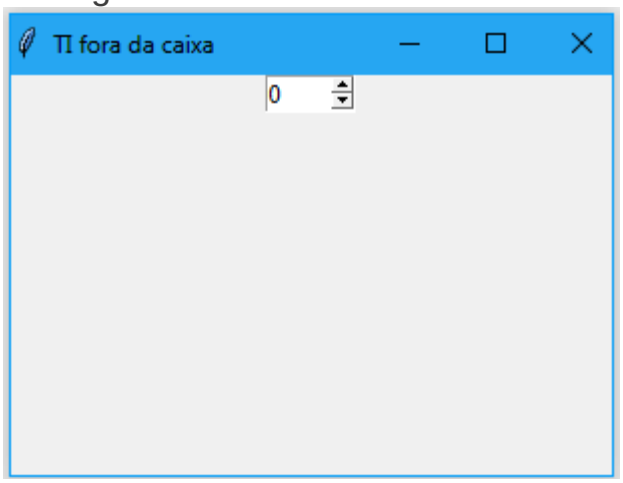
Exemplo:

spin = Spinbox(window, from_=0, to=100, width=5)

Exemplo de código:

```
from tkinter import *

window = Tk()
window.title("TI fora da caixa")
window.geometry("300x200")
spin = Spinbox(window, from_=0, to=100, width=5)
spin.pack()
window.mainloop()
```

Ao executar este código obtemos esse resultado:



Podemos ainda especificar uma lista de valores para o spinbox mostrar:

```
spin = Spinbox(window, values=(3, 8, 11), width=5)
```

Nesse caso (3, 8, 11) são os únicos valores que serão mostrados no spinbox

##

17.1- Definir um valor padrão para o Spinbox:

Para definir o valor padrão do spinbox, voce deve passar uma variavel com nome

textvariable como parâmetro para o spinbox dessa forma:

```
var =IntVar()


var.set(36)

spin=Spinbox(window, from_=0, to=100, width=5, textvariable=var)
```

Agora ao executar o programa o valor mostrado como padrao no spinbox sera o 36

##
##########

18-Barra de Progresso:

Podemos criar uma barra de progresso da seguinte forma:

```
from tkinter.ttk import Progressbar

barra = Progressbar(window, length=200)
```

E podemos definir a porcentagem da barra com o seguinte parâmetro:

```
barra['value'] = 70
```

##
##########

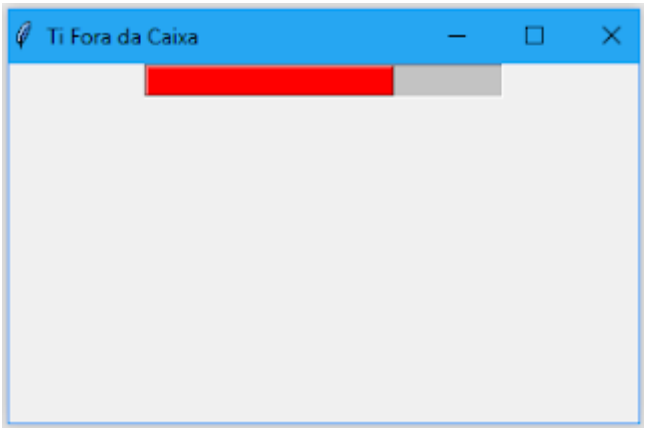
18.1-Mudar cor da barra de progresso:

```
from tkinter import *

from tkinter.ttk import Progressbar

from tkinter import ttk



window = Tk()

window.title("Ti Fora da Caixa")

window.geometry('350x200')

style = ttk.Style()

style.theme_use('default')

style.configure("red.Horizontal.TProgressbar",

background='red')

barra = Progressbar(window, length=200,

style='red.Horizontal.TProgressbar')
```

```
barra['value'] = 70
```

```
barra.pack()
```

O resultado sera:




```
##################################################
##########
```

19- Abrir Arquivos com tkinter - File Dialog:

Exemplo :

```
from tkinter import filedialog
```

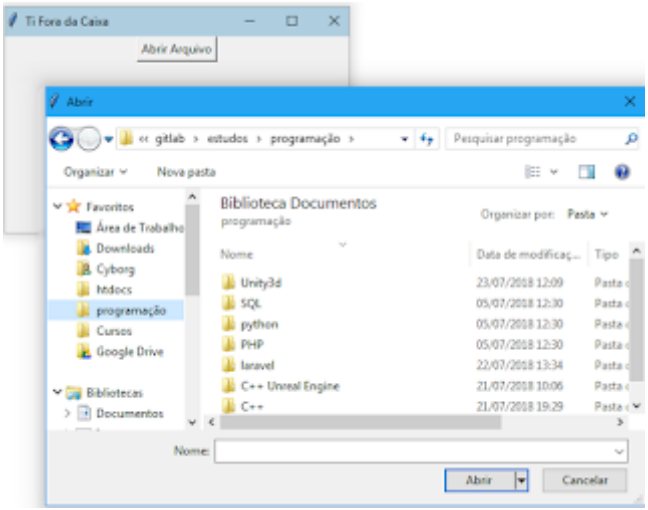
```
arquivo = filedialog.askopenfilename()
```

Exemplo de uso :

```
from tkinter import *
from tkinter import filedialog
```

```
def abrirArquivo():
    arquivo = filedialog.askopenfilename()


window = Tk()
window.title("Ti Fora da Caixa")
window.geometry('350x200')

botaoAbrir = Button(window, text="Abrir Arquivo", co
mmand = abrirArquivo)
botaoAbrir.pack()


window.mainloop()
```

O resultado ao clicar no botão será:



##
##########

19.1- Definir tipos de arquivos para abrir:

Você pode especificar os tipos de arquivos que o programa deve abrir com os seguintes parâmetros:

```
arquivo = filedialog.askopenfilename(filetypes = (("Text files","*.txt"),("all filles","*.*")))
```

Podemos também abrir pastas com o seguinte comando:

```
pasta = filedialog.askdirectory()
```

e podemos definir a pasta onde queremos abrir com o comando:

```
from os import path

file = filedialog.askopenfilename(

initialdir=path.dirname(__file__))
```

```
##########################################################
##########
```

20-Adicionar uma barra superior:

Siga este exemplo para adicionar uma barra de menu superior:

```
from tkinter import Menu

menu = Menu(window)

menu.add_command(label='Arquivo')
```

```
window.config(menu=menu)
```

e voce pode adicionar sub itens ao itens do menu com o seguinte codigo:

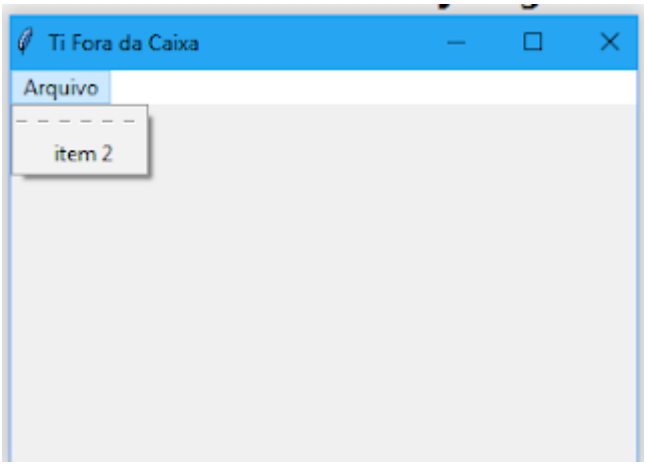
```
menu.add_cascade(label='File', menu=new_item)
```

Seu codigo deve ser algo parecido com isso:

```
from tkinter import *

from tkinter import filedialog

from tkinter import Menu



window = Tk()

window.title("Ti Fora da Caixa")
```

```
window.geometry('350x200')



menu = Menu(window)

window.config(menu=menu)

new_item = Menu(menu)

new_item.add_command(label='item 2')

menu.add_cascade(label='Arquivo', menu=new_item)

window.mainloop()
```

e o resultado sera esse:



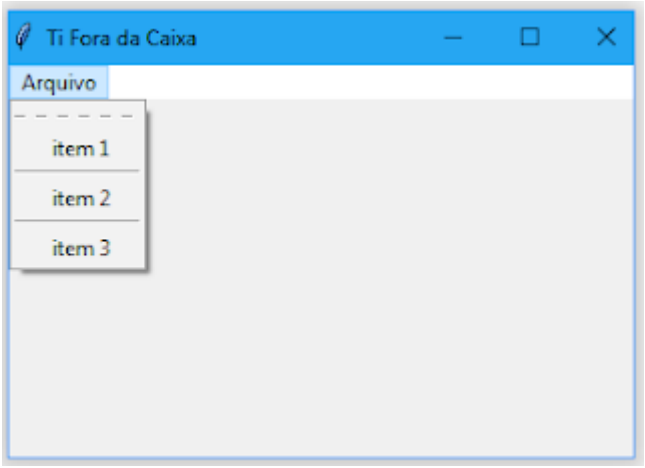
E podemos ainda adicionar outros itens e separadores:

```
from tkinter import *

from tkinter import filedialog

from tkinter import Menu



window = Tk()

window.title("Ti Fora da Caixa")
```

```
window.geometry('350x200')



menu = Menu(window)

window.config(menu=menu)

new_item = Menu(menu)

new_item.add_command(label='item 1')

new_item.add_separator()

new_item.add_command(label='item 2')

new_item.add_separator()

new_item.add_command(label='item 3')
```

```
menu.add_cascade(label='Arquivo', menu=new_item)




window.mainloop()
```

O resultado sera esse:




```
##############################################################
###########
```

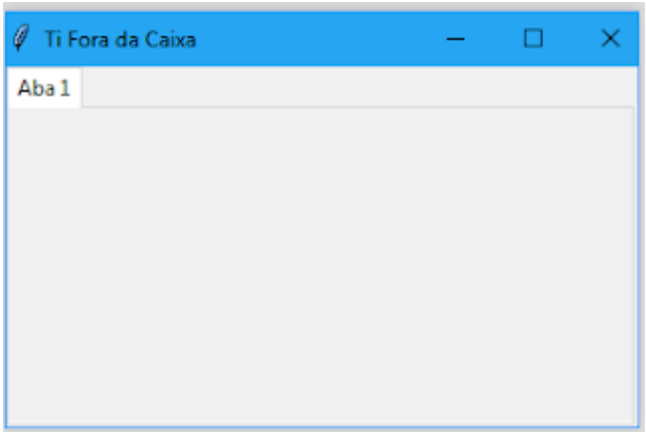
21-Adicionar separador por Abas:

Para criar um controle por abas, são necessarios realizar 3 tarefas

- 1 - Criar um controle de abas usando a classe notebook
- 2 - criar as abas usando a classe Frame
- 3 -Adicionar as abas ao controle de abas.
- E por fim adicionar as abas no controle de abas para torna-las visíveis
- Exemplo:

```
from tkinter import *
from tkinter import ttk

window = Tk()
window.title("Ti Fora da Caixa")
window.geometry('350x200')

tab_control = ttk.Notebook(window)#1
aba1 = ttk.Frame(tab_control)#2
tab_control.add(aba1, text='Aba 1')#3
tab_control.pack(expand=1, fill='both')


window.mainloop()
```

E o resultado sera esse:



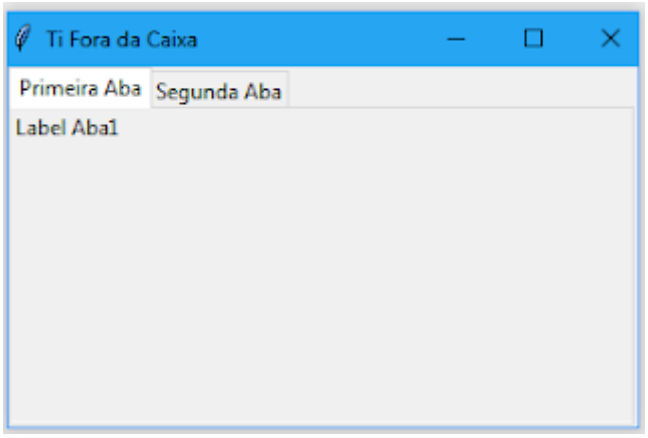

```
##################################################
##########
```

21.1 Adicionar widgets as abas:

Neste Exemplo adicionei duas labels em duas abas diferentes:

```
from tkinter import * from tkinter import ttk window = Tk() window.title("Ti Fora da Caixa") window.geometry('350x200')
tab_control = ttk.Notebook(window) aba1 = ttk.Frame(tab_control) aba2 = ttk.Frame(tab_control) tab_control.add(aba1, text='Primeira Aba') tab_control.add(aba2, text='Segunda Aba') lbl1 = Label(aba1, text= 'Label Aba1') lbl1.grid(column=0, row=0) lbl2 = Label(aba2, text= 'Label Aba2') lbl2.grid(column=0, row=0)
tab_control.pack(expand=1, fill='both') window.mainloop()
```

O resultado ao executar será esse:



##
##########

22 - Espaçamento padding:

Você pode adicionar um padding de forma bem simples:

```
lbl1 = Label(tab1, text= 'label1', padx=5, pady=5)
```

Basta modificar o padx e o pady do widget!

##
##########

23 - Prevenir o redimensionamento da janela:

Basta adicionar este parâmetro a janela:

```
window.resizable(width=False, height=False)
```

podemos ainda definir o tamanho minimo e maximo da janela:

```
window.minsize(width=800, height=600)
window.maxsize(width=300, height=200)
```

```
##################################################
###########
```

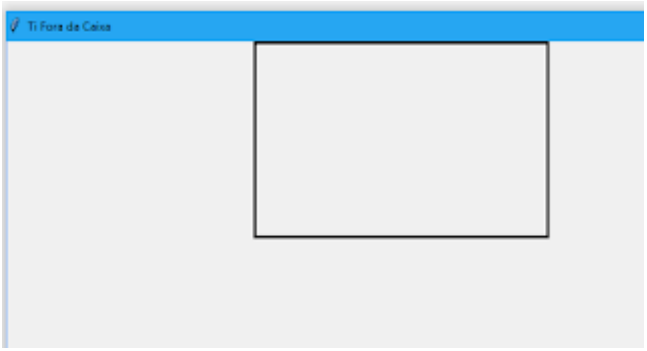
24 - Adicionar borda a um frame:

Ex:

```
from tkinter import *



window = Tk()

window.title("Ti Fora da Caixa")

window.geometry('800x600')



quadro = Frame(window, width=300, height=200, highli
ghtbackground="black", highlightthickness=3)
```

```
quadro.pack()


window.mainloop()
```

Resultado:



##
##########

24 - Adicionar borda a um frame:

Ex:

```
from tkinter import *


window = Tk()

window.title("Ti Fora da Caixa")

window.geometry('800x600')
```

```
quadro = Frame(window, width=300, height=200, highli
ghtbackground="black", highlightthickness=3)


quadro.pack()



window.mainloop()
```

##
##########

25 - Estilos de Bordas

Os estilos de bordas podem ser :
`"flat"`, `"raised"`, `"sunken"`, `"ridge"`, `"solid"`, and `"groove"`

Para usalos basta modificar o parametro relief ex:

```
l1 = Label(root, text="This", borderwidth=2, relief="groove")
```

Algumas labels com estilos de bordas: