

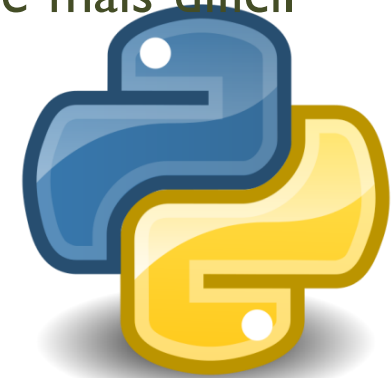


TÉCNICAS DE PROGRAMAÇÃO II

PROFº LUIZ CLÁUDIO

INTERFACE GRÁFICA DE USUÁRIO (GUI)

- **Interface gráfica de usuário (GUI)** é uma forma de interface de usuário que permite aos usuários interagir com computadores através de indicadores visuais usando itens como ícones, menus, janelas, etc.
- Tem vantagens sobre a Interface de linha de comando (CLI), onde os usuários interagem com computadores escrevendo comandos usando apenas o teclado e cujo uso é mais difícil do que a GUI.



INTERFACE GRÁFICA DE USUÁRIO (GUI)

- **Widgets**

- A interface de usuário do Tkinter é composta de widgets individuais.

Cada widget é representado como um objeto Python, instanciado de classes como `ttk.Frame`, `ttk.Label`, e `ttk.Button`.

- hierarquia dos widgets
- Os widgets são organizados em uma hierarquia.
- Quando criamos um widget filho, seu widget pai é passado como primeiro argumento
- para o construtor do widget.



O QUE É TKINTER?

- **Tkinter** é o módulo python embutido que é usado para criar aplicativos GUI. É um dos módulos mais comumente usados para criar aplicações GUI em Python, pois é simples e fácil de trabalhar.
- Você não precisa se preocupar com a instalação do módulo Tkinter separadamente, pois ele já vem com o Python.
- Ele fornece uma interface orientada a objetos para o kit de ferramentas Tk GUI



O QUE SÃO WIDGETS?

- **Widgets** no Tkinter são os elementos do aplicativo GUI que fornece vários controles (como Labels, Buttons, ComboBoxes, CheckBoxes, MenuBars, RadioButtons e muitos mais) para os usuários interagirem com o aplicativo.



DESENVOLVIMENTO DE INTERFACE GRÁFICA – 1º EXEMPLO

```
from tkinter import *
```

← Importação da Biblioteca

```
tela = Tk()
```

← Nome da variável para criação da tela

```
tela.mainloop()
```



← Resultado da
1º Tela

Obs. Neste primeiro
Exemplo não está
Definido o tamanho da tela



INTERFACE GRÁFICA – 2º EXEMPLO – DEFINE COR TAMANHO

```
from tkinter import *
#criação da variavel
tela = Tk()
```

```
#Titulo na barra de tarefas
tela.title("Fatec Registro")
```

```
#Configuração da cor da tela (opcional)
tela.configure(background= '#1e3743')
```

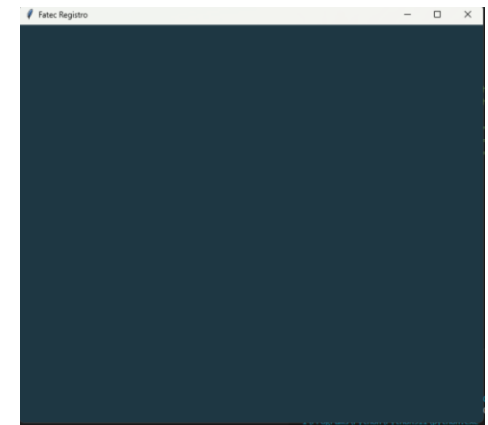
```
#Configuração do tamanho da tela
tela.geometry("700x600")
```

```
tela.mainloop()
```

Define o Título

**Define a cor de
fundo**

Define tamanho da tela



**Resultado da
Tela com cor de fundo e
tamanho**



INTERFACE GRÁFICA – 3º EXEMPLO- DEFINE REAJUSTE DA TELA

```
from tkinter import *
#criação da variavel
tela = Tk()

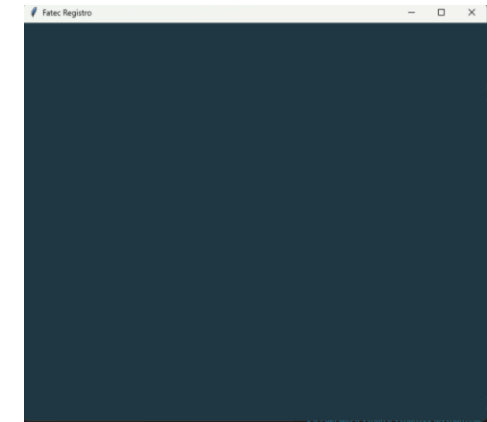
#Titulo na barra de tarefas
tela.title("Fatec Registro")
#Configuração da cor da tela (opcional)
tela.configure(background= '#1e3743')
#Configuração do tamanho da tela
tela.geometry("700x600")

tela.resizable(True, True)
#Tamanho maximo que a tela pode chegar
tela.maxsize(width=800, height=700)
#Tamanho minimo que a tela pode chegar
tela.minsize(width=500, height=300)
tela.mainloop()
```

Habilita reajuste da tela

para altura e largura

Define o tamanho e máximo e mínimo que pode ser reajustado a tela



Tela pode ser reajustado até o tamanho máximo ou mínimo

INTERFACE GRÁFICA – 4º EXEMPLO- ADICIONANDO LABEL

```
from tkinter import *
#criação da variavel
tela = Tk()

#Titulo na barra de tarefas
tela.title("Fatec Registro")
#Configuração da cor da tela (opcional)
tela.configure(background= '#1e3743')
#Configuração do tamanho da tela
tela.geometry("700x600")

tela.resizable(True, True)
#Tamanho maximo que a tela pode chegar
tela.maxsize(width=800, height=700)
#Tamanho minimo que a tela pode chegar
tela.minsize(width=500, height=300)

lbl_teste = Label(tela, text="TESTE").place (x=10, y=10)
#lbl_teste e o nome de indentificação interno da label
#Label e o tipo do componente
#tela a variavel que a label está ligado
#text="" e o texto a ser exibido na tela
#place o posicionamento da tela
tela.mainloop()
```

Resultado na tela , com
o label



Label cria o label , em text
define o texto do label e
place define a posição do
widget na tela



INTERFACE GRÁFICA – 5º EXEMPLO- PARA ADICIONAR MAIS DE UM LABEL

```
from tkinter import *

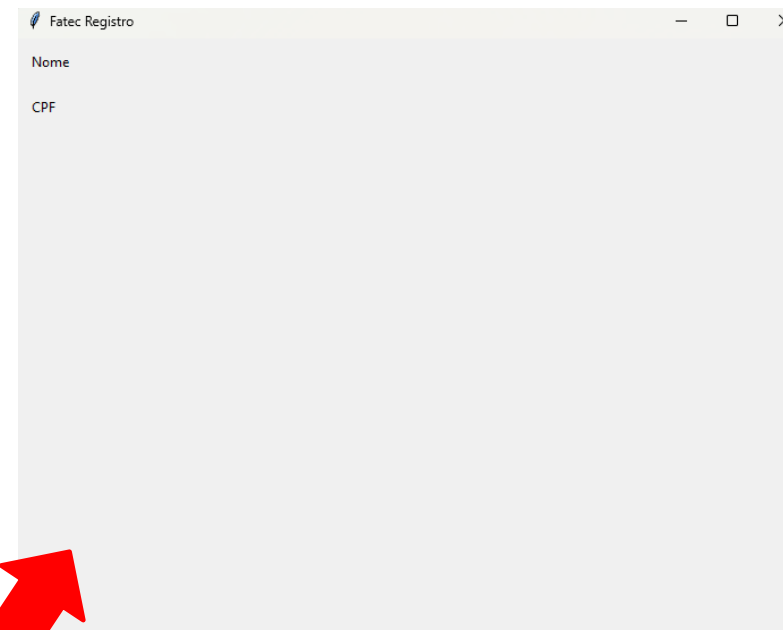
tela = Tk()

tela.title("Fatec Registro")
tela.geometry("700x600")
tela.resizable(True, True)
tela.maxsize(width=800, height=700)
tela.minsize(width=500, height=300)

lbl_nome = Label(tela, text="Nome").place (x=10, y=10)
lbl_cpf = Label(tela, text="CPF").place(x=10, y=50 )
tela.mainloop()
```

Para os labels não ficarem sobrepostos , devemos mudar as posições de y

Labels posicionados na tela



INTERFACE GRÁFICA – 6º EXEMPLO- CARREGAR INTERFACE NO CENTRO DA TELA

```
from tkinter import *

tela = Tk()

tela.title("Fatec Registro")

#dimensoes da janela
largura = 800
altura = 300

#resolução do sistema(tela)
largura_screen = tela.winfo_screenwidth()
altura_screen = tela.winfo_screenheight()

posx = largura_screen/2 - largura/2
posy = altura_screen/2 - altura/2

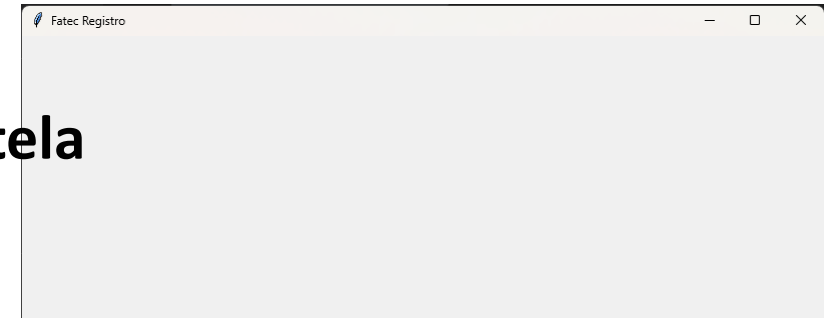
#visualização do tamanho da tela no terminal
print(largura_screen, altura_screen)

#definição do geometry
tela.geometry("%dx%d+%d+%d" % (largura, altura, posx, posy))
tela.mainloop()
```

Define o tamanho da tela

Pega as definições da tela do windows

Define o posicionamento centralizado



%d , substitui um número

% concatena as variáveis referentes a cada %d

A tela será carregada com tamanho definido no centro da tela



INTERFACE GRÁFICA – 7º EXEMPLO- FORMATANDO LABELS

```
from tkinter import *

tela = Tk()

tela.title("Fatec Registro")
tela.geometry("700x600")
tela.resizable(True, True)
tela.maxsize(width=800, height=700)
tela.minsize(width=500, height=300)

lbl_nome = Label(tela, text="Nome",
                 font="Arial 20 bold italic",
                 fg="#FF8C00").place(x=10, y=10)

lbl_cpf = Label(tela, text="CPF",
               font="Times 15 italic",
               fg="#7CFC00").place(x=10, y=50)

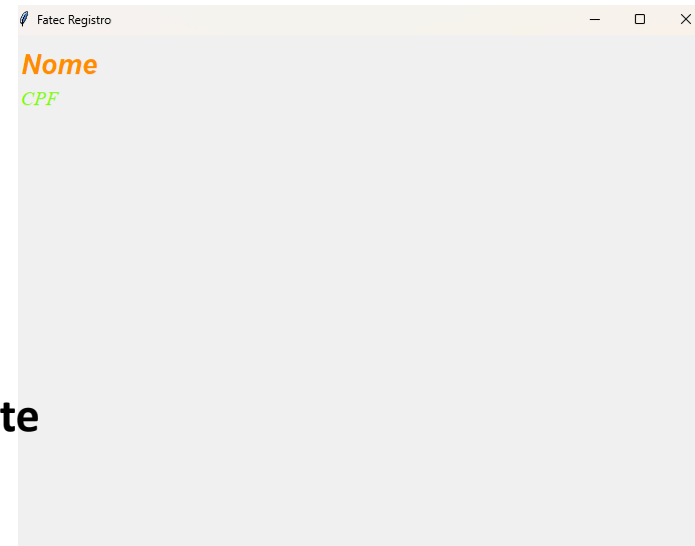
tela.mainloop()
```

font define o tipo de fonte

fg = cor da letra

bg= cor de fundo

Resultado do label com a cor da fonte e o tipo da fonte



INTERFACE GRÁFICA – 8º EXEMPLO- CRIANDO BOTÃO

```
from tkinter import *
```

```
tela = Tk()
```

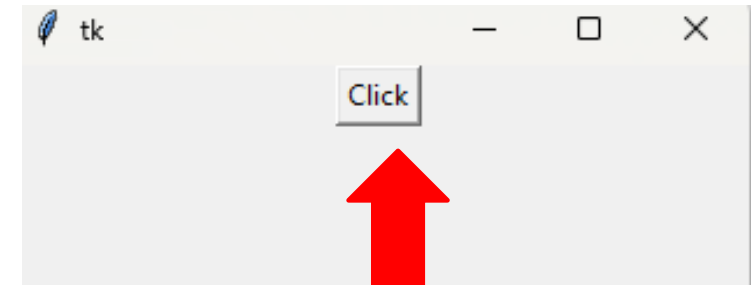
```
btn_botao = Button(tela, text="Click")
```

```
btn_botao.pack()
```

```
tela.mainloop()
```

Button cria o botão, e o text é o texto que vai aparecer no botão

Pack ,faz com que o widget esteja juntamente com o widget Pai que nesta caso é a tela



Botão sendo mostrado junto da tela



INTERFACE GRÁFICA – 8º EXEMPLO- ADICIONANDO CAIXA DE TEXTO

```
from tkinter import *
```

```
tela = Tk()
```

```
txt_nome = Entry(tela, width=20, borderwidth=5, fg="blue", bg="white")
txt_nome.pack()
txt_nome.insert(0, "Digite o seu nome")
```

```
btn_botao = Button(tela, text="Click")
btn_botao.pack()
```

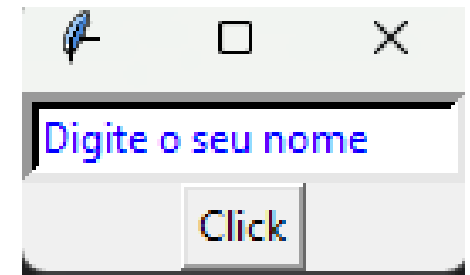
```
tela.mainloop()
```

Entry define a caixa de texto

Width = largura da caixa

Borderwidth= tamanho da borda

Define a entrada de dados na caixa de texto



Caixa de Texto na tela e com botão



INTERFACE GRÁFICA – 8º EXEMPLO- ADICIONANDO CAIXA DE TEXTO

```
from tkinter import *
```

```
tela = Tk()
```

```
txt_nome = Entry(tela, width=20, borderwidth=5, fg="blue", bg="white")
txt_nome.pack()
txt_nome.insert(0, "Digite o seu nome")
```

```
btn_botao = Button(tela, text="Click")
btn_botao.pack()
```

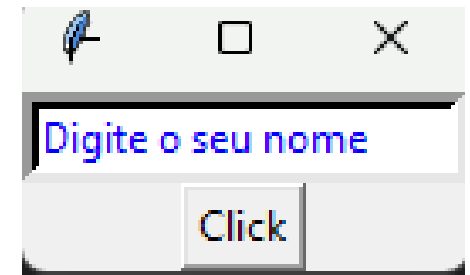
```
tela.mainloop()
```

Entry define a caixa de texto

Width = largura da caixa

Borderwidth= tamanho da borda

Define a entrada de dados na caixa de texto



Caixa de Texto na tela e com botão



CRIANDO FUNÇÃO

- **O que é função?**
- A ideia básica de uma função, implementada em alguma linguagem de programação, é encapsular um código que poderá ser invocado/chamado por qualquer outro trecho do programa.
- Seu significado e uso são muito parecidos com o de funções matemáticas, ou seja, existe um nome, uma definição e posterior invocação à função.



CRIANDO FUNÇÃO EM PYTHON

Def cria uma função
meu_click = nome da função

.get pega o valor digitado na caixa de texto

```
def meu_click():
    lbl_hello = Label(tela, text="Bem Vindo " + txt_nome.get())
    lbl_hello.pack()
```

Nesta função está sendo criado um label que irá mostrar a informação digitada Na caixa de texto



CHAMANDO A FUNÇÃO NO BOTÃO

```
btn_botao = Button(tela, text="Click", command=meu_click)
btn_botao.pack()

tela.mainloop()
```

command = evento para chamar a função

função em python, deve ter o mesmo nome do command do botão

```
def meu_click():
    lbl_hello = Label(tela, text="Bem Vindo " + txt_nome.get())
    lbl_hello.pack()
```



RESULTADO AO CLICAR NO BOTÃO

```
from tkinter import *

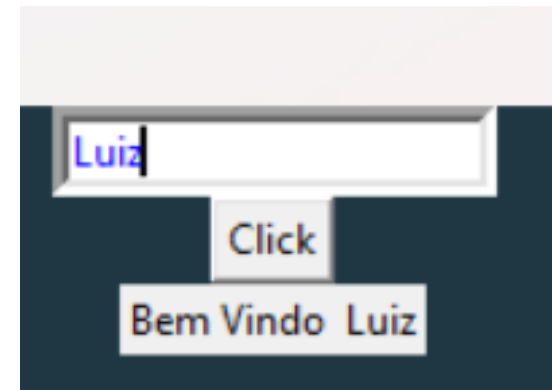
tela = Tk()

txt_nome = Entry(tela, width=20, borderwidth=5, fg="blue", bg="white")
txt_nome.pack()
txt_nome.insert(0, "Digite o seu nome")

def meu_click():
    lbl_hello = Label(tela, text="Bem Vindo " + txt_nome.get())
    lbl_hello.pack()

btn_botao = Button(tela, text="Click", command=meu_click)
btn_botao.pack()

tela.mainloop()
```



**Ao clicar no botão será
Mostrado no label o
que foi digitado na caixa de texto**





OBRIGADO

LUIZ.BARRETO2@FATEC.SP.GOV.BR