Computação II

Departamento de Ciência da Computação – UFRJ

Interface Gráfica – Parte 1

Professora: *Fernanda Duarte Vilela Reis de Oliveira*

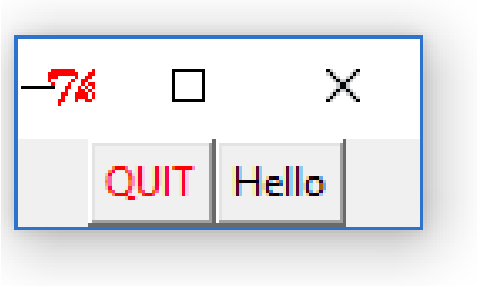
2019/01

1. Interfaces gráficas - Tkinter

- O que é o Tkinter?
  - Biblioteca open source;
  - Interface gráfica - portable graphical user interface (GUI);
  - Portabilidade;
  - Tkinter incluido nas distribuições de Python.

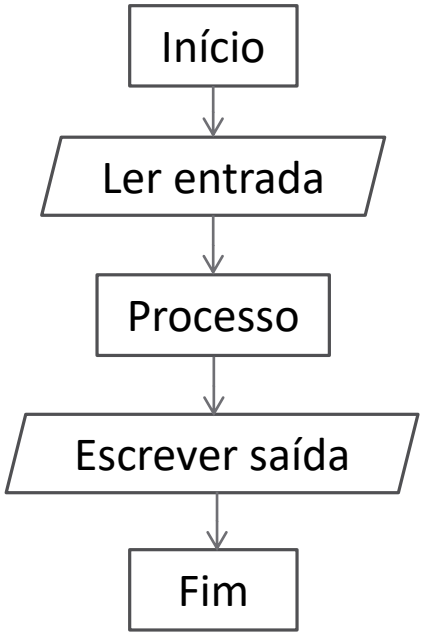
1. Interfaces gráficas - Tkinter

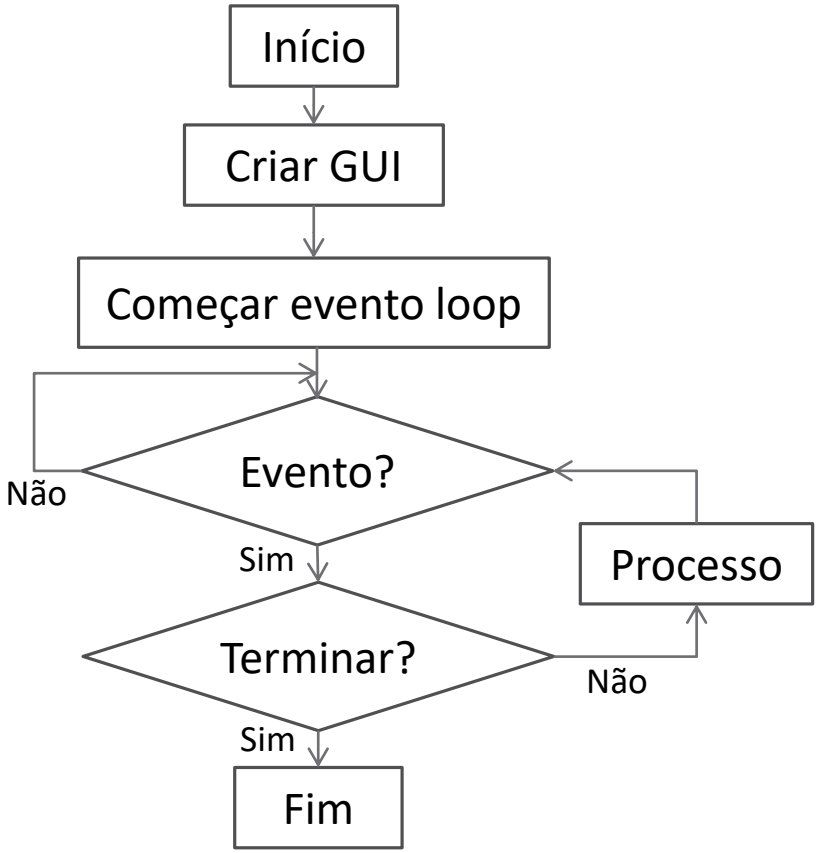
- Graphical User Interface (GUI) - conjunto de elementos gráficos (widgets) com:
  - Atributos (posição, tamanho, cor, forma...)
  - Métodos (reação a eventos gerados pelo usuário, pelo próprio programa pelo sistema operacional...)



1. Interfaces gráficas - Tkinter

- Console X GUI:





1. Interfaces gráficas - Tkinter

- Criando uma janela:

Python 2.7:

```
from Tkinter import *

root = Tk()

root.mainloop()
```

Python 3.7:

```
from tkinter import *

root = Tk()

root.mainloop()
```

1. Interfaces gráficas - Tkinter

- Criando uma janela:

Importa toda a biblioteca Tkinter

```
from tkinter import *

root = Tk()

root.mainloop()
```

1. Interfaces gráficas - Tkinter

- Criando uma janela:

```
from tkinter import *

root = Tk()

root.mainloop()
```

Cria um objeto da biblioteca Tkinter.
O objeto criado será a nossa janela principal, chamada Tk root widget.

1. Interfaces gráficas - Tkinter

- Criando uma janela:

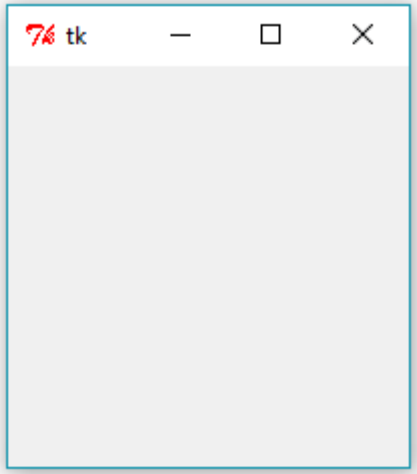
```
from tkinter import *

root = Tk()

root.mainloop()
```

Coloca a janela em loop até que haja um evento que encerre a janela.

1. Interfaces gráficas - Tkinter

- Criando uma janela:

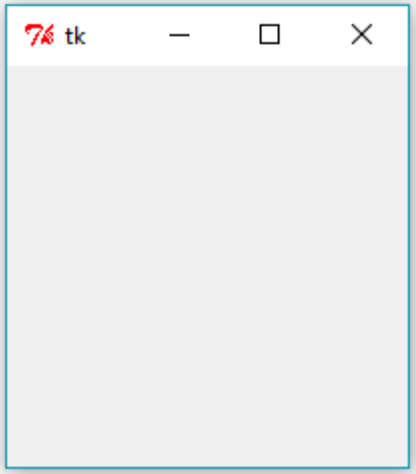
```
from tkinter import *

root = Tk()

root.mainloop()
```



1. Interfaces gráficas - Tkinter

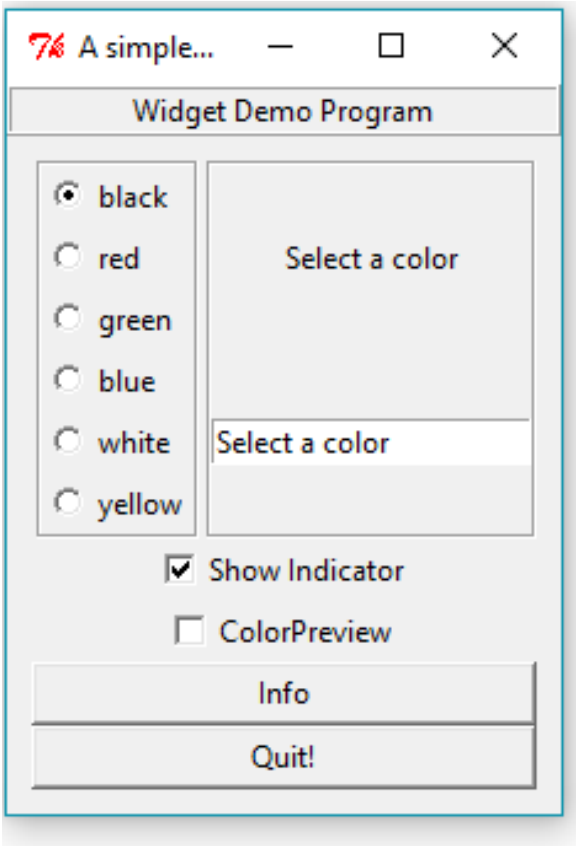
- Criando uma janela:

```
from tkinter import *

root = Tk()

root.mainloop()
```



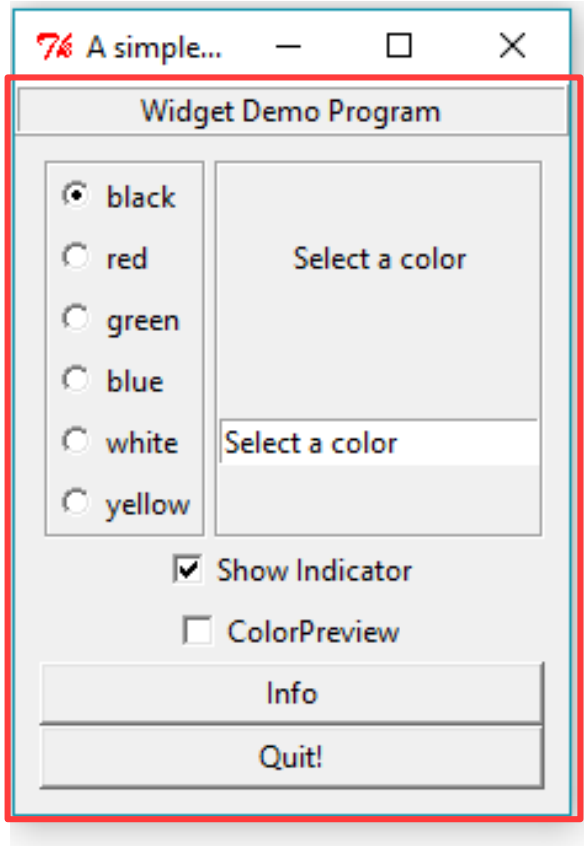
1. Interfaces gráficas - Tkinter

- Widgets - elementos da interface gráfica.
  - Frame;
  - Labels;
  - Button;
  - Radiobuttons;
  - Checkbuttom.



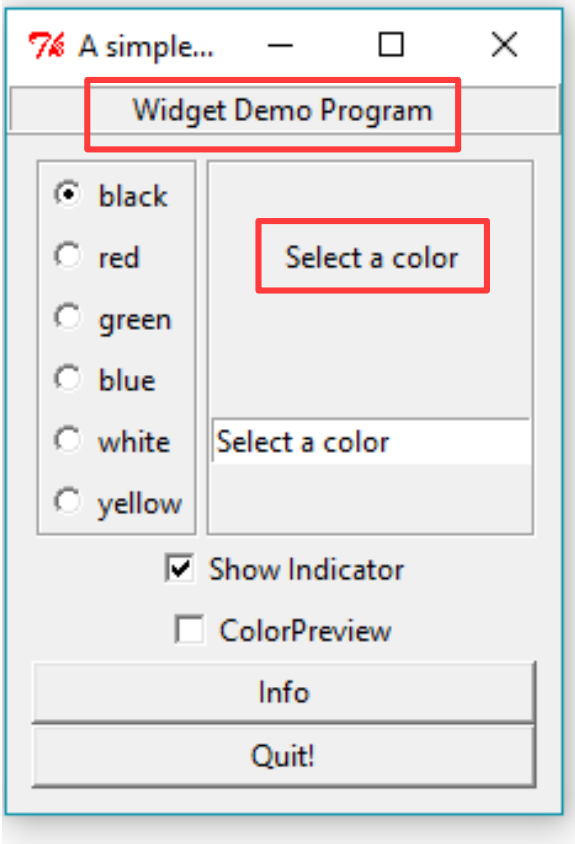
1. Interfaces gráficas - Tkinter

- Widgets - elementos da interface gráfica.
  - Frame;
  - Labels;
  - Button;
  - Radiobuttons;
  - Checkbuttom.



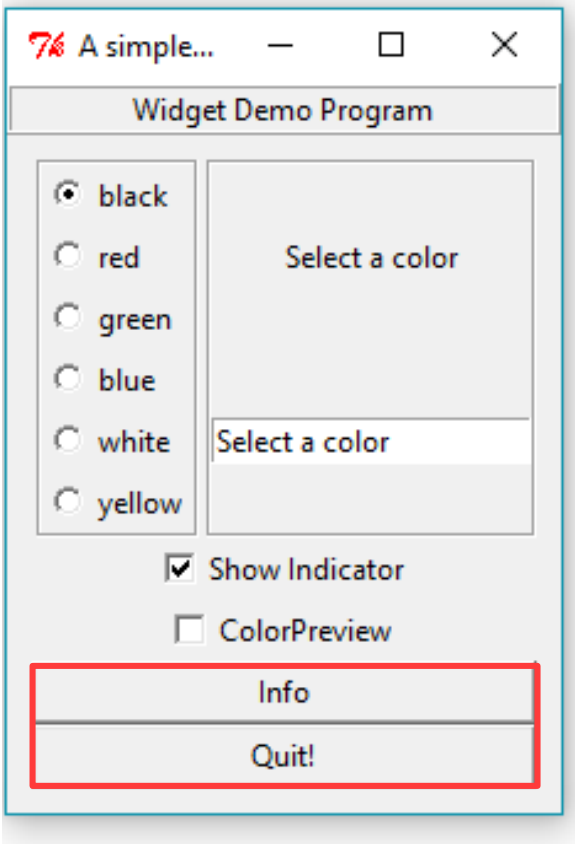
1. Interfaces gráficas - Tkinter

- Widgets - elementos da interface gráfica.
  - Frame;
  - Labels;
  - Button;
  - Radiobuttons;
  - Checkbuttom.



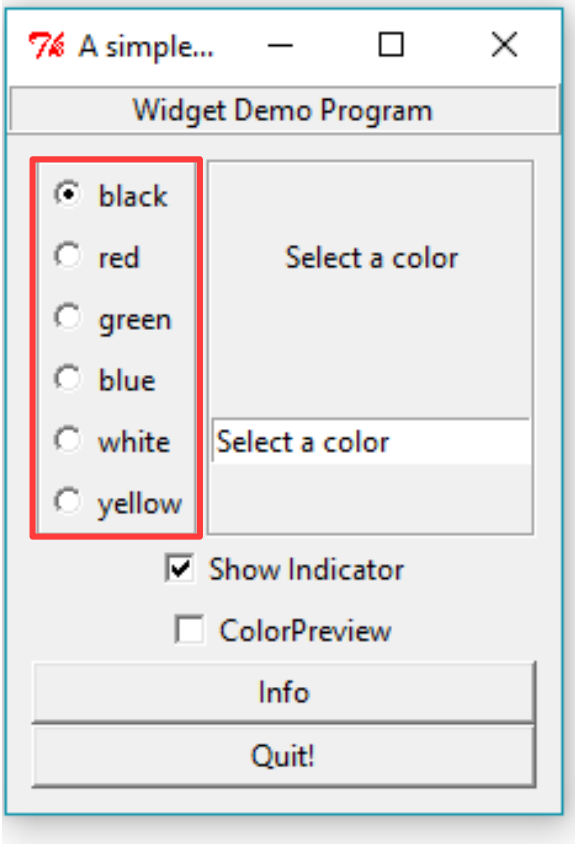
1. Interfaces gráficas - Tkinter

- Widgets - elementos da interface gráfica.
  - Frame;
  - Labels;
  - Button;
  - Radiobuttons;
  - Checkbuttom.



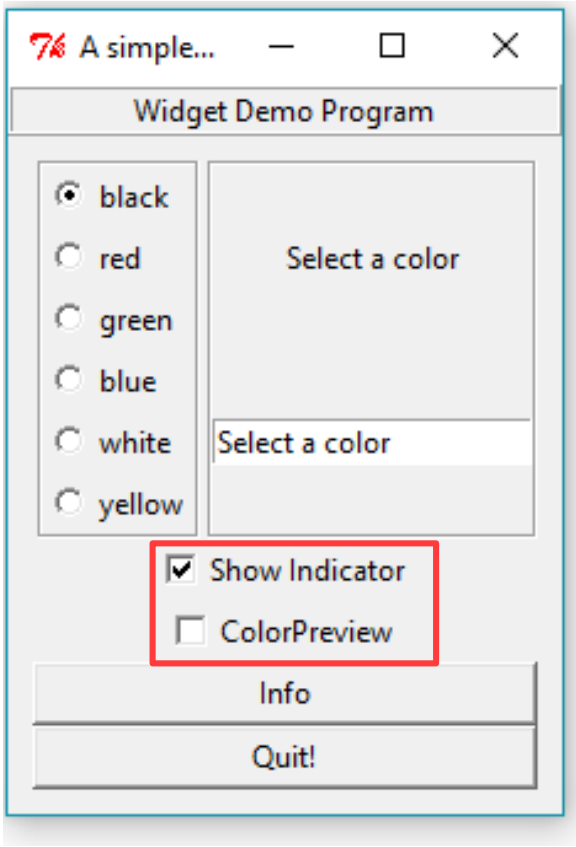
1. Interfaces gráficas - Tkinter

- Widgets - elementos da interface gráfica.
  - Frame;
  - Labels;
  - Button;
  - Radiobuttons;
  - Checkbuttom.



1. Interfaces gráficas - Tkinter

- Widgets - elementos da interface gráfica.
  - Frame;
  - Labels;
  - Button;
  - Radiobuttons;
  - Checkbuttom.



1. Interfaces gráficas - Tkinter

- Incluindo uma label na janela:

```
from tkinter import *

root = Tk()
lbl = Label(root,text='Hello world!')
lbl.pack()
root.mainloop()
```

1. Interfaces gráficas - Tkinter

- Incluindo uma label na janela:

```
from tkinter import *

root = Tk()
lbl = Label(root,text='Hello world!')
lbl.pack()
root.mainloop()
```

Cria a label

Método pack(): gerenciador de geometria, controla o layout da janela

1. Interfaces gráficas - Tkinter

- Incluindo uma label na janela:

```
from tkinter import *

root = Tk()
lbl = Label(root,text='Hello world!')
lbl.pack()
root.mainloop()
```

1. Interfaces gráficas - Tkinter

- Opções da label – cor e fonte:

```
from tkinter import *

root = Tk()

lbl = Label(root,text='Hello world!',\
            fg="red",\
            bg='blue', \
            font=("Arial", 16))
lbl.pack()

root.mainloop()
```

Texto que aparecerá na label

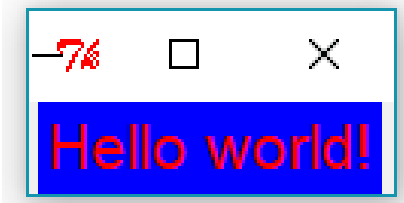
Cor do plano de frente (foreground), no caso, cor da letra

Cor do plano de fundo (background)

Tipo de fonte e tamanho

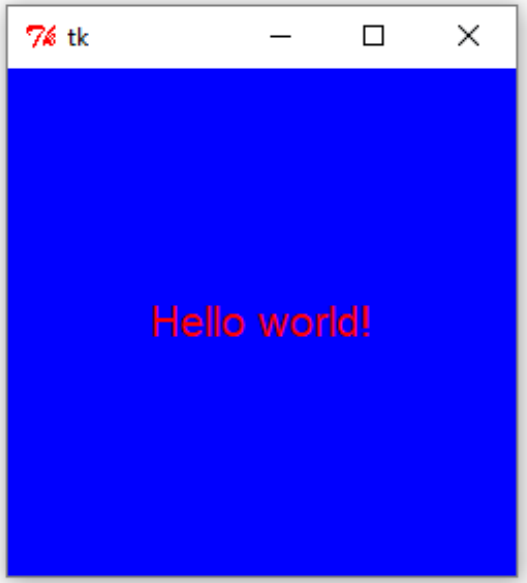
Como o tamanho da label não foi explicitamente definido, o tamanho depende do conteúdo da label.

Da mesma forma, o tamanho da janela do tkinter é definida de acordo com o tamanho da label.



1. Interfaces gráficas - Tkinter

- Opções da label – tamanho da label:




```
from tkinter import *

root = Tk()

lbl = Label(root,text='Hello world!',\
            fg='red',\
            bg='blue', \
            font=('Arial', 20),\
            height=10,\
            width=20)
lbl.pack()

root.mainloop()
```

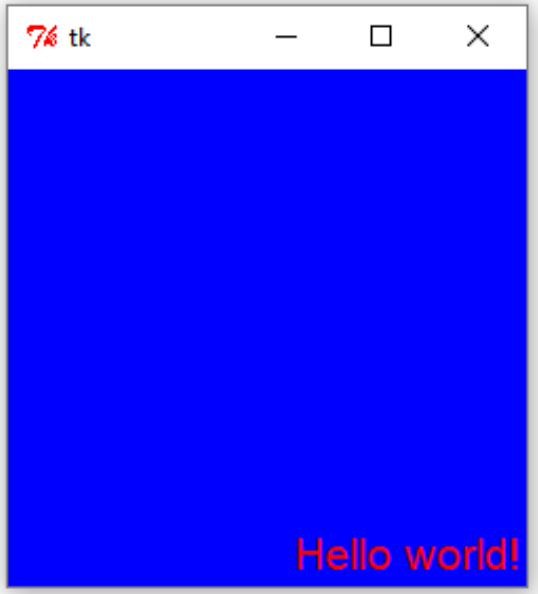
Definindo o tamanho da label.

Se o tamanho do texto for menor que o tamanho da label, o texto é cortado.

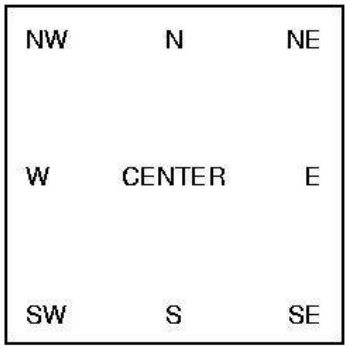
1. Interfaces gráficas - Tkinter

- Opções da label – posição do texto dentro da label:

```
from tkinter import *

root = Tk()

lbl = Label(root,text='Hello world!',\
            fg='red',\
            bg='blue', \
            font=('Arial', 16),\
            height=10,\
            width=20,\
            anchor='se')
lbl.pack()

root.mainloop()
```



Posição do texto dentro da label, só funciona se o tamanho da label for maior que o texto.



1. Interfaces gráficas - Tkinter

- Opções da label – adicionando mais de uma label:

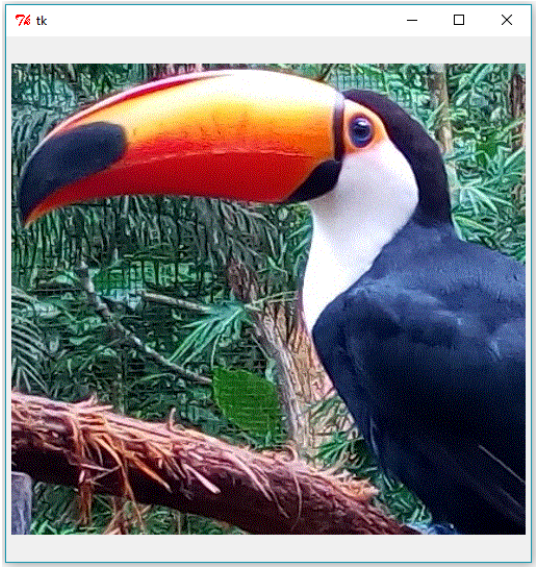
```
from tkinter import *

root = Tk()

lbl = Label(root,text='Hello world!',\
            fg='red',\
            bg='blue', \
            font=('Arial', 16),)
lbl.pack()
lbl = Label(root,text='Ola!',\
            fg='green',\
            bg='black', \
            font=('Times New Roman', 14))
lbl.pack()

root.mainloop()
```

Por padrão, cada nova label é adicionada abaixo da anterior, no centro da janela. Quem gerencia a posição das labels é o método pack.

1. Interfaces gráficas - Tkinter

- Opções da label – adicionando uma figura:

```
from tkinter import *

root = Tk()

photo=PhotoImage(file='tucano2.gif')
lbl = Label(root,image=photo,\
            height=500,\
            width=500)
lbl.pack()

root.mainloop()
```



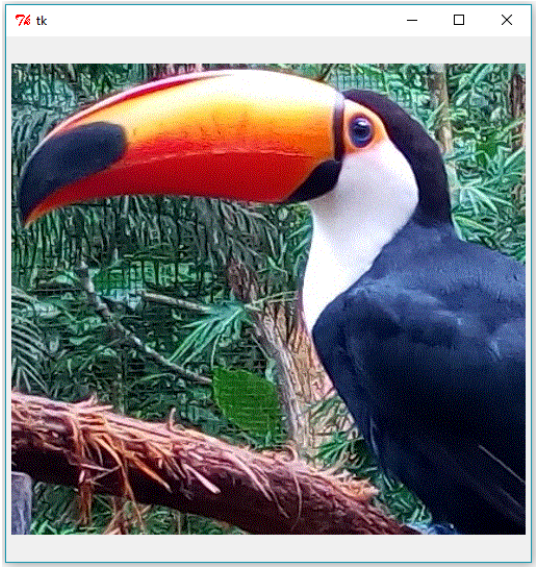
1. Interfaces gráficas - Tkinter

- Opções da label – adicionando uma figura:

```
from tkinter import *

root = Tk()
```

Função do tkinter, aceita gif, pgm e ppm

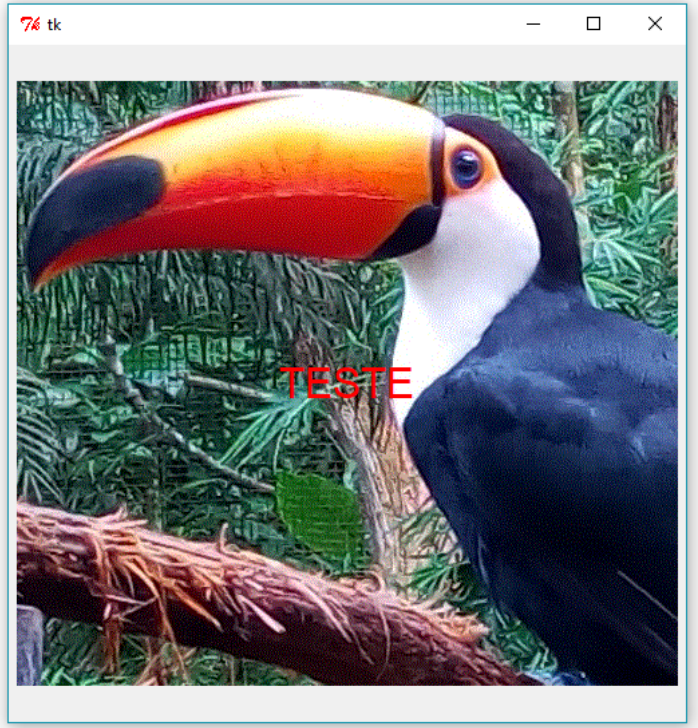
```
photo=PhotoImage(file='tucano2.gif')
lbl = Label(root,image=photo,\
            height=500,\
            width=500)
lbl.pack()

root.mainloop()
```



1. Interfaces gráficas - Tkinter

- Opções da label – adicionando uma figura:

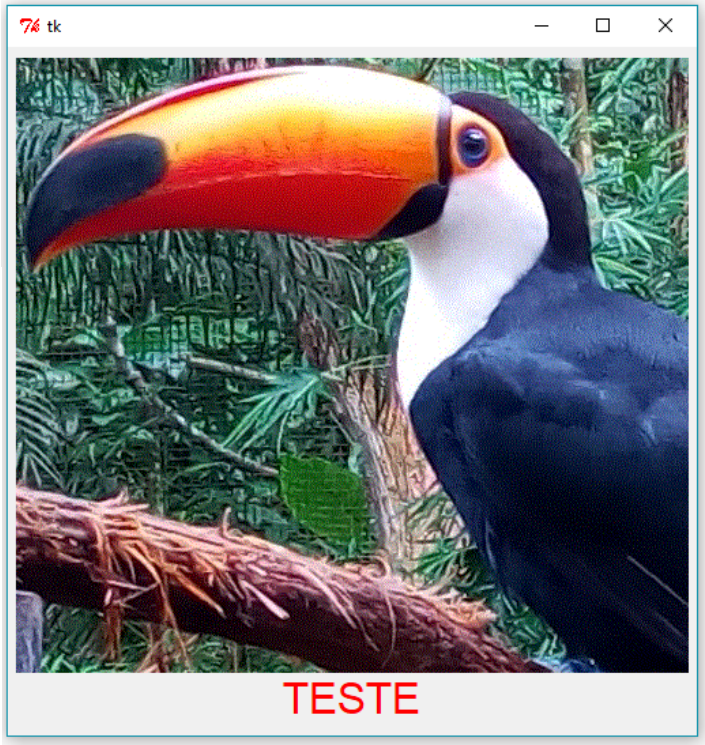
```
from tkinter import *

root = Tk()

photo=PhotoImage(file='tucano2.gif')
lbl = Label(root,image=photo,
            text = 'TESTE',
            fg = 'red',
            font = ('Arial',24),
            height = 500,
            width = 500,
            compound = 'c')
lbl.pack()

root.mainloop()
```



1. Interfaces gráficas - Tkinter

- Opções da label – adicionando uma figura:

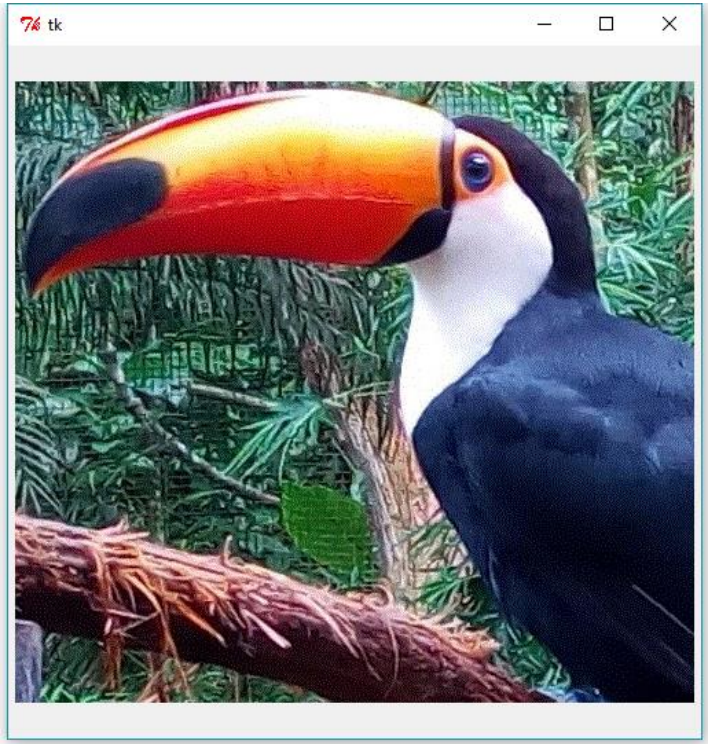
```
from tkinter import *

root = Tk()

photo=PhotoImage(file='tucano2.gif')
lbl = Label(root,image=photo,
            text = 'TESTE',
            fg = 'red',
            font = ('Arial',24),\
            height=500,\
            width=500,
            compound = 't')
lbl.pack()

root.mainloop()
```



1. Interfaces gráficas - Tkinter

- Opções da label – outros tipos de imagem - PIL:

```
from tkinter import *
from PIL import Image, ImageTk

root = Tk()

image = Image.open("tucanojpg.jpg")
photo = ImageTk.PhotoImage(image)
lbl = Label(root,image=photo,\
            height=500,\
            width=500)
lbl.pack()

root.mainloop()
```



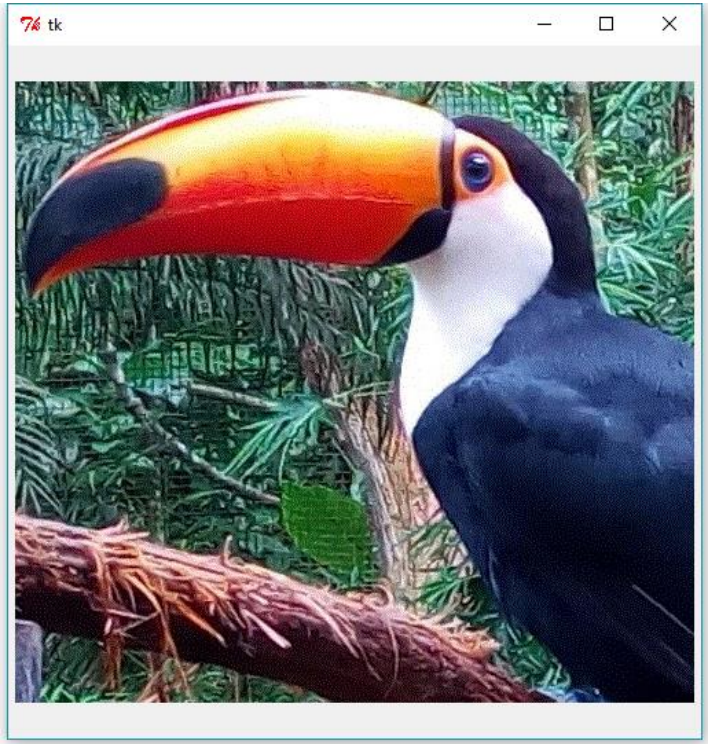
1. Interfaces gráficas - Tkinter

- Opções da label – outros tipos de imagem - PIL:

```
from tkinter import *
from PIL import Image, ImageTk

root = Tk()

image = Image.open("tucanojpg.jpg")
photo = ImageTk.PhotoImage(image)
lbl = Label(root,image=photo,\
            height=500,\
            width=500)
lbl.pack()

root.mainloop()
```

Esse módulo deve ser instalado:

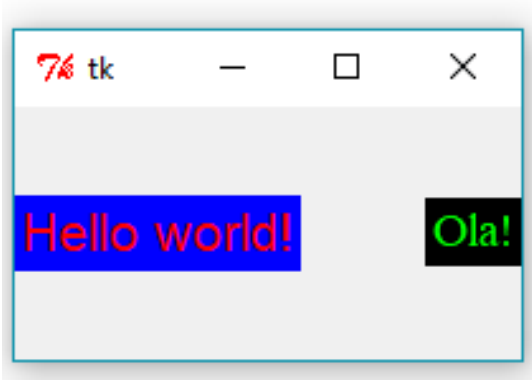
pip install pillow



1. Interfaces gráficas - Tkinter

- Modificando o tamanho da janela e a posição dos labels:

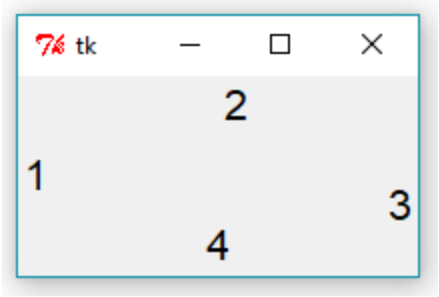
```
from tkinter import *

root = Tk()
root.geometry('200x100')

lbl = Label(root,text='Hello world!',\
            fg='red',\
            bg='blue', \
            font=('Arial', 16),)
lbl.pack(side=LEFT)
lbl = Label(root,text='Ola!',\
            fg='green',\
            bg='black', \
            font=('Times New Roman', 14))
lbl.pack(side=RIGHT)

root.mainloop()
```



1. Interfaces gráficas - Tkinter

- Modificando o tamanho da janela e a posição dos labels:

```
from tkinter import *

root = Tk()
root.geometry('200x100')

lbl = Label(root,text='1',\
            font=('Arial', 16))
lbl.pack(side=LEFT)
lbl = Label(root,text='2',\
            font=('Arial', 16))
lbl.pack(side=TOP)
lbl = Label(root,text='3',\
            font=('Arial', 16))
lbl.pack(side=RIGHT)
lbl = Label(root,text='4',\
            font=('Arial', 16))
lbl.pack(side=BOTTOM)

root.mainloop()
```



1. Interfaces gráficas - Tkinter

- Utilizando classes:

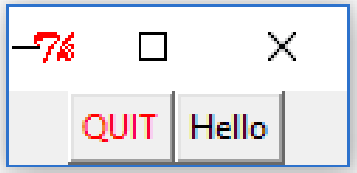
```
from tkinter import *

class window:
    def __init__(self, master):
        self.lbl = Label(master, text='Hello world!')
        self.lbl.pack()


root = Tk()

w = window(root)

root.mainloop()
```

1. Interfaces gráficas - Tkinter

- Criando frames e botões:

```
from tkinter import *

class window:
    def __init__(self, master):
        frame = Frame(master)
        frame.pack()

        self.button = Button(frame, text="QUIT", fg="red",\
                             command=master.destroy)
        self.button.pack(side=LEFT)
        self.hi_there = Button(frame, text="Hello",\
                               command=self.say_hi)
        self.hi_there.pack(side=LEFT)
    def say_hi(self):
        print ("Hi there, everyone!")

root = Tk()
app = window(root)
root.mainloop()
```

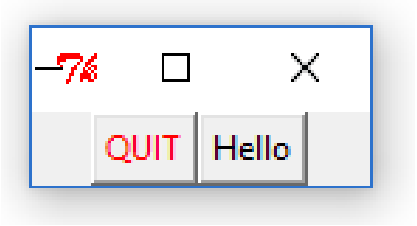
Frame: contêiner



1. Interfaces gráficas - Tkinter

- Criando frames e botões:

```
from tkinter import *

class window:
    def __init__(self, master):
        frame = Frame(master)
        frame.pack()

        self.button = Button(frame, text="QUIT", fg="red",\
                             command=master.destroy)
        self.button.pack(side=LEFT)
        self.hi_there = Button(frame, text="Hello",\
                               command=self.say_hi)
        self.hi_there.pack(side=LEFT)
    def say_hi(self):
        print ("Hi there, everyone!")

root = Tk()
app = window(root)
root.mainloop()
```

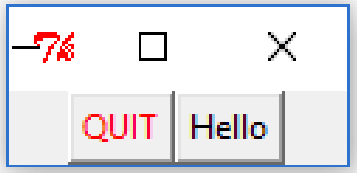


Inclusão dos botões no frame

1. Interfaces gráficas - Tkinter

- Criando frames e botões:

```
from tkinter import *

class window:
    def __init__(self, master):
        frame = Frame(master)
        frame.pack()

        self.button = Button(frame, text="QUIT", fg="red",\
                             command=master.destroy)
        self.button.pack(side=LEFT)
        self.hi_there = Button(frame, text="Hello",\
                               command=self.say_hi)
        self.hi_there.pack(side=LEFT)
    def say_hi(self):
        print ("Hi there, everyone!")

root = Tk()
app = window(root)
root.mainloop()
```

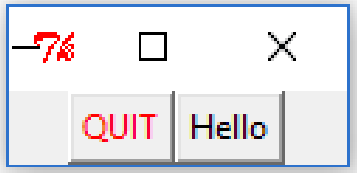


Ao clicar em um botão, um evento é gerado.
A resposta a esse evento é a chamada da função definida pelo argumento "command".

1. Interfaces gráficas - Tkinter

- Criando frames e botões:

```
from tkinter import *

class window:
    def __init__(self, master):
        frame = Frame(master)
        frame.pack()

        self.button = Button(frame, text="QUIT", fg="red",\
                             command=master.destroy)
        self.button.pack(side=LEFT)
        self.hi_there = Button(frame, text="Hello",\
                               command=self.say_hi)
        self.hi_there.pack(side=LEFT)
    def say_hi(self):
        print ("Hi there, everyone!")

root = Tk()
app = window(root)
root.mainloop()
```

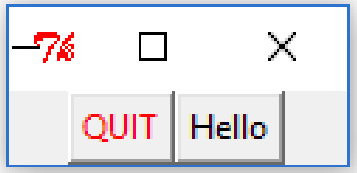


Ao clicar no botão onde está escrito "Hello", a função say_hi é chamada. Ao clicar no botão "QUIT", é chamada a função destroy, uma função do Tkinter que destrói uma janela. Utilizando essa função, o programa é fechado.

1. Interfaces gráficas - Tkinter

- Criando frames e botões:

```
from tkinter import *

class window:
    def __init__(self, master):
        frame = Frame(master)
        frame.pack()

        self.button = Button(frame, text="QUIT", fg="red",\
                             command=master.destroy)
        self.button.pack(side=LEFT)
        self.hi_there = Button(frame, text="Hello",\
                               command=self.say_hi)
        self.hi_there.pack(side=LEFT)
    def say_hi(self):
        print ("Hi there, everyone!")

root = Tk()
app = window(root)
root.mainloop()
```



Ao clicar no botão onde está escrito "Hello", a função say_hi é chamada. Ao clicar no botão "QUIT", é chamada a função destroy, uma função do Tkinter que destrói uma janela. Utilizando essa função, o programa é fechado.

1. Interfaces gráficas - Tkinter

- Criando uma caixa de texto:

```
from tkinter import *

def calcular():
    texto = caixaTexto.get()
    if '+' in texto:
        fatores = texto.split('+')
        print(float(fatores[0])+float(fatores[1]))
    elif '-' in texto:
        fatores = texto.split('-')
        print(float(fatores[0])-float(fatores[1]))
    else:
        print(texto)

root = Tk()

caixaTexto = Entry(root)
caixaTexto.pack()
b = Button(root, text="Calcular", width=10, command=calcular)
b.pack()

root.mainloop()
```

A widget “Entry” cria uma caixa de texto.

\Aula9_Tkinter\ex37CaixaTexto_Python3.py

1. Interfaces gráficas - Tkinter

- Criando uma caixa de texto:

```
from tkinter import *

def calcular():
    texto = caixaTexto.get()
    if '+' in texto:
        fatores = texto.split('+')
        print(float(fatores[0])+float(fatores[1]))
    elif '-' in texto:
        fatores = texto.split('-')
        print(float(fatores[0])-float(fatores[1]))
    else:
        print(texto)

root = Tk()

caixaTexto = Entry(root)
caixaTexto.pack()
b = Button(root, text="Calcular", width=10, command=calcular)
b.pack()

root.mainloop()
```

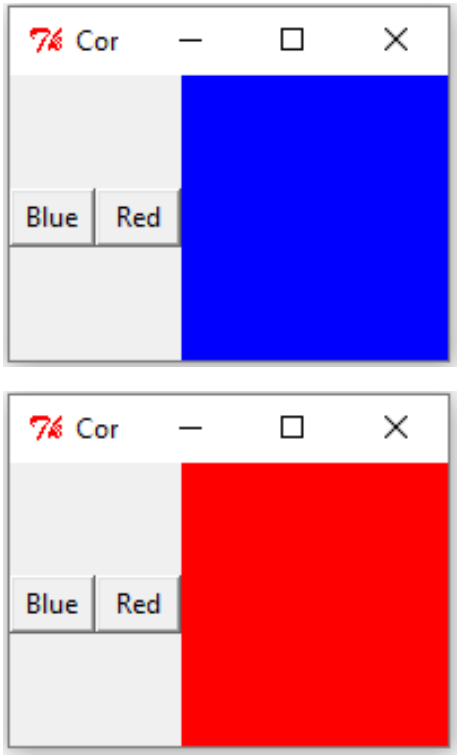
O texto digitado pode ser lido através do método get.

\Aula9_Tkinter\ex37CaixaTexto_Python3.py

1. Interfaces gráficas - Tkinter

- Modificando os parâmetros das widgets: parâmetros são guardados em um dicionário em que os nomes dos parâmetros são as chaves.

```
from tkinter import *
def blueLabel():
    lbl['bg']='Blue'
def redLabel():
    lbl['bg']='Red'
root = Tk()
root.title('Cor')
blueButton = Button(root, text="Blue",\
                    command=blueLabel,width=4)
blueButton.pack(side=LEFT)
redButton = Button(root, text="Red",\
                   command=redLabel,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```



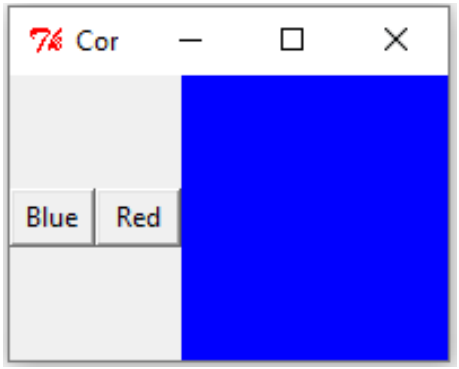
\Exemplos\ex01ButtonLabelWithPack_Python3.py

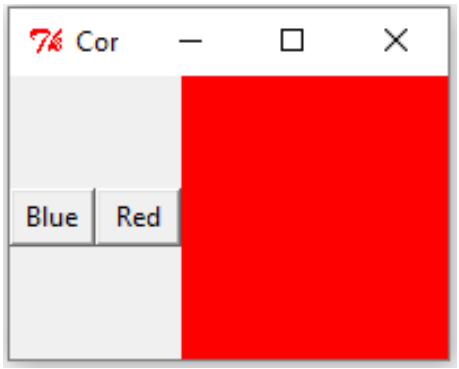
1. Interfaces gráficas - Tkinter

- Modificando os parâmetros das widgets: parâmetros são guardados em um dicionário em que os nomes dos parâmetros são as chaves.

```
from tkinter import *
def blueLabel():
    lbl['bg']='Blue'
def redLabel():
    lbl['bg']='Red'
root = Tk()
root.title('Cor')
blueButton = Button(root, text="Blue",\
                    command=blueLabel,width=4)
blueButton.pack(side=LEFT)
redButton = Button(root, text="Red",\
                   command=redLabel,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```

Modifica a cor da label lbl.





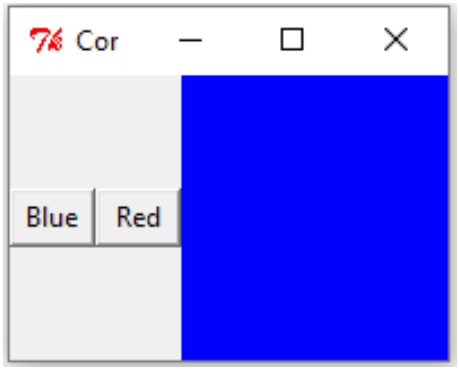
\Exemplos\ex01ButtonLabelWithPack_Python3.py

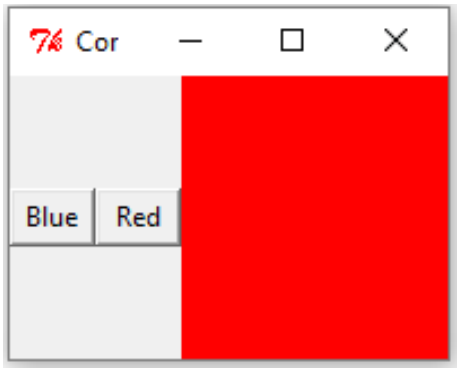
1. Interfaces gráficas - Tkinter

- Modificando os parâmetros das widgets: parâmetros são guardados em um dicionário em que os nomes dos parâmetros são as chaves.

```
from tkinter import *
def blueLabel():
    lbl['bg']='Blue'
def redLabel():
    lbl['bg']='Red'
root = Tk()
root.title('Cor')
blueButton = Button(root, text="Blue",\
                    command=blueLabel,width=4)
blueButton.pack(side=LEFT)
redButton = Button(root, text="Red",\
                   command=redLabel,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```

Lembrando: no Python 2 o nome do módulo é escrito com T maiúsculo:
from Tkinter import *



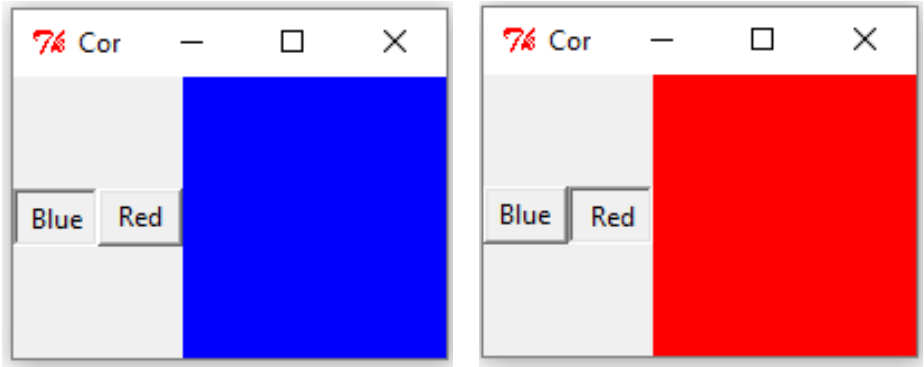


\Exemplos\ex01ButtonLabelWithPack_Python3.py

1. Interfaces gráficas - Tkinter

- Mantendo o botão pressionado:

```
from tkinter import *
def blueLabel():
    lbl['bg']='Blue'
    blueButton['relief']='sunken'
    redButton['relief']='raised'
def redLabel():
    lbl['bg']='Red'
    redButton['relief']='sunken'
    blueButton['relief']='raised'
root = Tk()
root.title('Cor')
blueButton = Button(root, text='Blue',\
                    command=blueLabel,width=4,relief=SUNKEN)
blueButton.pack(side=LEFT)
redButton = Button(root, text='Red',\
                   command=redLabel,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```

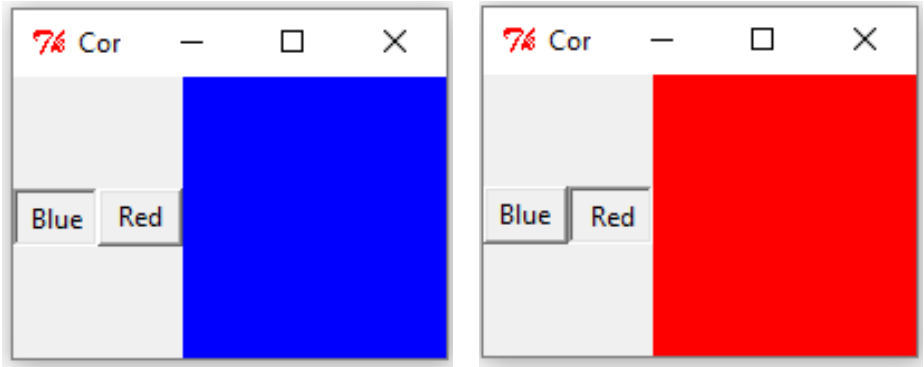


\Exemplos\ex02ButtonLabelWithPackPressed_Python3.py

1. Interfaces gráficas - Tkinter

- Mantendo o botão pressionado:

```
from tkinter import *
def blueLabel():
    lbl['bg']='Blue'
    blueButton['relief']='sunken'
    redButton['relief']='raised'
def redLabel():
    lbl['bg']='Red'
    redButton['relief']='sunken'
    blueButton['relief']='raised'
root = Tk()
root.title('Cor')
blueButton = Button(root, text='Blue',\
                    command=blueLabel,width=4,relief=SUNKEN)
blueButton.pack(side=LEFT)
redButton = Button(root, text='Red',\
                   command=redLabel,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```

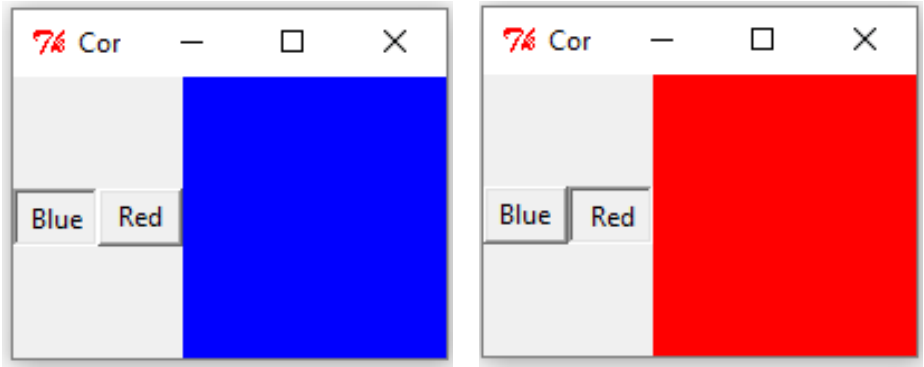


\Exemplos\ex02ButtonLabelWithPackPressed_Python3.py

1. Interfaces gráficas - Tkinter

- Método config:

```
from tkinter import *
def blueLabel():
    lbl.config(bg='Blue')
    blueButton.config(relief=SUNKEN)
    redButton.config(relief=RAISED)
def redLabel():
    lbl.config(bg='Red')
    redButton.config(relief=SUNKEN)
    blueButton.config(relief=RAISED)
root = Tk()
root.title('Cor')
blueButton = Button(root, text='Blue',\
                    command=blueLabel,width=4,relief=SUNKEN)
blueButton.pack(side=LEFT)
redButton = Button(root, text='Red',\
                   command=redLabel,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```



\Exemplos\ex03ButtonLabelWithPackConfig_Python3.py

1. Interfaces gráficas - Tkinter

- Tkinter possui classes para variáveis que permitem controlar quando ocorre alguma mudança.
  - BooleanVar, DoubleVar, IntVar, StringVar

```
from tkinter import *

def controlarVar(*arg):
    print('A variavel mudou')

root = Tk()
var = StringVar()
var.trace('w',controlarVar)
var.set('1')
var.set('2')
var.set('3')
```

\Exemplos\ex04StringVar_Python3.py

1. Interfaces gráficas - Tkinter

- Tkinter possui classes para variáveis que permitem controlar quando ocorre alguma mudança.
  - BooleanVar, DoubleVar, IntVar, StringVar

```
from tkinter import *

def controlarVar(*arg):
    print('A variavel mudou')

root = Tk()
var = StringVar()
var.trace('w',controlarVar)
var.set('1')
var.set('2')
var.set('3')
```

Quando a função é definida com um asterisco antes do nome da variável, a função recebe um número arbitrário de argumentos.

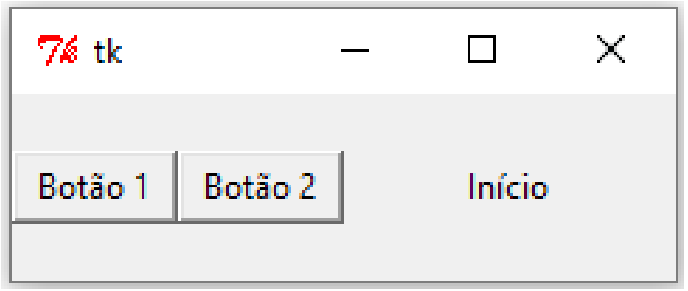
Por causa do método trace, a cada vez que set é chamado, a função controlarVar também é chamada.

\Exemplos\ex04StringVar_Python3.py

1. Interfaces gráficas - Tkinter

- Algumas widgets possuem parâmetros que podem ser associados a essas variáveis:

```
from tkinter import *
def textoLabel1():
    lblText.set('Botão 1 apertado')
    blueButton.config(relief=SUNKEN)
    redButton.config(relief=RAISED)
def textoLabel2():
    lblText.set('Agora o botão 2')
    redButton.config(relief=SUNKEN)
    blueButton.config(relief=RAISED)
root = Tk()
blueButton = Button(root, text='Botão 1',command=textoLabel1,width=7)
blueButton.pack(side=LEFT)
redButton = Button(root, text='Botão 2',command=textoLabel2,width=7)
redButton.pack(side=LEFT)
lblText = StringVar()
lblText.set('Início')
lbl = Label(root,textvariable=lblText,width=16, height=4)
lbl.pack(side=RIGHT)
root.mainloop()
```

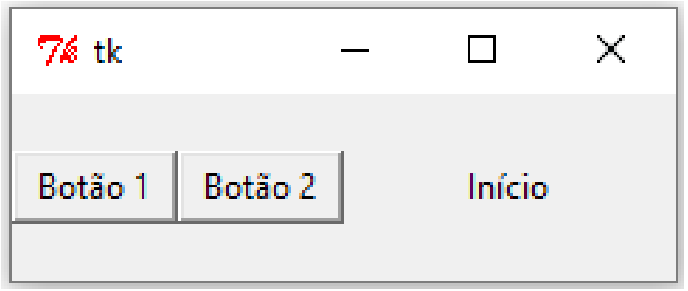


\Exemplos\ex05ButtonLabeltextvariable_Python3.py

1. Interfaces gráficas - Tkinter

- Algumas widgets possuem parâmetros que podem ser associados a essas variáveis:

```
from tkinter import *
def textoLabel1():
    lblText.set('Botão 1 apertado')
    blueButton.config(relief=SUNKEN)
    redButton.config(relief=RAISED)
def textoLabel2():
    lblText.set('Agora o botão 2')
    redButton.config(relief=SUNKEN)
    blueButton.config(relief=RAISED)
root = Tk()
blueButton = Button(root, text='Botão 1',command=textoLabel1,width=7)
blueButton.pack(side=LEFT)
redButton = Button(root, text='Botão 2',command=textoLabel2,width=7)
redButton.pack(side=LEFT)
lblText = StringVar()
lblText.set('Início')
lbl = Label(root,textvariable=lblText,width=16, height=4)
lbl.pack(side=RIGHT)
root.mainloop()
```



Criando uma variável do tipo string (StringVar) e associando essa variável à label através do parâmetro textvariable.

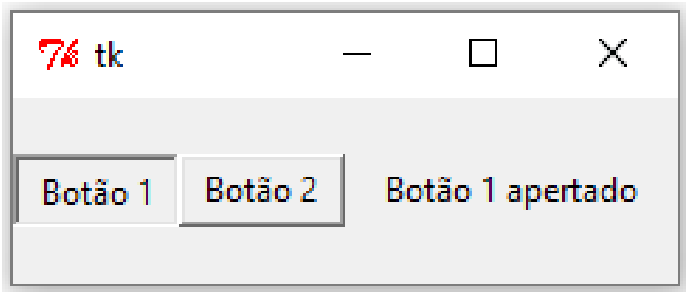
O texto inicial da label é definido como “Início”.

\Exemplos\ex05ButtonLabeltextvariable_Python3.py

1. Interfaces gráficas - Tkinter

- Algumas widgets possuem parâmetros que podem ser associados a essas variáveis:

Quando o botão é apartado, o texto é modificado utilizando a variável.




```
from tkinter import *
def textoLabel1():
    lblText.set('Botão 1 apertado')
    blueButton.config(relief=SUNKEN)
    redButton.config(relief=RAISED)
def textoLabel2():
    lblText.set('Agora o botão 2')
    redButton.config(relief=SUNKEN)
    blueButton.config(relief=RAISED)
root = Tk()
blueButton = Button(root, text='Botão 1',command=textoLabel1,width=7)
blueButton.pack(side=LEFT)
redButton = Button(root, text='Botão 2',command=textoLabel2,width=7)
redButton.pack(side=LEFT)
lblText = StringVar()
lblText.set('Início')
lbl = Label(root,textvariable=lblText,width=16, height=4)
lbl.pack(side=RIGHT)
root.mainloop()
```

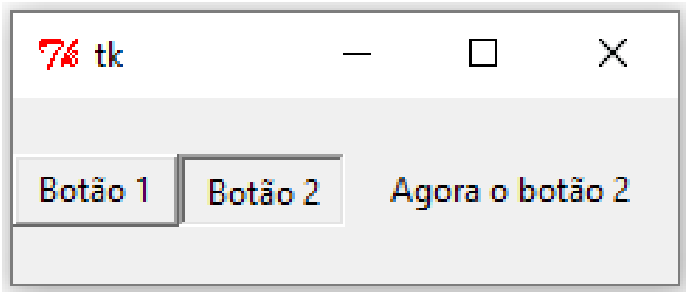
\Exemplos\ex05ButtonLabeltextvariable_Python3.py

1. Interfaces gráficas - Tkinter

- Algumas widgets possuem parâmetros que podem ser associados a essas variáveis:

```
from tkinter import *
def textoLabel1():
    lblText.set('Botão 1 apertado')
    blueButton.config(relief=SUNKEN)
    redButton.config(relief=RAISED)
def textoLabel2():
    lblText.set('Agora o botão 2')
    redButton.config(relief=SUNKEN)
    blueButton.config(relief=RAISED)
root = Tk()
blueButton = Button(root, text='Botão 1',command=textoLabel1,width=7)
blueButton.pack(side=LEFT)
redButton = Button(root, text='Botão 2',command=textoLabel2,width=7)
redButton.pack(side=LEFT)
lblText = StringVar()
lblText.set('Início')
lbl = Label(root,textvariable=lblText,width=16, height=4)
lbl.pack(side=RIGHT)
root.mainloop()
```

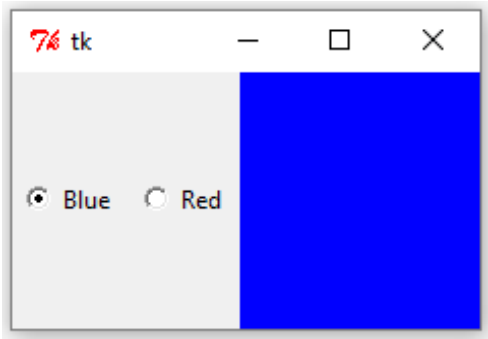
O outro botão coloca outro texto na label.



\Exemplos\ex05ButtonLabeltextvariable_Python3.py

1. Interfaces gráficas - Tkinter

- Um grupo de widgets Radiobutton devem ser associados a uma mesma variável:

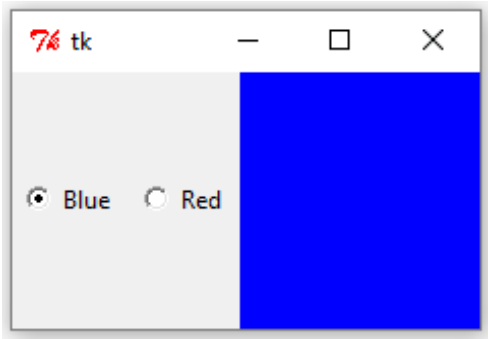



```
from tkinter import *
def colorLbl():
    lbl['bg']=var.get()

root = Tk()
var = StringVar()
var.set('Blue')
blueButton = Radiobutton(root, text='Blue',\
                         variable=var,value='Blue',\
                         command=colorLbl,width=4)
blueButton.pack(side=LEFT)
redButton = Radiobutton(root, text='Red',\
                        variable=var,value='Red',\
                        command=colorLbl,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```

\Exemplos\ex06RadioButtonLabel_Python3.py

1. Interfaces gráficas - Tkinter

- Um grupo de widgets Radiobutton devem ser associados a uma mesma variável:

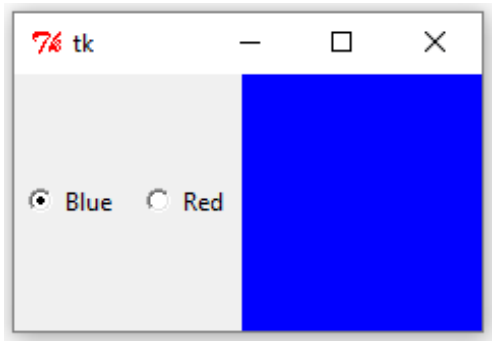



```
from tkinter import *
def colorLbl():
    lbl['bg']=var.get()

root = Tk()
var = StringVar()
var.set('Blue')
blueButton = Radiobutton(root, text='Blue',\
                         variable=var,value='Blue',\
                         command=colorLbl,width=4)
blueButton.pack(side=LEFT)
redButton = Radiobutton(root, text='Red',\
                        variable=var,value='Red',\
                        command=colorLbl,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```

\Exemplos\ex06RadioButtonLabel_Python3.py

1. Interfaces gráficas - Tkinter

- Um grupo de widgets Radiobutton devem ser associados a uma mesma variável:



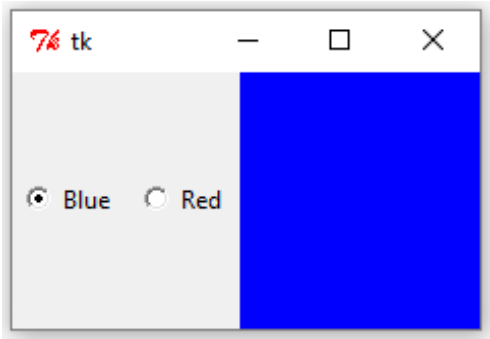

```
from tkinter import *
def colorLbl():
    lbl['bg']=var.get()

root = Tk()
var = StringVar()
var.set('Blue')
blueButton = Radiobutton(root, text='Blue',\
                         variable=var,value='Blue',\
                         command=colorLbl,width=4)
blueButton.pack(side=LEFT)
redButton = Radiobutton(root, text='Red',\
                        variable=var,value='Red',\
                        command=colorLbl,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```

Criando uma variável do tipo StringVar e inicializando essa variável.

Se a variável não for iniciada ela é definida por padrão como uma string vazia e todos os botões começam selecionados. Como inicializamos com o valor do primeiro botão (value='Blue'), este aparece selecionado. Inicializando com 'Red', o segundo aparece selecionado. Inicializando com um valor diferente de vazio, 'Blue' e 'Red' nenhum botão aparece selecionado.

1. Interfaces gráficas - Tkinter

- Um grupo de widgets Radiobutton devem ser associados a uma mesma variável:



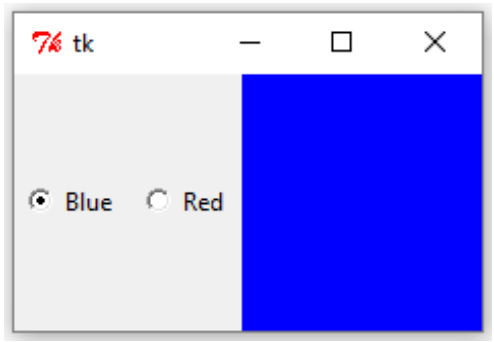

```
from tkinter import *
def colorLbl():
    lbl['bg']=var.get()

root = Tk()
var = StringVar()
var.set('Blue')
blueButton = Radiobutton(root, text='Blue',\
                         variable=var,value='Blue',\
                         command=colorLbl,width=4)
blueButton.pack(side=LEFT)
redButton = Radiobutton(root, text='Red',\
                        variable=var,value='Red',\
                        command=colorLbl,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```

Os dois botões são associados à mesma variável *var* do tipo StringVar. Quando o botão é selecionado, o valor da variável *var* é modificado para o valor determinado pelo parâmetro value e a função colorLbl é chamada.

\Exemplos\ex06RadioButtonLabel_Python3.py

1. Interfaces gráficas - Tkinter

- Um grupo de widgets Radiobutton devem ser associados a uma mesma variável:



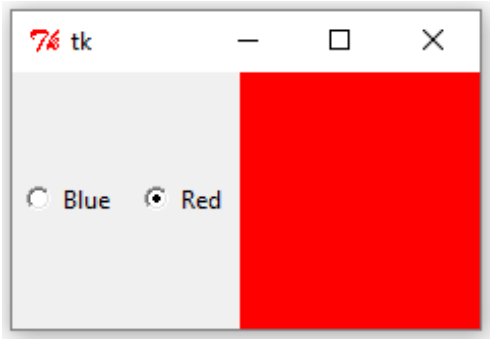

```
from tkinter import *
def colorLbl():
    lbl['bg']=var.get()

root = Tk()
var = StringVar()
var.set('Blue')
blueButton = Radiobutton(root, text='Blue',\
                         variable=var,value='Blue',\
                         command=colorLbl,width=4)
blueButton.pack(side=LEFT)
redButton = Radiobutton(root, text='Red',\
                        variable=var,value='Red',\
                        command=colorLbl,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```

A função colorLbl modifica a cor de acordo com o valor guardado na variável *var*.

\Exemplos\ex06RadioButtonLabel_Python3.py

1. Interfaces gráficas - Tkinter

- Um grupo de widgets Radiobutton devem ser associados a uma mesma variável:




```
from tkinter import *
def colorLbl():
    lbl['bg']=var.get()

root = Tk()
var = StringVar()
var.set('Blue')
blueButton = Radiobutton(root, text='Blue',\
                         variable=var,value='Blue',\
                         command=colorLbl,width=4)
blueButton.pack(side=LEFT)
redButton = Radiobutton(root, text='Red',\
                        variable=var,value='Red',\
                        command=colorLbl,width=4)
redButton.pack(side=LEFT)
lbl = Label(root,bg='blue',width=16, height=8)
lbl.pack(side=RIGHT)
root.mainloop()
```

Utilizando o método get(), a cor da label é modificada.

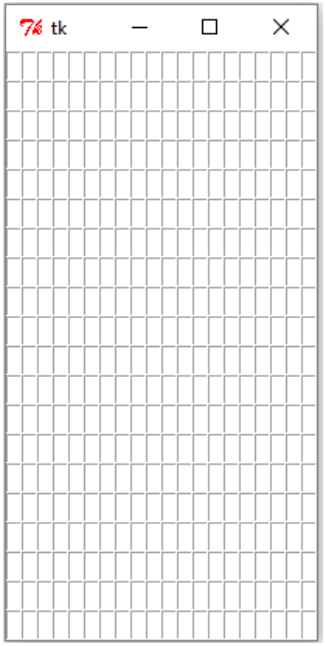
Outros tipos de variáveis também podem ser usados para iniciar

\Exemplos\ex06RadioButtonLabel_Python3.py

1. Interfaces gráficas - Tkinter

- Outro gerenciador de geometria - grid:
- Cria uma grade na janela da interface definindo células onde as widgets podem ser adicionadas.

```
from tkinter import *

root = Tk()

for r in range(0, 20):
    for c in range(0, 20):
        cell = Entry(root, width=1)
        cell.grid(row=r, column=c)

root.mainloop()
```

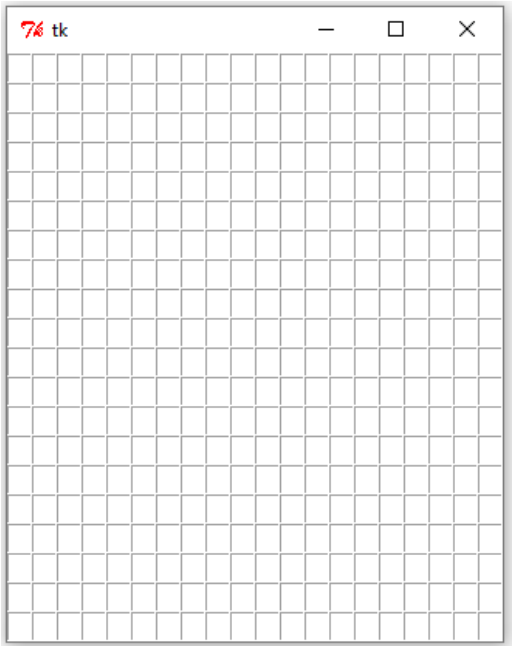


\Exemplos\ex07GridUnitario_Python3.py

1. Interfaces gráficas - Tkinter

- Modificando o tamanho da caixa de texto, o tamanho da célula também é modificado:

```
from tkinter import *

root = Tk()

for r in range(0, 20):
    for c in range(0, 20):
        cell = Entry(root, width=2)
        cell.grid(row=r, column=c)

root.mainloop()
```



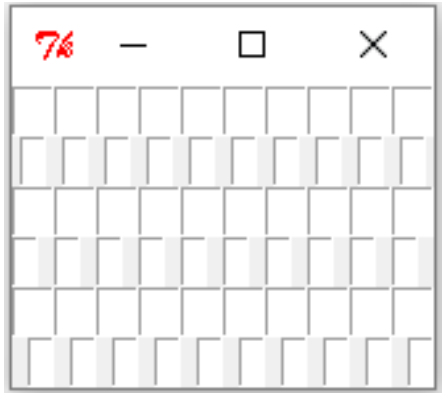
\Exemplos\ex08GridDuplo_Python3.py

1. Interfaces gráficas - Tkinter

- A célula assume do tamanho da maior widget na mesma linha e coluna.

  O parâmetro sticky indica a posição da widget dentro da célula.

```
from tkinter import *

root = Tk()
for r in range(0, 6):
    for c in range(0, 10):
        if r == 0 or r == 2 or r == 4:
            cell = Entry(root, width=2)
            cell.grid(row=r, column=c)
        elif r == 1:
            cell = Entry(root, width=1)
            cell.grid(row=r, column=c)
        elif r == 3:
            cell = Entry(root, width=1)
            cell.grid(row=r, column=c,sticky=W)
        else:
            cell = Entry(root, width=1)
            cell.grid(row=r, column=c,sticky=E)
root.mainloop()
```



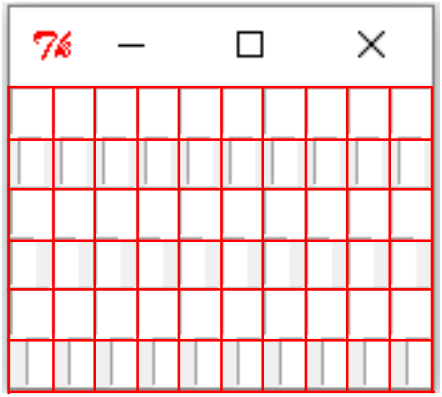
\Exemplos\ex09GridUnitarioEduplo_Python3.py

1. Interfaces gráficas - Tkinter

- A célula assume do tamanho da maior widget na mesma linha e coluna.

  O parâmetro sticky indica a posição da widget dentro da célula.

```
from tkinter import *

root = Tk()
for r in range(0, 6):
    for c in range(0, 10):
        if r == 0 or r == 2 or r == 4:
            cell = Entry(root, width=2)
            cell.grid(row=r, column=c)
        elif r == 1:
            cell = Entry(root, width=1)
            cell.grid(row=r, column=c)
        elif r == 3:
            cell = Entry(root, width=1)
            cell.grid(row=r, column=c,sticky=W)
        else:
            cell = Entry(root, width=1)
            cell.grid(row=r, column=c,sticky=E)
root.mainloop()
```

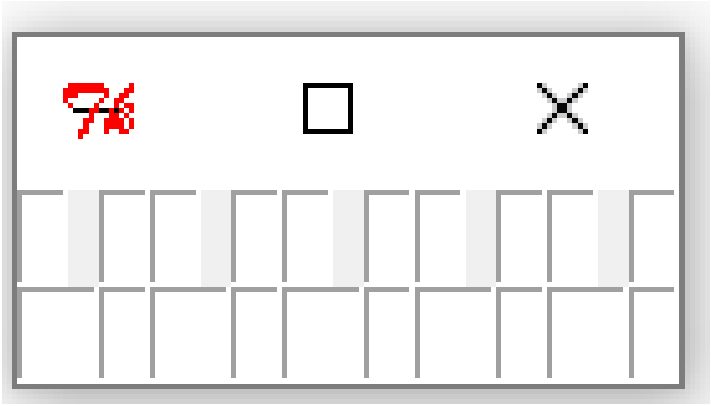


\Exemplos\ex09GridUnitarioEduplo_Python3.py

1. Interfaces gráficas - Tkinter

- Uma janela pode ter células de tamanhos diferentes:

```
from tkinter import *

root = Tk()

for c in range(0, 10):
    cell = Entry(root, width=1)
    cell.grid(row=0, column=c,sticky=W)

for c in range(0,10):
    if c%2 == 0:
        cell = Entry(root, width=2)
        cell.grid(row=1,column=c)
    else:
        cell = Entry(root, width=1)
        cell.grid(row=1,column=c)


root.mainloop()
```

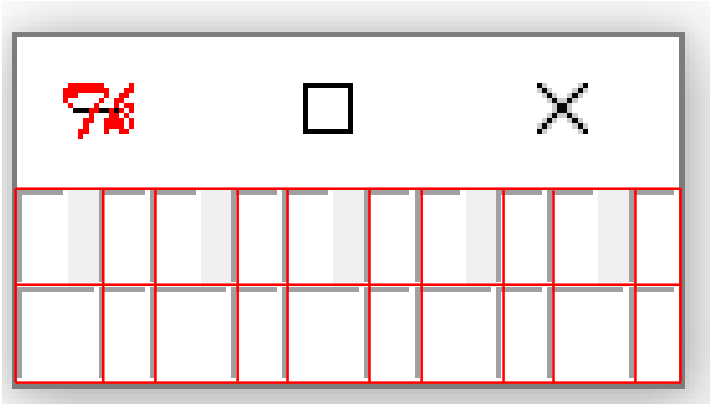


\Exemplos\ex10GridUnitarioEduploIrregular_Python3.py

1. Interfaces gráficas - Tkinter

- Uma janela pode ter células de tamanhos diferentes:

```
from tkinter import *

root = Tk()

for c in range(0, 10):
    cell = Entry(root, width=1)
    cell.grid(row=0, column=c,sticky=W)

for c in range(0,10):
    if c%2 == 0:
        cell = Entry(root, width=2)
        cell.grid(row=1,column=c)
    else:
        cell = Entry(root, width=1)
        cell.grid(row=1,column=c)


root.mainloop()
```

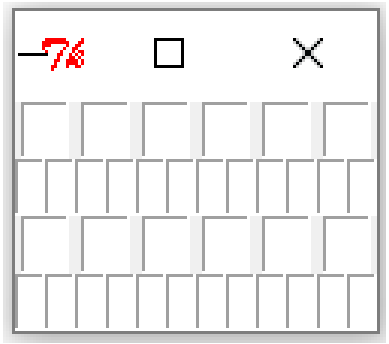


\Exemplos\ex10GridUnitarioEduploIrregular_Python3.py

1. Interfaces gráficas - Tkinter

- Colocando uma widget em mais de uma célula:
  - columnspan e rowspan

```
from tkinter import *

root = Tk()
for r in range(0, 4):
    for c in range(0, 12):
        if r == 0 or r == 2 or r == 4:
            cell = Entry(root, width=2)
            if c%2==0:
                cell.grid(row=r, column=c,columnspan=2)
        else:
            cell = Entry(root, width=1)
            cell.grid(row=r, column=c)
root.mainloop()
```

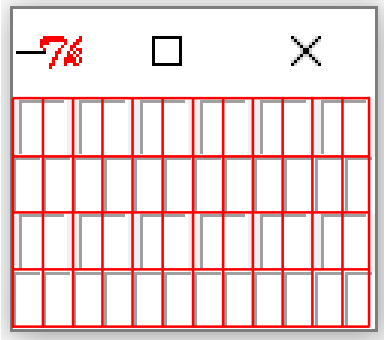


\Exemplos\ex11GridColumnspan_Python3.py

1. Interfaces gráficas - Tkinter

- Colocando uma widget em mais de uma célula:
  - columnspan e rowspan

```
from tkinter import *

root = Tk()
for r in range(0, 4):
    for c in range(0, 12):
        if r == 0 or r == 2 or r == 4:
            cell = Entry(root, width=2)
            if c%2==0:
                cell.grid(row=r, column=c,columnspan=2)
        else:
            cell = Entry(root, width=1)
            cell.grid(row=r, column=c)
root.mainloop()
```



\Exemplos\ex11GridColumnspan_Python3.py