TKINTER

- Biblioteca que permite interação com o user através de uma interface gráfica (GUI).
- Interfaces são criadas por janelas, menus, ícones onde permitem a interação com rato e teclado.
- A programação baseada em ambiente GUI é feita através de recurso a programação guiada por eventos (PGE).
- Desta forma, terá que existir um ciclo de inicialização que aguarda ocorrência de eventos.
- De cada vez que o evento ocorre, é encaminhado para a parte do programa que responde a esse evento.

TKINTER

- Biblioteca gráfica nativa do Python. Python faz uso desta biblioteca. Não é desenvolvida em Python (Tcl/Tk)
- Multiplataforma (Linux, Microsoft, …)
- from tkinter import * ##na versão 2 a biblioteca era chamada Tkinter

- Este módulo funciona com base em princípios básicos:
  - Definir uma janela;
  - Nessa janela são colocados um conjunto de elementos (widgets deriva da expressão Windows gadget)

TKINTER

- Objetos (tudo é um objeto)
  - Atributos (características dos objetos, como a cor, a largura, o ícone, …)
    - Métodos e eventos (permitem alterar os atributos)

Os métodos são ações que podem decorrer da intervenção do utilizador como redimensionar uma janela, minimizá-la, fazer clique num botão, …)

Eventos são o resultado da ação dos métodos. Podem ser ativados pela ação do utilizador como por ação da própria app, como por exemplo, ter na app um relógio que se atualiza de segundo ao segundo…

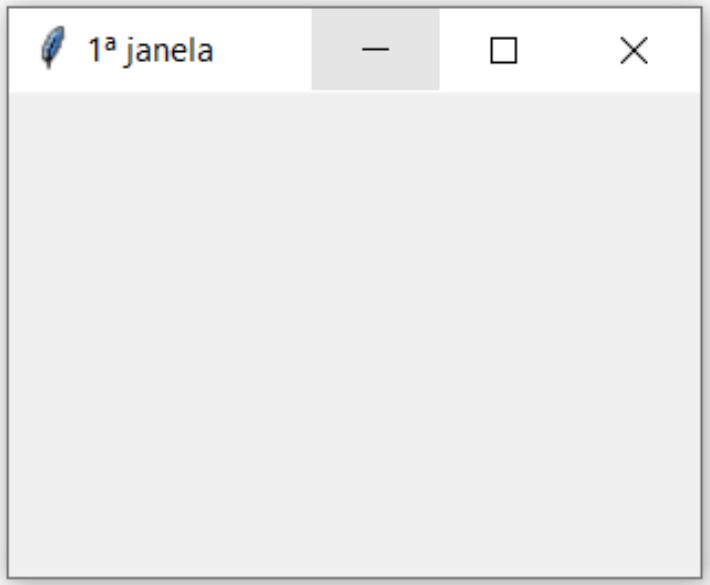
Criar janela

Para criar uma simples janela basta invocar uma instância Tk()

```
from tkinter import *
janela = Tk()

#invocar o ciclo de gestão de eventos
janela.mainloop()  #permite ficar à espera que ocorra qualquer evento
```

Janela com título e dimensão

```
from tkinter import *

janela=Tk()
##mudar título da janela
janela.title('1ª janela')

#definir dimensões da janela
janela.geometry('1250x1175')

janela.mainloop()
```



Neste exemplo, após executar podemos constatar que abre uma janela, com o título «1ª janela» com a dimensão 250x175 px

Redimensionar janelas

- resizable(largura, altura)
  - Os valores do método resizable são booleanos
  - Exemplo:

    janela.resizable(True, False)

Nota: em vez dos valores True e False podemos usar os números 1 e 0 respetivamente: janela.resizable(1, 0)

Redimensionar entre valores mínimos e máximos

- Podemos definir mínimos e máximos da resolução da janela

minsize(largura, altura)

```
janela.minsize(width = 800, height = 600)
```

maxsize(largura, altura)

```
janela.maxsize(1024, 768)
```

Os valores podem ser dados simplesmente com os números ou então utilizar as propriedades width e height

Arranque da janela Maximizada/Minimizada

- Podemos arrancar a nossa janela maximizada ou ainda minimizada.
- Para tal, devemos utilizar o método state(tipo)

- Exemplo:
  - janela.state("zoomed") #maximizado
  - janela.state("iconic") #minimizado

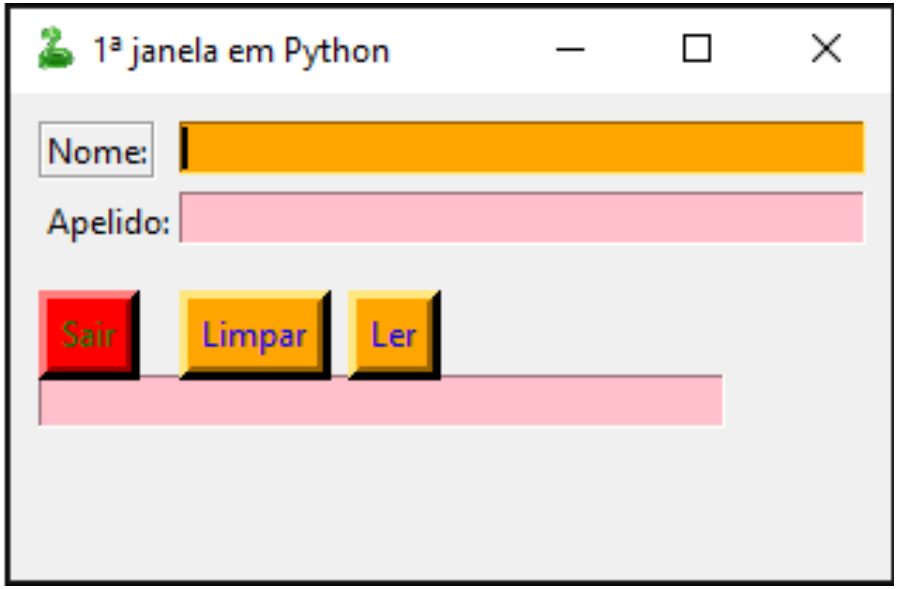
Janela – cor de fundo e posição fixa

```
from tkinter import *
janela=Tk()
##mudar título da janela
janela.title('1ª janela')
#definir dimensões da janela em pixeis
janela.geometry('1250x1175+200+200') ##larg x alt + Pos_x + Pos_y
##definir background
janela["bg"] = "black"  ##podemos usar a chave "bg" ou "background"
janela.mainloop()
```

Janela – alteração do ícone

- janela.iconbitmap("nome_ficheiro.ico")
  - O ficheiro deverá ter o formato .ico
  - Se indicarmos apenas o nome do ficheiro então este deverá estar na mesma localização do script, caso contrário, deveremos indicar o caminho relativo para ele.

Objetos simples



- Rótulos de texto (Label)
  - label1 = Label(janela, text="Nome:", relief=GROOVE)
- Caixas de texto (Entry)
  - entry1 = Entry(janela, width=40,bg="orange",fg="Blue")
- Botões (Button)
  - blimpa = Button(janela, text="Limpar", command=limpar, bg="orange", fg="blue", bd=5)

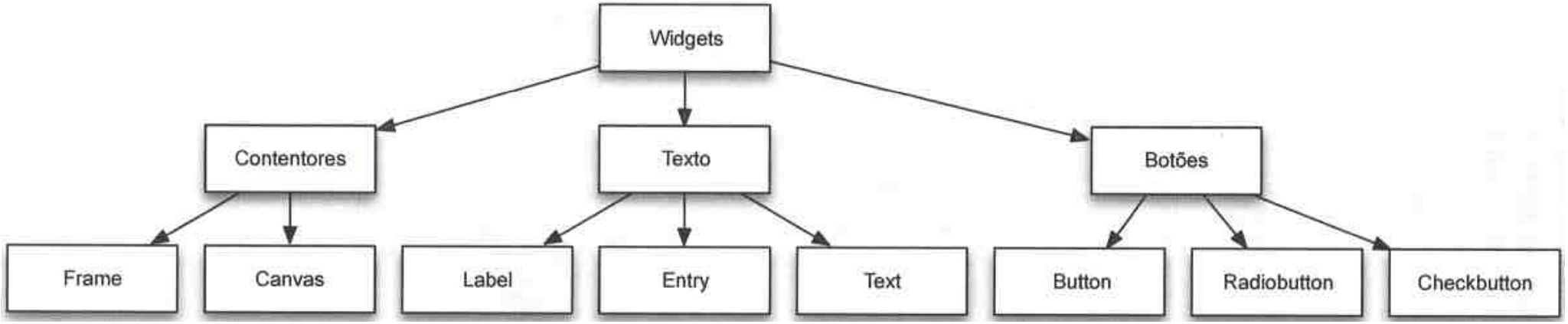
Colocação dos objetos

- Pack(opções)
  - expand - Quando definido como True, a janela expande-se para preencher qualquer espaço.
  - fill - determina se a janela preenche qualquer espaço: NONE (padrão), X (preencher apenas horizontalmente), Y (preencher apenas verticalmente) ou BOTH (preencher horizontalmente e verticalmente)
  - side - Determina qual lado a ser posicionado: “top” (padrão), “bottom”, “left” ou “right”.

Colocação dos objetos

- Grid(linha, coluna)
  - Funciona como uma tabela. Ajusta a dimensão das colunas e linhas
- Place(pos_x, pos_y)
  - O objeto é posicionado nas coordenadas especificadas. Pode sobrepor objetos ocultando outros.

Componentes de uma janela



Frame

- Frame funciona como um contentor que irá conter objetos.
- Primeiro passo, será criar uma Frame na janela que criamos anteriormente:

```
from tkinter import *
janela=Tk()
janela.title('1ª janela')
janela.geometry('250x175')


tela=Frame(janela)  #cria a frame
tela.grid() #o método grid() posiciona a frame na janela


janela.mainloop()
```

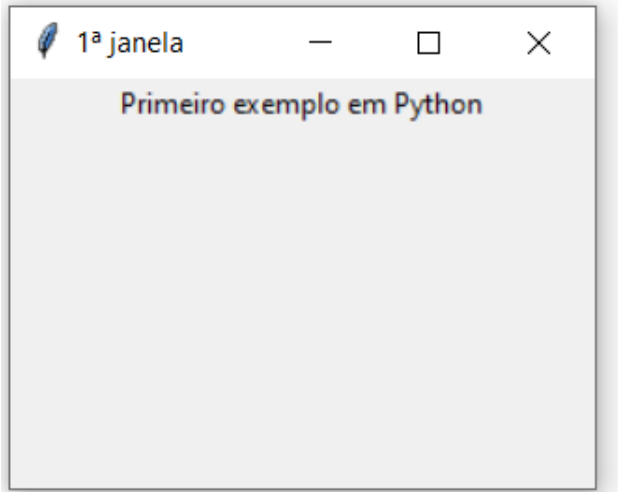
Frame

- Se pretendemos utilizar Frames para conter outros objetos, então esta deve ser criada em primeiro lugar, em seguida criar os objetos e posicioná-los na Frame e por fim posicionar a Frame:

```
from tkinter import *
janela=Tk()
frame1 = Frame(janela) # frame1 é filho de janela
#criar widgets
lnome = Label(frame1, text = "Nome: ")
enome = Entry(frame1)
#depois posicionamos na frame
lnome.grid(row = 0,column = 0, stick=W)
enome.grid(row = 0,column = 1, stick=W)
#posicionar a Frame
frame1.grid()
janela.mainloop()
```

Label

- Uma label funciona como uma etiqueta ou rótulo. Assim, ao associar uma label ao contentor Frame devemos indicar qual o texto que pretendemos.

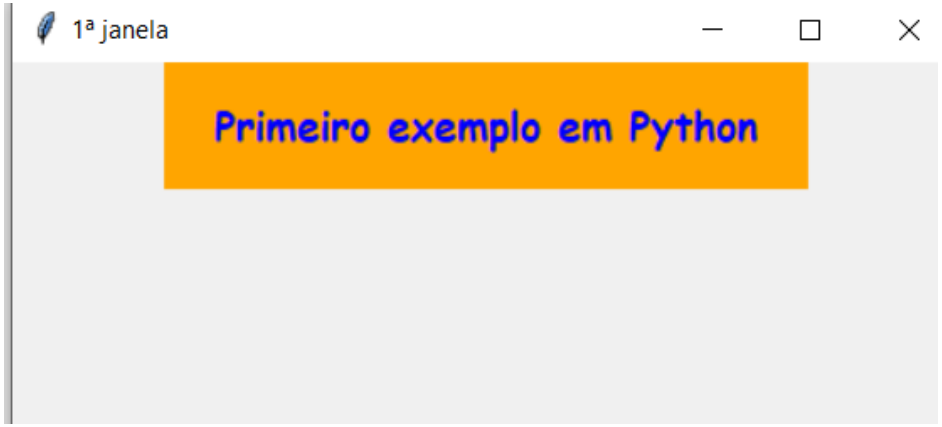



```
from tkinter import *

janela=Tk()

tela=Frame(janela)
tela.pack()

etiqueta=Label(tela, text="Primeiro exemplo em Python")
etiqueta.pack()

janela.mainloop()
```

Label - fontes

- Podemos definir o tipo de letra, a fonte e cores de background (bg) foreground(fg), altura e largura,.... Utilizamos o método configure()

```
etiqueta=Label(tela, text="Primeiro exemplo em Python")
fonte=("Comic Sans MS",14,"bold")
etiqueta.configure(font=fonte, width=25, height=2, bg="orange")
etiqueta.configure(fg="Blue")
etiqueta.pack()
```



Label – redimensionamento e relevo

- Se redimensionarmos a janela a nossa label está no estado imutável, ou seja, não se altera, não acompanha a "redimensão" da janela.
- Para o fazermos temos de recorrer às opções expand(YES ou NO) e fill(X,Y ou BOTH – expande em função do eixo do X, Y ou ambos).

etiqueta.pack(expand = YES, fill=BOTH)

- O relevo (relief) numa label pode ser dos seguintes tipo:
  - flat, solid, sunken, ridge, raised, groove

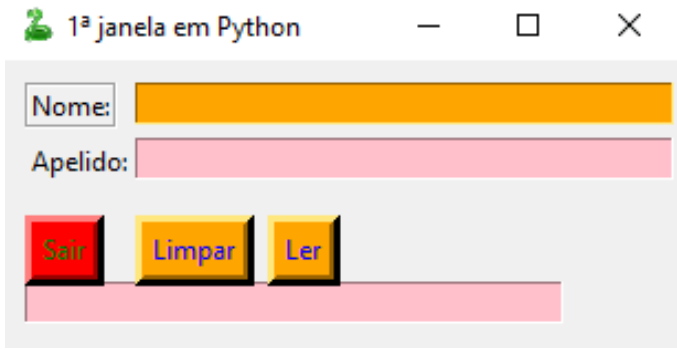
Label – redefinir as propriedades

- Por vezes, depois de termos a label apresentada graficamente, temos necessidade de redefinir as suas propriedades.
- O objeto Label, tem como representação interna o tipo "dictionary", onde o tipo de propriedade é a "key" e o valor é o "value"
- Assim, por exemplo, para alterar o texto de uma label:
  - label1["text"] = "Novo texto"
  - label1["relief"] = "sunken"

Entry – caixa de texto

- Caixa de texto que permite a inserção de dados por parte do utilizador.
- Podemos aplicar diversos tipos de configurações, tais como, o comprimento da caixa de texto, a cor de fundo e cor de texto ou ainda configurar uma fonte específica

```
etexto=Entry(janela, width=40,bg="pink",fg="Blue")
lnome = Label(janela, text="Nome:", relief=GROOVE)
enome = Entry(janela, width=40,bg="orange",fg="Blue")
lapelido=Label(janela, text="Apelido:")
eapelido = Entry(janela, width=40,bg="pink",fg="Blue")
```



Entry – escondendo os carateres

- Por vezes, no caso de uma caixa de texto que represente uma password, não queremos que os carateres dessa password sejam mostrados à medida que o utilizador a escreve.
- Assim, devemos recorrer à key "show" indicando o carater que pretendemos utilizar para mascarar o carater real

```
lUser = Label(janela, text = "Username:")
eUser = Entry(janela, width=25)
lSenha = Label(janela, text = "Password:")
eSenha = Entry(janela, width=25, show="#")
bLer = Button(janela,text="Login...",bd=5,width=10)
bLimpar =Button(janela,text="Limpar",bd=5,width=10)
bSair = Button(janela,text="Sair",bd=5,fg='Yellow',bg='Red')
```

Entry – leitura e escrita

- Para obter os dados escritos numa ENTRY podemos recorrer ao método GET
  - nome = entry1.get()
- Quando pretendemos inserir algo numa ENTRY, utilizamos o método INSERT, indicando a partir de que posição pretendemos inserir os texto e qual o valor a inserir
  - entry1.insert(0,"texto")
- Para apagar o conteúdo de uma ENTRY podemos utilizar o método DELETE, indicando o início e o fim das posições
  - entry1.delete(0, END)

Estado de uma Entry

- Uma ENTRY poderá apresentar 1 dos 3 estados seguintes:
  - normal (Editável)
  - disabled (Não editável. Se pretende editar terá que mudar para estado normal)
  - readonly (Não editável. Se pretende editar terá que mudar para estado normal. Neste estado consegue-se ler o conteúdo)

Button

- O objeto Button permite-nos desencadear uma série de ações através do recurso a funções em python, graças à sua key "command"

```
bsair = Button(janela, text="Sair", command=abandono,
                    bg="red",fg="Green", bd=5)
```

```
def abandono():
    janela.destroy()
```