

The background features a complex network of nodes and lines. Nodes are represented by small circles in various colors: red, green, blue, cyan, and dark grey. These nodes are interconnected by a web of thin, grey lines, some of which are straight and others curved, creating a sense of dynamic movement and connectivity. The overall composition is layered, with a dark grey rectangular area on the left side that serves as a backdrop for the text. The right side of the image is dominated by the network diagram, which extends across the top and right edges of the frame.

Tkinter

PROGRAMAÇÃO EM PYTHON

Tkinter

O Tkinter tem vários pontos fortes. É multiplataforma, então o mesmo código funciona no Windows, macOS e Linux. Os elementos visuais são renderizados usando elementos nativos do sistema operacional, para que os aplicativos construídos com o Tkinter pareçam pertencer à plataforma onde são executados.

O Tkinter é leve e relativamente fácil de usar em comparação com outras estruturas. Isso o torna uma escolha atraente para a construção de aplicativos GUI em Python, especialmente para aplicativos onde um brilho moderno é desnecessário, e a principal prioridade é construir algo que seja funcional e multiplataforma rapidamente.

Tkinter

Primeiro temos que instalar o tkinter:

```
pip install tk
```

Com isso podemos fazer o nosso primeiro código, importando o tkinter e criando a primeira janela

```
import tkinter as tk
```

```
janela = tk.Tk()  
janela.mainloop()
```

Tkinter

Agora que temos uma janela, podemos adicionar um widget. Usamos a `tk.Label` para adicionar algum texto a uma janela. Criando um widget `Label` com o texto "Olá Mundo" e atribuindo a uma variável chamada `rotulo`:

```
rotulo = tk.Label(text="Olá Mundo")
```

A janela criada anteriormente não muda. Criando o widget `Label` não o adiciona à janela. Existem várias maneiras de adicionar widgets a uma janela. Agora, podemos utilizar o método `.pack()` do widget `Label`:

```
rotulo.pack()
```

Tkinter

Foi utilizado o comando `pack()`, mas existem outras formas de adicionar um elemento para a janela:

- `pack ()` - O gerenciador de geometria Pack empacota widgets em linhas ou colunas.
- `grid ()` - O gerenciador de geometria Grid coloca os widgets em uma tabela bidimensional. O widget mestre é dividido em várias linhas e colunas, e cada “célula” na tabela resultante pode conter um widget.
- `place ()` - O gerenciador de geometria Place é o mais simples dos três gerenciadores gerais de geometria fornecidos no Tkinter. Ele permite que você defina explicitamente a posição e o tamanho de uma janela, em termos absolutos ou em relação a outra janela.

Tkinter

Os widgets são os principais componentes da tkinter e existem alguns deles:

Label - Um widget usado para exibir texto na tela

Button - Um botão que pode conter texto e realizar uma ação quando clicado

Entry - Um widget de entrada de texto que permite apenas uma única linha de texto

Text - Um widget de entrada de texto que permite a entrada de texto em várias linhas

Frame - Uma região retangular usada para agrupar widgets relacionados

Label

Os widgets de Label exibem texto com a cor de texto do sistema padrão e a cor de fundo do texto do sistema padrão. Normalmente, são preto e branco, respectivamente, mas é possível ver cores diferentes se forem alteradas essas configurações no sistema operacional.

Também dá para controlar o texto do rótulo e as cores de fundo usando os parâmetros de background e foreground: (todas as cores possíveis no link: <http://www.tcl.tk/man/tcl8.5/TkCmd/colors.htm>)

```
rotulo = tk.Label(  
    text="Hello, Tkinter",  
    foreground="white",  
    background="black"  
)
```

Label

Os widgets de Label exibem texto com a cor de texto do sistema padrão e a cor de fundo do texto do sistema padrão. Normalmente, são preto e branco, respectivamente, mas é possível ver cores diferentes se forem alteradas essas configurações no sistema operacional.

Também dá para controlar o texto do rótulo e as cores de fundo usando os parâmetros de background e foreground:

```
rotulo = tk.Label(  
    text="Olá mundo",  
    foreground="white",  
    background="black"  
)
```

```
rotulo = tk.Label(  
    text=" Olá mundo",  
    fg="#FFFFFF",  
    bg="#000000"  
)
```


Label

Também podemos modificar o tamanho do label:

```
rotulo = tk.Label(  
    text="Olá mundo",  
    fg="white",  
    bg="black",  
    width=10,  
    height=10  
)
```

Button

Os widgets de Button são usados para exibir botões clicáveis. Eles podem ser configurados para chamar uma função sempre que forem clicados. Existem muitas semelhanças entre os widgets Button e Label. De muitas maneiras, um botão é apenas um rótulo clicável. Os mesmos argumentos de palavra-chave para criar e estilizar o Label funcionarão com widgets de Botão.

```
botao = tk.Button(  
    text="Clique aqui",  
    width=25,  
    height=5,  
    bg="blue",  
    fg="yellow"  
)
```

Button

Um dos atributos mais importantes do botão é o `command`, que recebe uma função que vai ser chamada quando o botão é clicado.

```
botao = tk.Button(text='Clique', command=janela.destroy)
```

Button

Um dos atributos mais importantes do botão é o `command`, que recebe uma função que vai ser chamada quando o botão é clicado.

```
def clique():  
    print("clicado")
```

```
botao = tk.Button(text='Clique', command=clique)
```

Entry

Quando precisamos de um pouco de texto de um usuário, como um nome ou endereço de e-mail, usamos um widget Entry. Eles exibem uma pequena caixa de texto na qual o usuário pode digitar algum texto. Criar e definir o estilo de um widget de Entrada funciona exatamente como os widgets de Label e Button. Por exemplo, o código a seguir cria um widget com um fundo azul, algum texto amarelo e uma largura de 50 unidades de texto:

```
entrada = tk.Entry(fg="yellow", bg="blue", width=50)
```

Entry

O interessante sobre os widgets do Entry não é como estilizá-los.

É como usá-los para obter a entrada de um usuário. Existem três operações principais:

- Recuperando texto com `.get ()`
- Excluindo texto com `.delete ()`
- Inserindo texto com `.insert ()`

Entry

Uma forma interessante de se trabalhar com o Entry é o uso de variáveis StringVar, que auxiliam no uso desses campos, elas são variáveis que ficam associadas e sincronizadas com esses campos se definirmos no momento da criação.

A StringVar possui as funções .get e .set, para resgatar o valor que está no campo e modificá-lo.

Entry

Vamos ver um exemplo, primeiro fazemos um Label e um Entry:

```
nome_var = tk.StringVar()

def clique():
    print("Seu nome é", nome_var.get())
    nome_var.set("")

lNome = tk.Label(text="Nome")
entrada = tk.Entry(textvariable=nome_var)

botao = tk.Button(text="Enviar", command=clique)
```


Menus

Além de muitos outros recursos, também conseguimos trabalhar com menus dentro do tkinter

```
menu = tk.Menu()

arquivo = tk.Menu(menu, tearoff = 0)
menu.add_cascade(label = 'Arquivo', menu = arquivo)
arquivo.add_command(label = 'Novo Arquivo', command = None)
arquivo.add_command(label = 'Abrir', command = None)
arquivo.add_command(label = 'Salvar', command = None)
arquivo.add_separator()
arquivo.add_command(label = 'Sair', command = janela.destroy)
```

Menus

Além de muitos outros recursos, também conseguimos trabalhar com menus dentro do tkinter

```
ajuda = tk.Menu(menu, tearoff = 0)
menu.add_cascade(label = 'Ajuda', menu = ajuda)
ajuda.add_command(label = 'Versão', command = None)
ajuda.add_command(label = 'Sobre', command = None)

janela.config(menu = menu)

janela.mainloop()
```

Frames

Existem muitos outros widgets possíveis de se trabalhar com o Tkinter, mas não conseguimos abordar todos aqui. A documentação sempre pode auxiliar com essa tarefa.

O último tipo de widget que vamos olhar rapidamente é o frame, ele serve como “container” para outros widgets. Até agora não foi utilizado, mas todo widget pede como primeiro parâmetro o container em que vai ficar, na hora da criação. Se ele fica diretamente na janela, esse parâmetro pode ser omitido, mas se fica em um Frame, por exemplo, precisa ser especificado.

Frames

```
frame1 = tk.Frame()
```

```
botao1 = tk.Button(frame1, text="legal")  
botao1.pack(side="left")  
botao2 = tk.Button(frame1, text="legal")  
botao2.pack(side="right")
```

```
frame2 = tk.Frame()
```

```
botao3 = tk.Button(frame2, text="legal")  
botao3.pack()  
botao4 = tk.Button(frame2, text="legal")  
botao4.pack()
```

```
frame1.pack()  
frame2.pack()
```

Exercícios

Como exercícios tentem criar uma calculadora, com os botões de numerais de 0 a 9, soma, subtração, multiplicação e divisão e o igual, além do botão de apagar um dígito ou tudo.

O primeiro passo é fazer ela de forma visual, o segundo é fazer ela ficar funcional, mostrando o resultado ao apertar o botão de igual.