

Apostila: Projeto "Gráficos e Listas" em Python

Você vai:

- Usar **listas** para armazenar dados relacionados
- Criar um **gráfico** usando o `pygal` biblioteca
- Carregue dados tendo seu programa **ler um arquivo**

```
main.py
1  #!/bin/python3
2  from pygal import Bar
3
4  # Create a chart
5
6
7  # Add data to the chart
8
9  # Display the chart
10
```

Caso precise instalar o pygal, (**Verifique primeiro, elabore o código e execute**)

No Windows

No **Prompt de comando** digite o seguinte e pressione o Enter chave:

```
pip install pygal
```

Aguarde a conclusão da instalação e, em seguida, continue com o projeto.

Em um Mac

No **Terminal** digite o seguinte e pressione o Enter chave:

```
pip3 install pygal
```

Aguarde a conclusão da instalação e, em seguida, continue com o projeto.

No Linux, incluindo o sistema operacional Raspberry Pi

No **Terminal** digite o seguinte e pressione o Enter chave:

```
pip install pygal
```

Aguarde a conclusão da instalação e, em seguida, continue com o projeto.

O projeto inicial já possui algum código para importar o `pygal` biblioteca, que você usará para desenhar seu gráfico.

principal.py

```
from pygal import bar
```

Fazer um gráfico

Encontre o `# Create a chart` comente e adicione código abaixo dele para criar um gráfico de barras chamado `chart`, dentro dos colchetes, dê um título ao seu gráfico.

principal.py

```
# Create a chart chart = Bar(title='Olympic medals')
```

Chamada `chart.render()` para exibir o gráfico.

principal.py

```
# Display the chart chart.render()
```

Teste: Execute seu código para ver o gráfico. Vai ficar vazio porque ainda não tem dados.

Olympic medals



Depurar: Se você vir um erro sobre `Bar()` ou `chart.render()` estar `not defined`:

- Se o erro for para `Bar()`, certifique-se de que tenha `B` maiusculo no início e colchetes no final
- Se o erro for para `chart.render()`, verifique se tem o `.` entre `chart` e `render`, bem como os colchetes no final

Depurar: Se você não estiver usando o editor de código Raspberry Pi, e o gráfico não tiver aparecido quando você executar seu código, substitua `chart.render()` com

```
chart.render_in_browser().
```

Adicionar alguns dados

Python pode armazenar dados relacionados como um **lista**. . Você pode criar listas usando colchetes `[]`. Os itens de uma lista são separados por vírgulas.

Crie três listas de dados para mostrar no seu gráfico.

Cada lista armazenará o nome de uma nação e o número de medalhas conquistadas por essa nação.

principal.py

```
7 | # Add data to the chart
8 | us = ['United States', 2399]
9 | gb = ['Great Britain', 1304]
10| fr = ['France', 751]
```

Quando você armazena algo em uma lista, ele recebe um **índice**. . Um índice é um número que informa a posição de um item em uma lista. Os índices de lista começam em 0, em vez de 1.

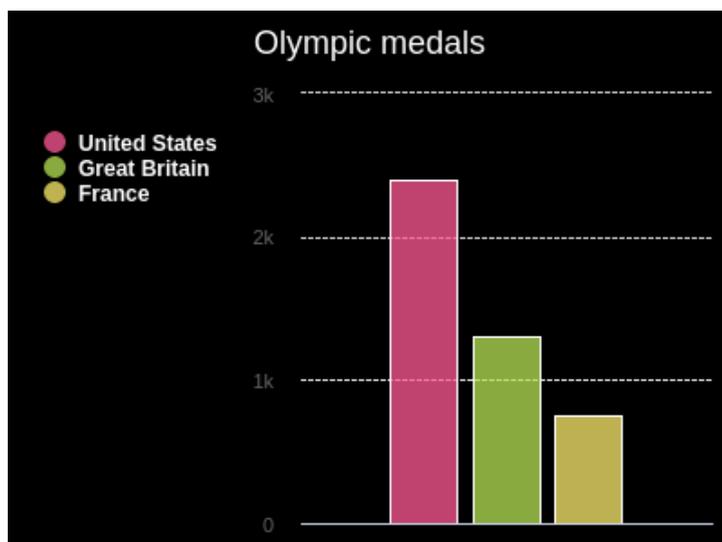
Você pode obter um item de uma lista pelo seu índice. Por exemplo, `my_list[3]` vai conseguir o **quarto** item em `my_list`, porque os índices vêm em 0.

Use os índices de suas listas e `chart.add()` para exibir seus dados. O nome da nação no item 0 será usado como um rótulo de categoria para o gráfico e a quantidade de medalhas no item 1 determinará a altura da barra.

principal.py

```
12 chart.add(us[0], us[1])
13 chart.add(gb[0], gb[1])
14 chart.add(fr[0], fr[1])
```

Teste: Execute seu código para ver o gráfico.



Agora carregue mais equipes adicionando novas listas e `chart.add()`

Elaborador: Cezar Augusto Trevisan