# Introdução à programação com Python

## Dia 2 - Avançando com Python


Mariana Brito Azevedo

Universidade Federal do Rio Grande do Norte

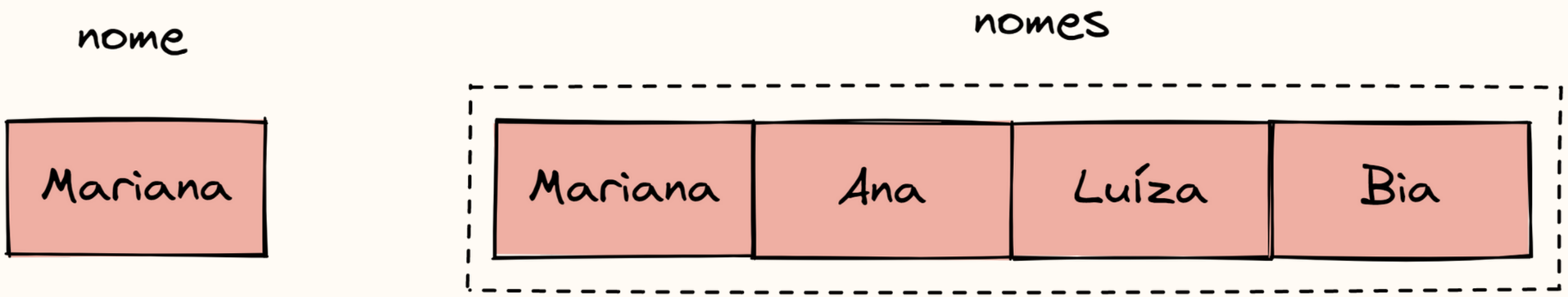
# Conteúdos

- Listas e matrizes
- Manipulando listas e matrizes
- Estruturas de repetição
- Trabalhando com tabelas: biblioteca Pandas
- Trabalhando com gráficos: biblioteca Plotly

# O que é uma lista?

- Antes, estávamos guardando apenas uma informação em uma variável (um número, uma palavra, uma frase...)
- Em uma lista, podemos guardar uma sequência de valores em uma única variável
- Exemplo: lista de compras

# O que é uma lista?

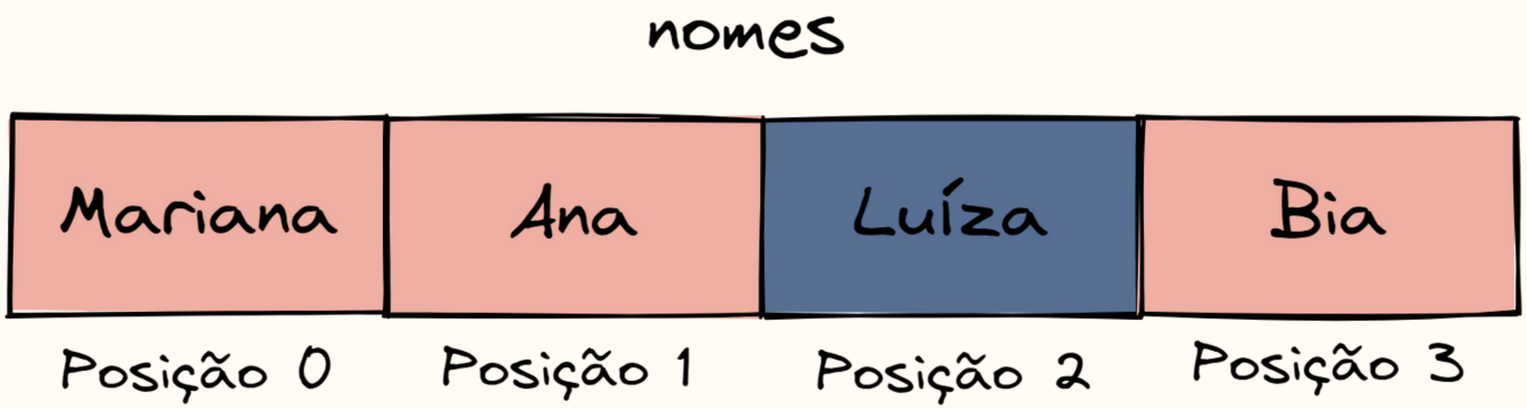



- A variável **nome** é do tipo **string**, guarda apenas uma informação
- A variável **nomes** é do tipo **lista**, guarda várias informações

```
#Criando variável do tipo string
nome = 'Mariana'

#Criando variável do tipo lista
nomes = ['Mariana', 'Ana', 'Luíza', 'Bia']
```

# Como acessar elementos de uma lista?

- Para acessar os elementos de uma lista, é preciso utilizar o índice daquele elemento
- O índice é a posição de um elemento em uma lista, e na programação, ele sempre começa pelo valor zero



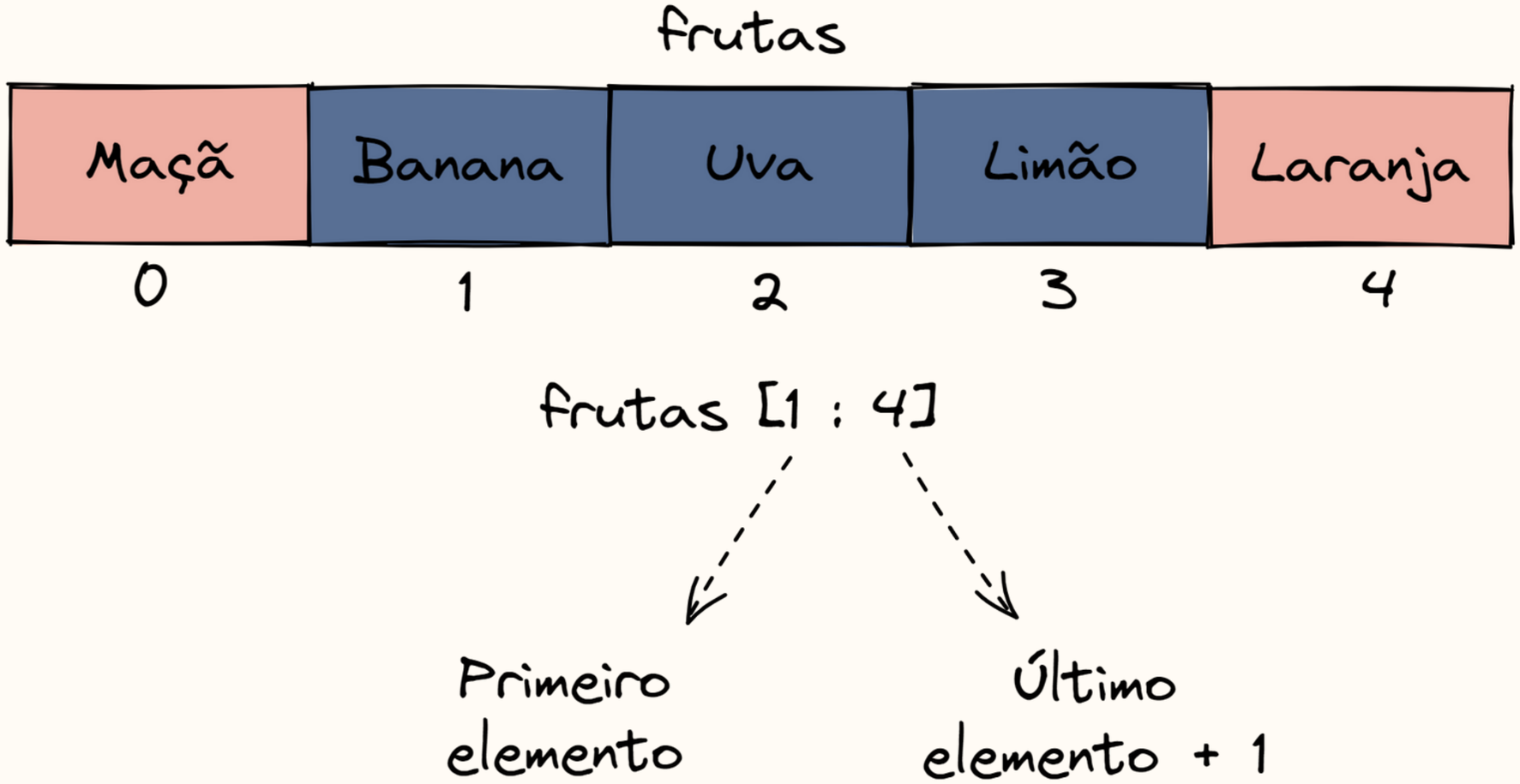

- Para acessar o nome Luíza, usamos nomes[2]

# Praticando


- Como posso acessar a fruta Banana? E a fruta Laranja?

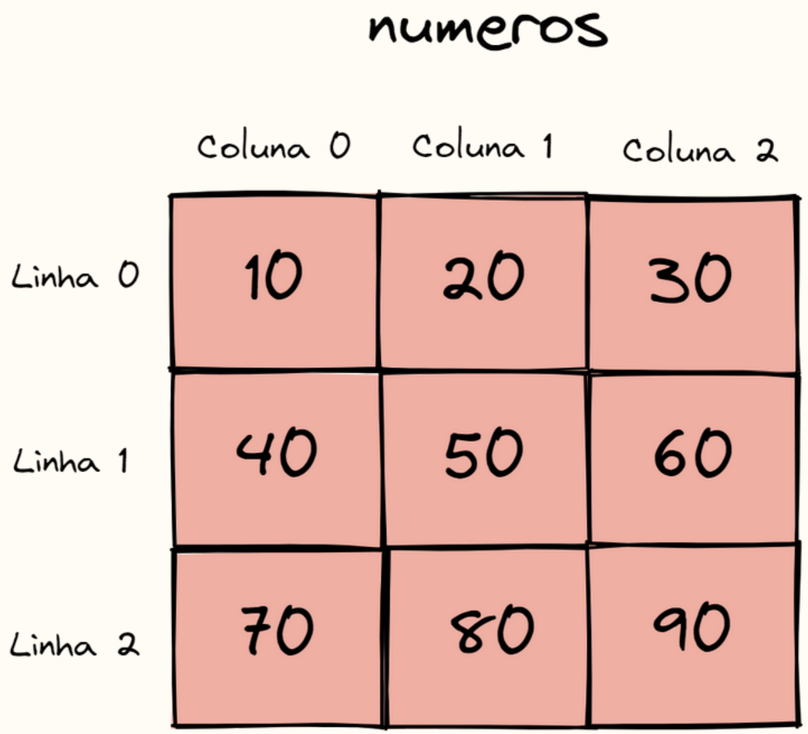
# Acessando mais de um valor da lista

- E se eu quiser acessar os valores banana, uva e limão?



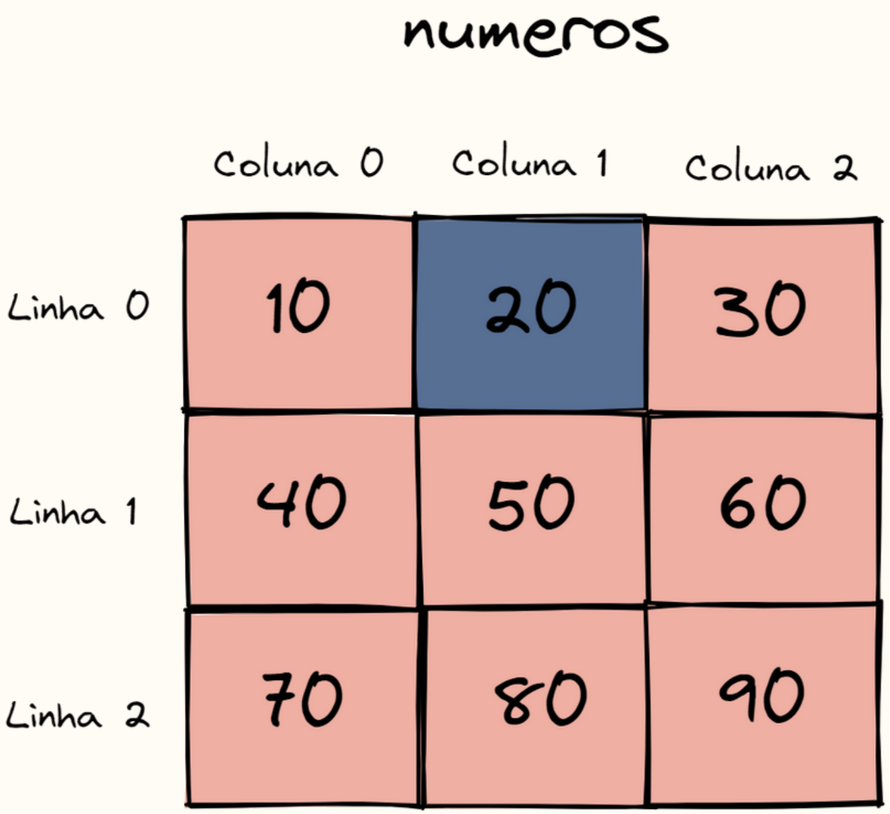
# O que é uma matriz?

- Uma matriz pode ser interpretada como uma lista de listas, ou simplesmente uma tabela
- Nós acessamos elementos de uma matriz a partir da sua posição da linha e coluna

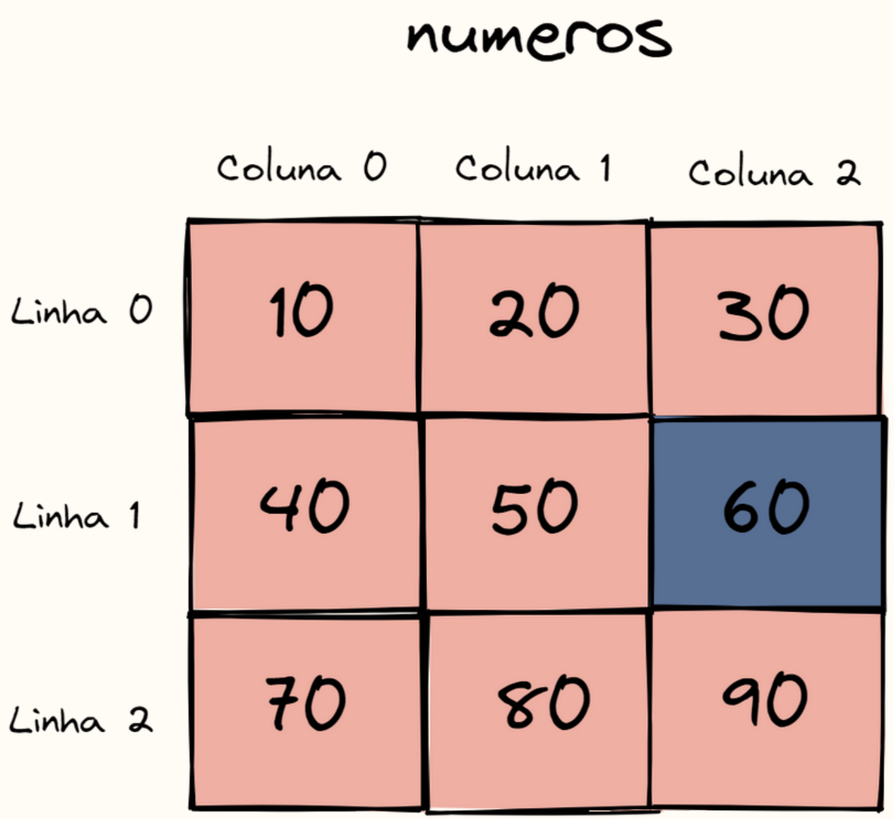



```
#Criando uma matriz
numeros = [[10, 20, 30],
           [40, 50, 60],
           [70, 80, 90]]
```

# Acessando elementos na matriz




numeros[0][1]




numeros[1][2]

# Estruturas de repetição

- Na programação, existem muitas situações em que precisamos fazer uma mesma coisa **várias vezes**
- **Exemplo:** contar de 1 a 10

```
#Maneira 1: contando de 1 a 10
print("Número " + 1)
print("Número " + 2)
print("Número " + 3)
print("Número " + 4)
print("Número " + 5)
print("Número " + 6)
print("Número " + 7)
print("Número " + 8)
print("Número " + 9)
print("Número " + 10)
```

Percebe que estamos fazendo a mesma coisa várias vezes, apenas mudando o número?

# Estruturas de repetição

- Podemos escrever o mesmo programa de antes utilizando uma estrutura de repetição **for** (em inglês, significa para)
- Colocamos uma **condição** para que a contagem seja feita do 1 ao 10

**Minha solução:**

para (números entre 1 e 10):
    então, faça a contagem

**Estrutura geral (python):**

for (condição):
    então, faça algo

# Estruturas de repetição

## ANTES

```
#Maneira 1: contando de 1 a 10
print("Número " + 1)
print("Número " + 2)
print("Número " + 3)
print("Número " + 4)
print("Número " + 5)
print("Número " + 6)
print("Número " + 7)
print("Número " + 8)
print("Número " + 9)
print("Número " + 10)
```

## DEPOIS

```
#Maneira 2: contando de 1 a 10
for numero in range(1, 11):
  print("Número " + numero)
```

# Outra estrutura de repetição: while

- Outro tipo de estrutura de repetição muito utilizada é o **while** (do inglês, enquanto)
- Enquanto algo for **falso**, a ação será repetida. Ao se tornar verdadeiro, a ação deixa de ser repetida
- **Exemplo:** fornecer uma senha

**Estrutura geral (python):**

```
while (condição):
    então, faça algo
```

# Outra estrutura de repetição: while

- Enquanto o usuário fornecer a senha errada, o programa irá pedir para ele verificar novamente a senha

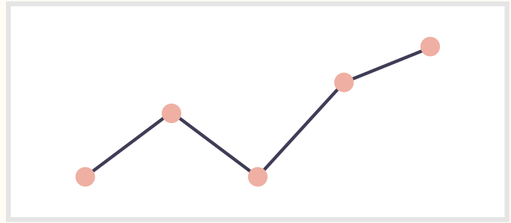
```
#Adivinhar senha
senha_correta = "senha1234"
senha_digitada = input("Digite sua senha")

while(senha_digitada != senha_correta):
  print("A senha digitada está incorreta, tente novamente")
  senha_digitada = input("Digite sua senha")

print("Senha correta digitada!")
```

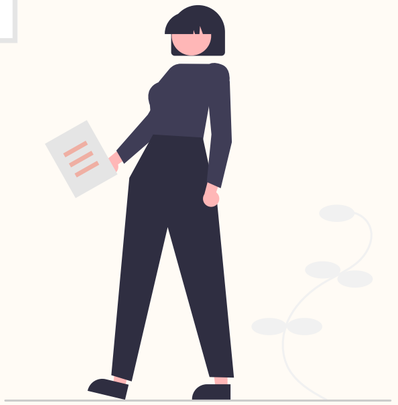
# Pandas: trabalhando com tabelas

- Na Geografia, existem muitas situações em que precisamos lidar com dados em tabelas
- **Exemplo:** crescimento da população brasileira ao longo dos anos

População brasileira: 1872 a 2018

| Ano | Total de habitantes |
|---|---|
| 1872 | 9.930.478 |
| 1890 | 14.333.915 |
| 1900 | 17.438.915 |
| 1920 | 30.635.605 |
| 1940 | 41.236.315 |
| 1950 | 51.944.397 |
| 1960 | 70.070.457 |
| 1970 | 94.508.583 |
| 1980 | 119.002.706 |
| 1991 | 146.825.475 |
| 2000 | 169.799.170 |
| 2010 | 190.755.799 |
| 2018 | 208.494.900 |





# Pandas: trabalhando com tabelas

- Para trabalhar com tabelas no Python, podemos utilizar uma biblioteca chamada Pandas
- Em Pandas, uma tabela é chamada de DataFrame
- O primeiro passo é ler os dados e criar a tabela

```
#Importando a biblioteca do pandas
import pandas as pd

#Criando a tabela com dados do IBGE
dados = pd.read_excel('dados_ibge.xlsx')

#Exibindo os dados das 5 primeiras linhas
dados.head()
```

# Pandas: trabalhando com tabelas

- Resultado da tabela em Python

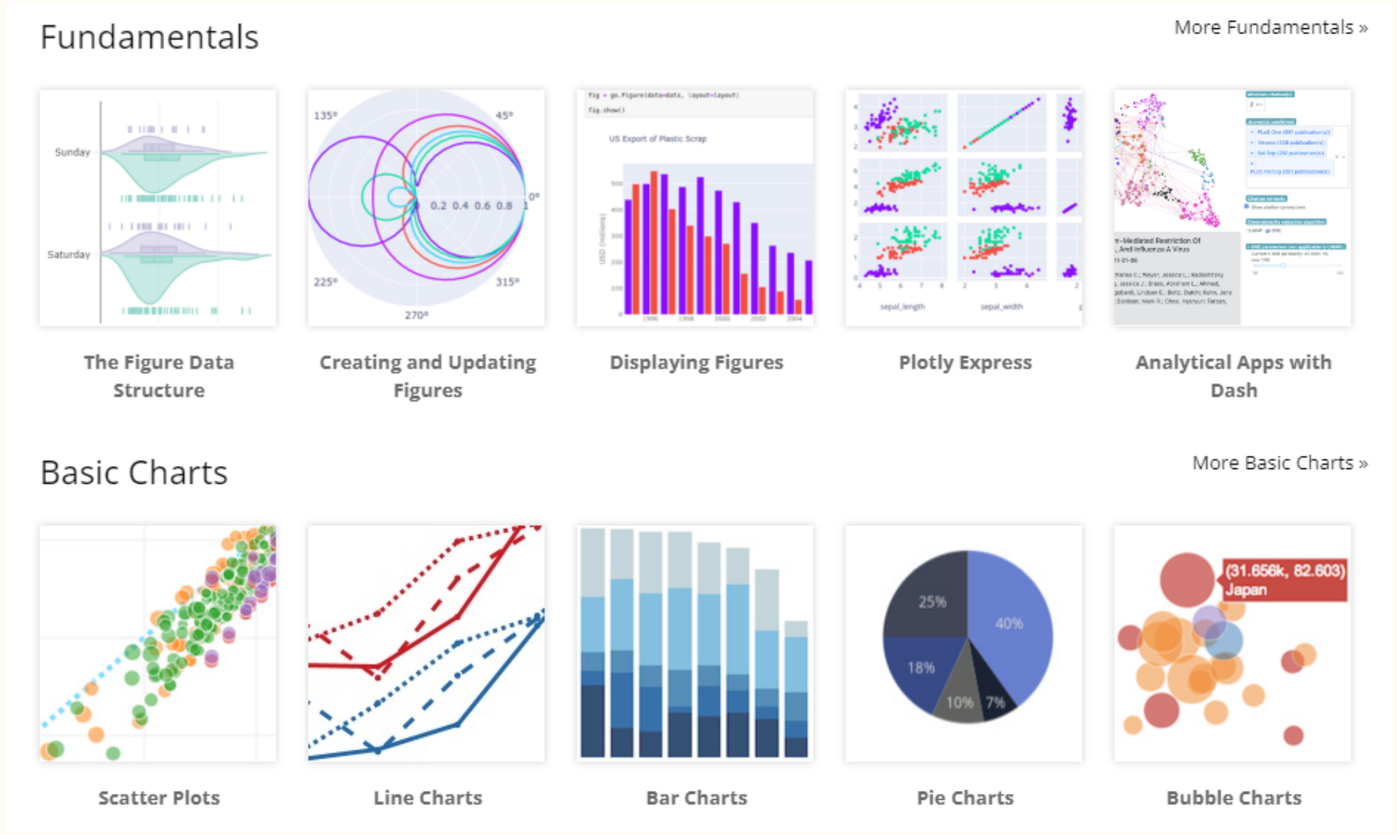
| | Ano | População |
|---|---|---|
| 0 | 1872 | 9930478 |
| 1 | 1890 | 14333915 |
| 2 | 1900 | 17438915 |
| 3 | 1920 | 30635605 |
| 4 | 1940 | 41236315 |

Muitas outras operações podem ser feitas com tabelas em Python, mas neste curso, vamos focar em como montar um gráfico a partir destes dados

OBS: percebe que a estrutura de uma tabela é semelhante a uma matriz?

# Plotly: trabalhando com gráficos

- A partir de tabelas, é possível montar diversos gráficos em Python utilizando a biblioteca Plotly




plotly.com/python/

Site para visualizar todos os gráficos que podem ser construídos com essa biblioteca

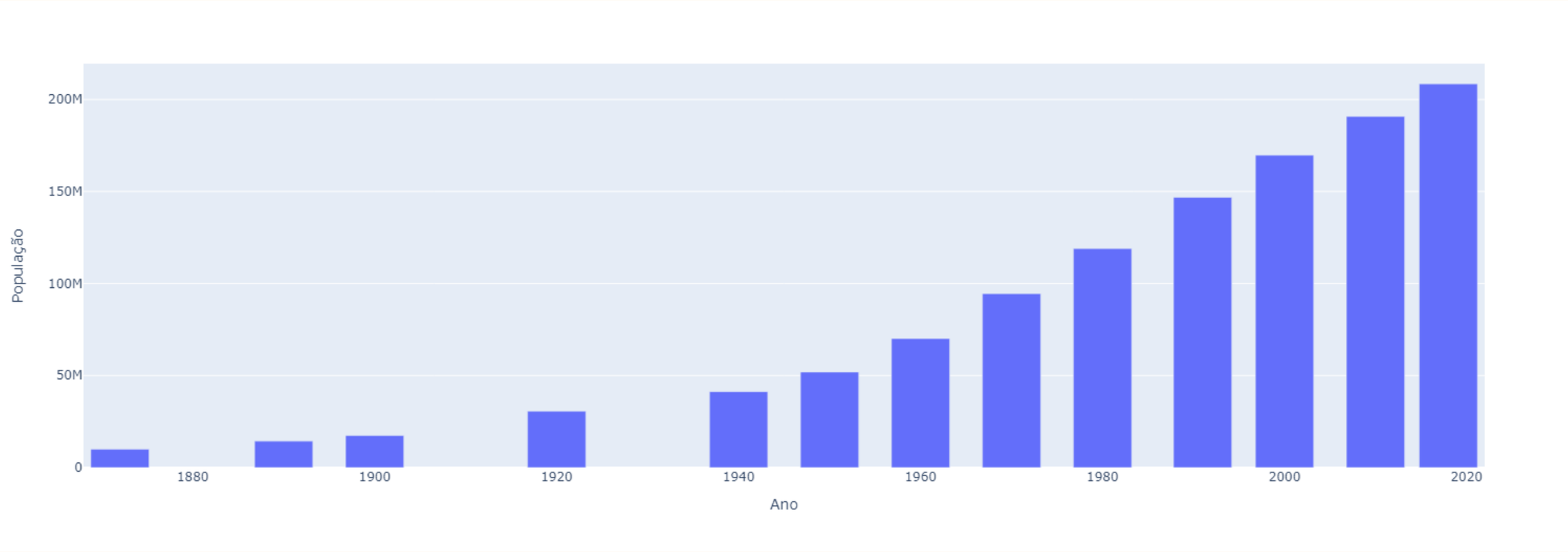
# Plotly: trabalhando com gráficos

- Para trabalhar com os dados do IBGE, podemos fazer um primeiro teste com um gráfico de barras
- Em plotly, um gráfico de barras se chama bar

```
# Importando a biblioteca do plotly
import plotly.express as px

# Criando uma figura para o gráfico
fig = px.bar(dados, x='Ano', y='População')

# Mostrando a figura criada
fig.show()
```

# Plotly: trabalhando com gráficos

- Resultado final:



# Plotly: trabalhando com gráficos

- Como podemos melhorar esse gráfico?

  Adicionar um título

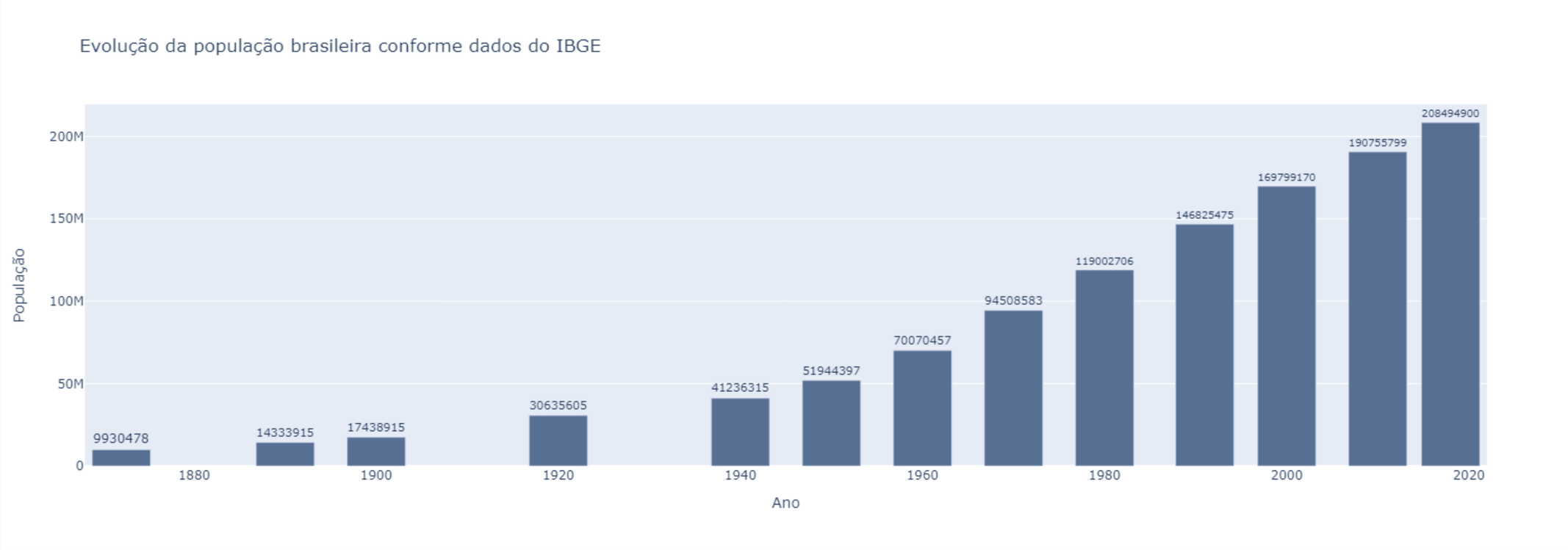
  Mudar as cores

  Mostrar valores exatos dos meus dados

```
# Importando a biblioteca do plotly
import plotly.express as px

# Criando uma figura para o gráfico
fig = px.bar(dados, x='Ano', y='População', title='Evolução da população brasileira conforme dados do IBGE',
color_discrete_sequence=['#586F94'], text='População')

# Mostrando a figura criada
fig.update_traces(textposition='outside')
fig.show()
```

# Plotly: trabalhando com gráficos

- Resultado final:



# Plotly: trabalhando com gráficos

Desafio: você consegue fazer agora um gráfico de linhas com estes dados?

# Hora de praticar!