



escola  
britânica de  
artes criativas  
& tecnologia

---

## Módulo | Python: Programação Orientada a Objetos

Caderno de **Exercícios**

Professor [André Perez](#)

---

### ▼ Tópicos

1. Um pouco de teoria;
  2. Classes;
  3. Objetos;
  4. Herança.
- 

### ▼ Exercício bônus

1. Classe para ler o arquivo csv

Crie a classe `ArquivoCSV2`. Ela deve estender (herdar) a classe `ArquivoTexto` para reaproveitar o seu atributos `self.arquivo` e o método `self.extrair_linha`.

A classe também deve conter o seguinte método:

- `self.extrair_coluna_da_linha`: Método que recebe como parâmetro o numero da linha e o indice da coluna e retorna o valor em questão.

```
class ArquivoCSV2(ArquivoTexto):
```

```

def __init__(self, arquivo: str):
    super().__init__(arquivo=arquivo)

def extrair_coluna_da_linha(self, numero_linha: int, indice_coluna: int):
    ...

arquivo_csv2 = ArquivoCSV2(arquivo='carros.csv')

numero_linha = 10
indice_coluna = 2
print(arquivo_csv2.extrair_coluna_da_linha(numero_linha=numero_linha, indice_coluna=indice

```

## ▼ Exercícios

### ▼ 0. Preparação do ambiente

Neste exercício vamos trabalhar com os arquivos de csv e texto definidos abaixo. Execute cada uma das células de código para escrever os arquivos na sua máquina virtual.

- **carros.csv**: arquivo csv com informações sobre carros (venda, manutenção, portas, etc.).

```

%%writefile carros.csv
id,valor_venda,valor_manutencao,portas,pessoas,porta_malas
1,vhigh,med,2,2,small
2,med,vhigh,2,2,small
3,low,vhigh,2,2,small
4,low,high,2,2,small
5,low,high,2,2,small
6,low,high,4,4,big
7,low,high,4,4,big
8,low,med,2,2,small
9,low,med,2,2,small
10,low,med,2,2,small
11,low,med,4,4,big
12,low,low,2,2,small
13,low,low,4,4,small
14,low,low,4,4,med

```

- **musica.txt**: arquivo texto com a letra da música Roda Viva do Chico Buarque.

```
%%writefile musica.txt
```

Roda Viva

Chico Buarque

Tem dias que a gente se sente  
Como quem partiu ou morreu  
A gente estancou de repente  
Ou foi o mundo então que cresceu  
A gente quer ter voz ativa  
No nosso destino mandar  
Mas eis que chega a roda viva  
E carrega o destino pra lá

Roda mundo, roda-gigante  
Roda moinho, roda pião

O tempo rodou num instante  
Nas voltas do meu coração  
A gente vai contra a corrente  
Até não poder resistir  
Na volta do barco é que sente  
O quanto deixou de cumprir  
Faz tempo que a gente cultiva  
A mais linda roseira que há  
Mas eis que chega a roda viva  
E carrega a roseira pra lá

Roda mundo, roda-gigante  
Roda moinho, roda pião

---

## 1. Classe para ler arquivos de texto

Crie a classe `ArquivoTexto`. Ela deve conter os seguintes atributos:

- `self.arquivo`: Atributo do tipo `str` com o nome do arquivo;
- `self.conteudo`: Atributo do tipo `list` onde cada elemento é uma linha do arquivo;

A classe também deve conter o seguinte método:

- `self.extrair_conteudo`: Método que realiza a leitura do arquivo e retorna o conteúdo.
- `self.extrair_linha`: Método que recebe como parâmetro o número da linha e retorna a

---

linha do conteúdo.

```
class ArquivoTexto(object):

    def __init__(self, arquivo: str):
        self.arquivo = ...
        self.conteudo = ...

    def extrair_conteudo(self):
        ...

    def extrair_linha(self, numero_linha: int):
        ...
```

Utilize o código abaixo para testar sua classe.

```
arquivo_texto = ArquivoTexto(arquivo='musica.txt')

numero_linha = 1
print(arquivo_texto.extrair_linha(numero_linha=numero_linha)) # Roda Viva

numero_linha = 10
print(arquivo_texto.extrair_linha(numero_linha=numero_linha)) # Mas eis que chega a roda v
```

---

## 2. Classe para ler arquivos de csv

Crie a classe `ArquivoCSV`. Ela deve estender (herdar) a classe `ArquivoTexto` para reaproveitar os seus atributos (`self.arquivo` e `self.conteudo`). Além disso, adicione o seguinte atributo:

- `self.colunas`: Atributo do tipo `list` onde os elementos são os nome das colunas;

A classe também deve conter o seguinte método:

- `self.extrair_nome_colunas`: Método que retorna o nome das colunas do arquivo.
- `extrair_coluna`: Método que recebe como parâmetro o índice da coluna e retorna o valor em questão.

```
class ArquivoCSV(ArquivoTexto):
```

```
def __init__(self, arquivo: str):
    super().__init__(arquivo=arquivo)
    self.colunas = ...

def extrair_nome_colunas(self):
    ...

def extrair_coluna(self, indice_coluna: int):
    ...
```

Utilize o código abaixo para testar sua classe.

```
arquivo_csv = ArquivoCSV(arquivo='carros.csv')

numero_linha = 1
print(arquivo_csv.extrair_linha(numero_linha=numero_linha)) # id,valor_venda,valor_manuten

print(arquivo_csv.colunas) # ['id', 'valor_venda', 'valor_manutencao', 'portas', 'pessoas'

numero_linha = 10
print(arquivo_csv.extrair_linha(numero_linha=numero_linha)) # 9,low,med,2,2,small

indice_coluna = 2
print(arquivo_csv.extrair_coluna(indice_coluna=indice_coluna)) # ['med', 'vhigh', 'vhigh',
```

---

