

Object-oriented programming: With CSV

Aqui veremos o uso de programação orientada a objetos com arquivos csv.
Para pegar todo o conteúdo de uma coluna e colocá-los em uma lista



```
1 class ArquivoCSV(object):
2
3     def __init__(self, arquivo: str):
4         self.arquivo = arquivo
5         self.conteudo = self._extrair_conteudo()
6         self.colunas = self._extrair_nome_colunas()
7
8     def _extrair_conteudo(self):
9         conteudo = None
10        with open(file=self.arquivo, mode='r', encoding='utf8') as arquivo:
11            conteudo = arquivo.readlines()
12        return conteudo
13
14    def _extrair_nome_colunas(self):
15        return self.conteudo[0].strip().split(sep=',')
16
17    def extrair_coluna(self, indice_coluna: str):
18        coluna = list()
19        for linha in self.conteudo:
20            conteudo_linha = linha.strip().split(sep=',')
21            coluna.append(conteudo_linha[indice_coluna])
22        coluna.pop(0)
23        return coluna
```

Primeiro devemos criar a Classe e suas funções e depois utilizá-la muitas outras vezes



```
1 arquivo_banco = ArquivoCSV(arquivo='./banco.csv')
```



```
1 education = arquivo_banco.extrair_coluna(indice_coluna=3)
2 print(education)
```

```
['primary', 'secondary', 'tertiary', 'tertiary', 'secondary', 'tertiary', 'tertiary',
'secondary', 'tertiary', 'primary']
```