

Capítulo 8 - Solução

December 20, 2020

```
[1]: # Importando o IPython.display
import IPython.display as ipy

# Importando os módulos de widgets
import ipywidgets as widget
from ipywidgets import interact

#Importando o módulo de leitura do CSV
import csv

## Função md

def md(Texto):
    display(ipy.Markdown(Texto))
```

1 Atividade 3: Para casa

1.1 Exercício 1: Lista de compras modificada

Modifique a Tarefa 1 mostrando o preço do produto e a quantidade em uma sub-lista dentro do item.

- O preço só deve ser mostrado na sub-lista se e somente se o usuário especificar
- O programa deve entender os preços no seguintes formatos:
 - 4.50
 - 4,50
 - R\$ 4,50
 - R\$ 4.50
- A quantidade deve ser mostrada se e somente se o usuário especificar.

Veja um exemplo:

A string:

```
Arroz - 4,50 - 2
Carne - R$ 12,40 - 4
Lentilha - - 1
Ração Whiskas - 16.30 - 2
Leite - 4,50
```

Resulta em:

- Arroz
 - Preço: R\$ 4,50
 - Qtd: 2
- Carne
 - Preço: R\$ 12,40
 - Qtd: 4
- Lentilha
 - Qtd: 1
- Ração Whiskas
 - Preço: R\$ 16,30
 - Qtd: 2
- Leite
 - Preço: R\$ 4,50

Total: R\$ 95,70

Solução:

```
[2]: # Substituindo a função no programa
md("Escreva a sua lista de compras abaixo:")
@interact(Lista=widget.Textarea(
    description="Lista: ",
    placeholder='Item - Preço - Qtd.')
)
def Lista_Saída(Lista):
    if Lista:
        ListaVec=Lista.splitlines()
        saídaLista="Saída da lista de compras:\n"
        Total = 0
        for Item in ListaVec:
            Item = Item.split('-')

            Item = [i.strip() for i in Item]

            #Verificando se o nome do item é válido
            if Item[0].replace(' ', '').isalpha():
                saídaLista+='- {} \n'.format(Item[0])

                Preço = 0 # Supõe que o preço é nulo
                if len(Item)>=2:
                    Item[1] = Item[1].replace(",",".").replace('R$', '').
→replace(' ', '')

                    if isfloat(Item[1]):
                        Preço = float(Item[1])
                        saídaLista+='    - Preço: ' + conv_Preço(Preço) + '\n'

                Qtd = 1 # Supõe que a quantidade é 1
```

```

        if len(Item)==3:
            if Item[2].isdigit():
                Qtd = int(Item[2].replace(' ',''))
                saídaLista+='      - Qtd: ' + str(Qtd) + '\n'

                #Incrementa o valor no total
                Total += Preço*Qtd

md(saídaLista)
md('**Total:** {}'.format(conv_Preço(Total)))

```

Escreva a sua lista de compras abaixo:

```

interactive(children=(Textarea(value='', description='Lista: ',
↳placeholder='Item - Preço - Qtd. '), Output()),...

```

1.2 Exercício 2: Tabela em Markdown

Crie a seguinte Tabela em Markdown:

Nome

Sexo

Idade

André

Masculino

45

Mariana

Feminino

32

João

Masculino

37

Pedro

Masculino

65

José

Masculino

93

Bianca

Feminino

40

Carlos

Feminino

44

Solução: | Nome | Sexo | Idade | | —: | :— | —: | | André | Masculino | 45 | | Mariana | Feminino | 32 | | João | Masculino | 37 | | Pedro | Masculino | 65 | | José | Masculino | 93 | | Bianca | Feminino | 40 | | Carlos | Feminino | 44 |

1.3 Exercício 3: Valor nutricional dos alimentos

O arquivo `csv` em anexo contém a base de dados do valor nutricional de alguns alimentos consumidos pelos americanos, considerando uma porção de 100 gramas. A versão original da base de dados pode ser encontrada em:

<https://tools.myfooddata.com/nutrition-facts-database-spreadsheet.php>

Neste exercício vamos utilizar o arquivo `comida.csv` que contém apenas um trecho da base de dados, com os seguintes campos:

- Name (Nome da comida)
- Food Group (Grupo da comida)
- Calories (Calorias)
- Fat (Gordura)
- Protein (Proteína)
- Carbohydrate (Carboidrato)

Utilizando os conceitos deste capítulo crie uma tabela dinâmica capaz de filtrar os seguintes campos:

- Food Group (Use um widget Dropdown)
- Calories, Fat, Protein e Carbohydrate (Use um widget do tipo RangeSlider)

Os RangeSlider tem as seguintes configurações:

- Calories (Calorias) → `min=17` e `max=642`
- Fat (Gordura) → `min=0` e `max=70`
- Protein (Proteína) → `min=0` e `max=40`
- Carbohydrate (Carboidrato) → `min=0` e `max=90`

A última linha da tabela deve conter o valor médio das colunas `Calories`, `Fat`, `Protein`, `Carbohydrate`. Esta linha deve estar em **negrito**.

Adicione as seguintes cores nos campos `Food Group`:

- `Vegatables` → Green
- `Sweets` → Fuchsia
- `Snacks` → Orange
- `Baked Foods` → DarkKhaki
- `Fruits` → Blue
- `Meats` → Red

Adicione um campo de busca no qual o usuário pode buscar pelo nome da comida (Campo Name). A busca deve ser case insensitive.

Observação: - Lembre-se que esse arquivo csv tem cabeçalho.

- Use apenas os conhecimentos ensinados até o momento (Não usar dicionários e nem pandas - Isso deixa a tarefa muito simples).

Dica: Pesquise sobre o método join para criar o cabeçalho.

Veja o exemplo abaixo:

```
[3]: Cabeçalho = ('Nome', 'Sexo', 'Idade')
saída = '|'+'|'.join(Cabeçalho)+'\n' #Método Join foi utilizado para criar o
↳cabeçalho usando o próprio cabeçalho da tabela
saída += '|+':-:|'*len(Cabeçalho)+'\n'
md(saída)
```

```
Nome  Sexo  Idade
```

Solução:

```
[4]: @interact(
    name = widget.Text(placeholder='Busque pelo nome'),
    FoodGroup = widget.
↳Dropdown(options=['Todos', 'Vegetables', 'Sweets', 'Snacks', 'Baked_
↳Foods', 'Fruits', 'Meats'],
        description='Food Group'),
    Calories = widget.IntRangeSlider(min=17,max=642,value=(17,642)),
    Fat = widget.FloatRangeSlider(min=0,max=70,value=(0,70),step=0.
↳1,readout_format='.1f'),
    Protein = widget.FloatRangeSlider(min=0,max=40,value=(0,40),step=0.
↳1,readout_format='.1f'),
    Carbohydrate = widget.FloatRangeSlider(min=0,max=90,value=(0,90),step=0.
↳1,readout_format='.1f')
)
def printTable(name, FoodGroup, Calories, Fat, Protein, Carbohydrate):
    with open('comida.csv') as dados:

        Header = True
        for comida in csv.reader(dados):
            # Criando o cabeçalho

            if Header:
                saída = '|'+'|'.join(comida)+'\n'
                saída += '|+':-:|'*len(comida)+'\n'
                Header=False
            else:
```

```

# Convertendo valores para float
comida[2:]=[float(valor) for valor in comida[2:]]

# Mudando a cor do grupo de comida
if comida[1]=='Vegetables':
    cor = 'Green'
elif comida[1]=='Sweets':
    cor = 'Fuchsia'
elif comida[1]=='Snacks':
    cor = 'Orange'
elif comida[1]=='Baked Foods':
    cor = 'DarkKhaki'
elif comida[1]=='Fruits':
    cor = 'Blue'
elif comida[1]=='Meats':
    cor = 'Red'

comida[1]='<span style="color:{}">{}</span>'.
↪format(cor,comida[1])

##Filtros

#Nome
if name.casefold() not in comida[0].casefold():
    continue

#Categoria
if FoodGroup != 'Todos' and FoodGroup not in comida[1]:
    continue

#Sliders
if comida[2] < Calories[0] or comida[2] > Calories[1]:
    continue

if comida[3] < Fat[0] or comida[3] > Fat[1]:
    continue

if comida[4] < Protein[0] or comida[4] > Protein[1]:
    continue

if comida[5] < Carbohydrate[0] or comida[5] > Carbohydrate[1]:
    continue

saída+='|{}|{}|{: .3n}|{: .3n}|{: .3n}|{: .3n}|\n'.format(*comida)
md(saída)

```

```
interactive(children=(Text(value='', description='name', placeholder='Busque  
↳ pelo nome'), Dropdown(description...
```