



Engenharia de Software
Computacional thinking with Python
Aula 08 – Trabalhando com mais de uma
função em um dataframe

Prof. Dr. Francisco Elânio

Funções podem chamar outras funções

É crucial compreender que cada função que escrevemos pode ser empregada e invocada por outras funções criadas por nós.

Essa prática representa uma das principais estratégias utilizadas por cientistas da computação para abordar problemas complexos:

fragmentando-os em problemas menores. Esse procedimento de dividir um problema em subproblemas menores é denominado ***decomposição funcional***.

Exemplo 1

Estrutura de repetição while em dataframe

```
import pandas as pd
```

```
dados = {'Nome': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],  
         'Idade': [25, 30, 35, 40, 45],  
         'Salário': [5000, 6000, 7000, 8000, 9000]}
```

```
df = pd.DataFrame(dados)
```

```
i = 0
```

```
while i < len(df):
```

```
    print("Nome:", df['Nome'][i], "- Idade:", df['Idade'][i])
```

```
    i += 1
```

Exemplo 2

Estrutura de repetição while em dataframe

```
import pandas as pd
```

```
dados = {'Nome': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],  
        'Idade': [25, 30, 35, 40, 45],  
        'Salário': [5000, 6000, 7000, 8000, 9000]}
```

```
df = pd.DataFrame(dados)
```

```
i = 0  
while i < len(df):  
    if df['Salário'][i] > 6000:  
        print(df.iloc[i])  
    i += 1
```


Exemplo 3

Estrutura de repetição for em dataframe

```
import pandas as pd
```

```
dados = {'Nome': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],  
         'Idade': [25, 30, 35, 40, 45],  
         'Salário': [5000, 6000, 7000, 8000, 9000]}
```

```
df = pd.DataFrame(dados)
```

```
for salario in df['Salário']:  
    print("Salário:", salario)
```

Exemplo 4

Utilizando for para calcular estatísticas descritivas de colunas numéricas

```
import pandas as pd
```

```
dados = {'Nome': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],  
         'Idade': [25, 30, 35, 40, 45],  
         'Salário': [5000, 6000, 7000, 8000, 9000]}
```

```
df = pd.DataFrame(dados)
```

```
for coluna in df.select_dtypes(include=['int', 'float']):  
    media = df[coluna].mean()  
    print("Média de", coluna, ":", media)
```

Exemplo 5

Utilizando for para calcular a contagem de valores únicos em cada coluna categórica

```
import pandas as pd
```

```
dados = {'Nome': ['Alice', 'Bob', 'Bob', 'David', 'Emily'],  
         'Gênero': ['F', 'M', 'M', 'M', 'F'],  
         'Cargo': ['Gerente', 'Analista', 'Diretor', 'Analista', 'Gerente']}
```

```
df = pd.DataFrame(dados)
```

```
for coluna in df.select_dtypes(include=['object']):  
    contagem_valores = df[coluna].value_counts()  
    print("Contagem de valores únicos em", coluna, ":", contagem_valores)
```

Exemplo 6

Utilizando for para contagem de valores

```
import pandas as pd

dados = {'Nome': ['Alice', 'Bob', 'Bob', 'David', 'Emily'],
         'Gênero': ['F', 'M', 'M', 'M', 'F'],
         'Cargo': ['Gerente', 'Analista', 'Diretor', 'Analista', 'Gerente']}

df = pd.DataFrame(dados)

for coluna in df.select_dtypes(include=['object']):
    contagem_valores = df[coluna].value_counts()
    print("Contagem de valores únicos em", coluna, ":", contagem_valores)
```


Exemplo 7

Utilizando while para calcular estatísticas descritivas de colunas numéricas, mas interrompendo o loop quando um determinado critério é atingido

```
import pandas as pd

dados = {'Nome': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],
        'Idade': [25, 30, 35, 40, 45],
        'Salário': [5000, 6000, 7000, 8000, 9000]}

df = pd.DataFrame(dados)

soma_salarios = 0
contador = 0
i = 0
while i < len(df):
    soma_salarios += df['Salário'][i]
    contador += 1
    media_salarios = soma_salarios / contador
    if media_salarios > 7000:
        break
    i += 1

print("Salário médio até o momento:", media_salarios)
```

Exemplo 8

Utilizando for para calcular bonus

```
import pandas as pd

dados = {'Nome': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],
        'Idade': [25, 30, 35, 40, 45],
        'Salário': [5000, 6000, 7000, 8000, 9000]}

df = pd.DataFrame(dados)

def calcular_bonus(salario):
    return salario * 0.1

bonus = []
for indice, linha in df.iterrows():
    bonus.append(calcular_bonus(linha['Salário']))

print("Bônus calculados:", bonus)
```

```

• import pandas as pd
•
dados = {'Nome': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],
        'Idade': [25, 30, 35, 40, 45],
        'Salário': [5000, 6000, 7000, 8000, 9000],
        'Cargo': ['gerente', 'coordenador', 'coordenador', 'gerente', 'gerente']}

```

```
df = pd.DataFrame(dados)
```

```
tipos_de_cargo = df['Cargo'].unique()
```

```

medias_por_cargo = {}
for cargo in tipos_de_cargo:
    sub_df = df[df['Cargo'] == cargo]
    media_salario = sub_df['Salário'].mean()
    media_idade = sub_df['Idade'].mean()
    medias_por_cargo[cargo] = {'Média Salário': media_salario, 'Média Idade': media_idade}

```

```

print("Média de salário e idade para cada tipo de cargo:")
for cargo, medias in medias_por_cargo.items():
    print(cargo + ":")
    print(" - Média Salário:", medias['Média Salário'])
    print(" - Média Idade:", medias['Média Idade'])

```

Exemplo 9

Usando for para calcular média de salário e idade por cargo.


```
import pandas as pd
```

```
dados = {'Nome': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],  
        'Idade': [25, 30, 35, 40, 45],  
        'Salário': [5000, 6000, 7000, 8000, 9000],  
        'Cargo': ['gerente', 'coordenador', 'coordenador',  
                  'gerente', 'gerente']}
```

```
df = pd.DataFrame(dados)
```

```
medias_por_cargo = df.groupby('Cargo').agg({'Salário':  
      'mean', 'Idade': 'mean'})
```

```
print("Média de salário e idade para cada tipo de  
cargo:")  
print(medias_por_cargo)
```

Exemplo 10

Usando for para calcular média de salário e idade por cargo.

Exemplo 11 – df sobre falhas em máquinas

Calcular a média do torque quando tipo é igual a M e o torque é maior do que 50 usando for

```
import pandas as pd

torque_para_media = []

for indice, linha in df.iterrows():
    if linha['Tipo'] == 'M' and linha['Torque [Nm]'] > 50:
        torque_para_media.append(linha['Torque [Nm]'])

if torque_para_media:
    media_torque = sum(torque_para_media) / len(torque_para_media)
    print("A média do torque para tipo 'M' e torque maior que 50 é:", media_torque)
else:
    print("Não há linhas que correspondam aos critérios especificados.")
```

```

import pandas as pd

torque_para_media = []

total_linhas = len(df)
indice = 0

while indice < total_linhas:
    linha = df.iloc[indice]
    if linha['Tipo'] == 'M' and linha['Torque [Nm]'] > 50:
        torque_para_media.append(linha['Torque [Nm]'])
        indice += 1

if torque_para_media:
    media_torque = sum(torque_para_media) /
len(torque_para_media)
    print("A média do torque para tipo 'M' e torque maior que
50 é:", media_torque)
else:
    print("Não há linhas que correspondam aos critérios
especificados.")

```

Exemplo 12

Calcular a média do torque quando tipo é igual a M e o torque é maior do que 50 usando while

***“O que sabemos é uma gota; o
que ignoramos é um oceano.”
(Issac Newton)***

Referências

- **ASCENCIO, A. F. G, CAMPOS, E. A. V. Fundamentos da Programação de Computadores: algoritmos, Pascal, C/C++ e Java, 2ª Edição, São Paulo: Pearson 2007.**
- **FURGERI, Sérgio. Introdução à Programação em Python. São Paulo: Editora Senac, 2021.**
- **MENEZES, Nilo. Introdução à Programação em Python. São Paulo: Novatec, 2019**
- **SALVETTI, Dirceu Douglas; BARBOSA, Lisbete Madson. Algoritmos. São Paulo: Pearson, 2004.**