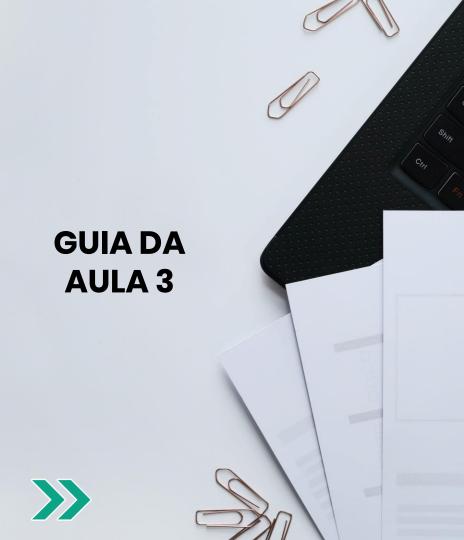


# Profissão: Analista de dados





**ARQUIVOS E FUNÇÕES** 







# Aprenda funções

- Motivação
- Definição

- Retorno
- Parâmetros
- Revisitando a motivação



Acompanhe aqui os temas que serão tratados na videoaula





# 1. Motivação

Você trabalha na bolsa de valores e precisa simular o retorno de um investimento para diversos cenários:





## 1. Motivação

Como podemos fazer para reaproveitar o código e evitar repetições?





# 2. Definição

Um bloco de código que só executado quando chamado:

```
def <nome>(<param 1>, <param 2>, ...): bloco de código
           return <valor de retorno>
         var = <nome da funcao>(<param 1>, <param 2>, ...)
In [ ]:
         def imprime (mensagem: str):
           print (mensagem)
In [ ]:
         texto = 'Fala pessoal, meu nome é André Perez!'
In [ ]:
         imprime (mensagem='Fala pessoal, meu nome é André Perez!')
```





#### 3. Retorno

Toda função retorna pelo menos um valor, se não especificado, retorna o valor nulo.

```
In []: def maiusculo(texto: str) -> str:
    text_maiusculo = texto.upper()
    return text_maiusculo

In []: nome = 'André Perez'
    print(nome)

    nome_maiusculo = maiusculo(texto=nome)
    print(nome_maiusculo)
```





#### 3. Retorno

```
In []:
    def extrair_usuario_email_provedor(email: str) -> (str,
        str): email_separado = email.split(sep='@')
        usuario = email_separado[0]
        provedor = email_separado[1]
        return usuario, provedor

In []:
    email = 'andre.perez@gmail.com'
    usuario, provedor = extrair_usuario_email_provedor(email=email)
        print(usuario)
        print(provedor)
```





#### 4. Parâmetros

Parâmetros são os valores que a passamos na chamada da função.

# Função sem parâmetro:

```
In [ ]:
         def pi() -> float:
           return 3.14159265359
In [ ]:
         pi = pi()
         print(pi)
In [ ]:
         def imprime pi() -> None:
           print(3.14159265359)
           return None
In [ ]:
         imprime pi()
```





#### 4. Parâmetros

# Função com parâmetro:

```
In [ ]:
         def escreve arquivo csv(nome: str, cabecalho: str, conteudos: list) -> bool:
           try:
             with open(file=nome, mode='w', encoding='utf8') as
               fp: linha = cabecalho + '\n'
               fp.write(linha)
               for conteudo in conteudos:
                 linha = str(conteudo) + '\n'
                 fp.write(linha)
           except Exception as exc:
             print(exc)
             return False
           return True
```





#### 4. Parâmetros

```
In []:
    nome = 'idades-funcao-erro.csv'
    cabecalho = 'idade'
    # conteudos = [30, 33, 35, 30, 59, 35, 36, 39, 41, 43]
    conteudos = 10

    escreveu_com_sucesso = escreve_arquivo_csv(
        nome=nome,
        cabecalho=cabecalho,
        conteudos=conteudos
    )
    print(escreveu_com_sucesso)
```





### 5. Revisitando a motivação

```
In [ ]:
         def juros compostos anual(
             valor inicial: float,
             taxa juros anual: float,
             anos: int.
         ) -> float:
           valor final = valor inicial
           for ano in range(1, anos+1):
             valor final = valor final * (1 + taxa juros anual)
           valor final = round(valor final, 2)
           print(f'Para um valor inicial de R$ {valor inicial} ' +
                 f'e uma taxa de juros anual de {taxa juros anual}, ' +
                 f'em {anos} anos você terá R$ {valor final}'
           return valor final
         valor inicial, taxa juros anual, anos = 1000.00, 0.05,
         10 valor final = juros compostos anual (
             valor inicial=valor inicial,
             taxa juros anual=taxa juros anual,
             anos=anos
```





# 5. Revisitando a motivação

