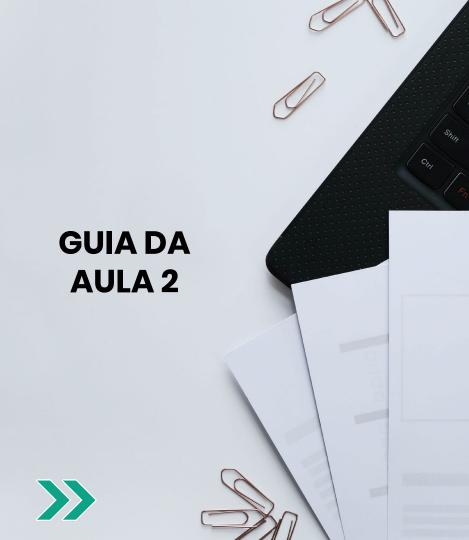


# Python para análise de dados





### PROGRAMAÇÃO FUNCIONAL







## Domine a função map

- Definição
- Funções de alta ordem



Acompanhe aqui os temas que serão tratados na videogula





#### 1. Definição

Aplica uma função em todos os elementos de uma coleção (list, dict etc.) e retorna todos os elementos transformados.

```
variavel = map (função, coleção)

In []: numeros = [1, 2, 3]
    numeros_ao_cubo = map(lambda num: num ** 3, numeros)
    print(list(numeros_ao_cubo))
```





#### 2. Funções de alta ordem

**Exemplo**: Função lambda para extrair provedor de e-mail (1 parâmetro).

```
In [ ]:
    emails = [
        'andre.perez@gmail.com',
        'andre.perez@live.com',
        'andre.perez@yahoo.com'
]
    extrair_provedor_email = lambda email: email.split(sep='@')[-1]

In [ ]:
    provedores = []
    for email in emails:
        provedor = extrair_provedor_email(email)
        provedores.append(provedor)

    print(provedores)
```





#### 2. Funções de alta ordem

```
In []:    provedores = map(extrair_provedor_email, emails)
print(provedores)

In []:    provedores = list(map(extrair_provedor_email, emails))
    print(provedores)

In []:    provedores = map(lambda email: email.split(sep='@')[-1], emails)
    ...
    print(list(provedores))
```





#### 2. Funções de alta ordem

**Exemplo**: Investimento (Mais de 1 parâmetros).

```
In []:
    anos = [10, 10, 10]
    taxas_juros = [0.05, 0.10, 0.15]
    valores_iniciais = [1000, 1000, 1000]

    def retorno(valor_inicial: float, taxa_juros: float, anos: int) ->
        float: valor_final = valor_inicial
        for ano in range(1,anos+1):
            valor_final = valor_final * (1+taxa_juros)
        return round(valor_final, 2)

    cenarios = list(map(retorno, valores_iniciais, taxas_juros, anos)) print(cenarios)
```

