

1. Análise de Transações Financeiras

(Com compreensão de listas)

python



```
def calcular_total_transacoes_positivas(transacoes):
    transacoes_positivas = [valor for valor in transacoes if valor > 0]
    total_transacoes_positivas = sum(transacoes_positivas)
    quantidade_transacoes_positivas = len(transacoes_positivas)
    return total_transacoes_positivas, quantidade_transacoes_positivas

def calcular_total_transacoes_negativas(transacoes):
    transacoes_negativas = [valor for valor in transacoes if valor < 0]
    total_transacoes_negativas = sum(transacoes_negativas)
    return total_transacoes_negativas

# Lista de transações financeiras de um cliente
transacoes = [100, -50, 200, -20, 150, -30, 180]

# Chamando as funções para calcular os totais
total_positivas, quantidade_positivas = calcular_total_transacoes_positivas(transacoes)
total_negativas = calcular_total_transacoes_negativas(transacoes)

# Impressão dos resultados
print("Total das transações positivas:", total_positivas)
print("Quantidade de transações positivas:", quantidade_positivas)
print("Total das transações negativas:", total_negativas)
```

2. List Comprehension – Quadrados de 1 a 5

python



```
quadrados = [x**2 for x in range(1, 6)]
print(quadrados)
# Saída: [1, 4, 9, 16, 25]
```

3. Quadrados de 1 a 5 – Forma Tradicional

python



```
quadrados = []
for x in range(1, 6):
    quadrados.append(x**2)
print(quadrados)
# Saída: [1, 4, 9, 16, 25]
```

4. List Comprehension – Números Pares de 0 a 9

python



```
numeros_pares = [x for x in range(10) if x % 2 == 0]
print(numeros_pares)
# Saída: [0, 2, 4, 6, 8]
```

5. Números Pares – Forma Tradicional

python



```
numeros_pares = []
for x in range(10):
    if x % 2 == 0:
        numeros_pares.append(x)
print(numeros_pares)
# Saída: [0, 2, 4, 6, 8]
```

6. List Comprehension – Palavras em Maiúsculas

python



```
palavras = ['python', 'é', 'incrível']
maiusculas = [palavra.upper() for palavra in palavras]
print(maiusculas)
# Saída: ['PYTHON', 'É', 'INCRÍVEL']
```

7. Palavras em Maiúsculas – Forma Tradicional

python



```
palavras = ['python', 'é', 'incrível']
maiusculas = []
for palavra in palavras:
    maiusculas.append(palavra.upper())
print(maiusculas)
# Saída: ['PYTHON', 'É', 'INCRÍVEL']
```

8. List Comprehension – Expressão Complexa (Pares de Soma Par)

python



```
pares_soma = [(x, y, x + y) for x in range(2) for y in range(2, 4) if (x + y) % 2 == 0]
print(pares_soma)
# Saída: [(0, 2, 2), (1, 3, 4)]
```

9. Expressão Complexa – Forma Tradicional

python



```
pares_soma = []
for x in range(2):
    for y in range(2, 4):
        if (x + y) % 2 == 0:
            pares_soma.append((x, y, x + y))
print(pares_soma)
# Saída: [(0, 2, 2), (1, 3, 4)]
```

10. Compreensão de Dicionário – Quadrados

python



```
quadrados_dict = {x: x**2 for x in range(1, 6)}
print(quadrados_dict)
# Saída: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

11. Dicionário de Quadrados – Forma Tradicional

python



```
quadrados_dict = {}
for x in range(1, 6):
    quadrados_dict[x] = x**2
print(quadrados_dict)
# Saída: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

Atividades Propostas (Exercícios)

12. Dicionário dos Quadrados de 1 a 10 (Compreensão de Dicionário)

python



```
quadrados = {x: x**2 for x in range(1, 11)}
print(quadrados)
```

13. Dicionário Par/Ímpar para 1 a 20 (Compreensão de Dicionário)

python



```
par_impar = {x: 'par' if x % 2 == 0 else 'ímpar' for x in range(1, 21)}
print(par_impar)
```

14. Lista do Dobro dos Ímpares de 1 a 10 (Compreensão de Lista)

python



```
dobro_impares = [x * 2 for x in range(1, 11) if x % 2 != 0]
print(dobro_impares)
```

15. Dicionário de Primos de 1 a 10 (Compreensão de Dicionário)

python



```
def eh_primo(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

primos = {x: eh_primo(x) for x in range(1, 11)}
print(primos)
```

16. Lista FizzBuzz para 1 a 20 (Compreensão de Lista)

python



```
fizzbuzz = ['Fizz' if x % 3 == 0 else 'Buzz' if x % 5 == 0 else x for x in range(1, 21)]
print(fizzbuzz)
```

17. Dicionário de Contagem de Letras em uma String

python



```
texto = "exemplo de texto"
contagem = {letra: texto.count(letra) for letra in texto if letra != ' '}
print(contagem)
```

18. Lista de Palavras com Mais de 5 Letras em uma Frase

python



```
frase = "Python é uma linguagem de programação incrível"
palavras_maiores = [palavra for palavra in frase.split() if len(palavra) > 5]
print(palavras_maiores)
```

19. Dicionário de Meses do Ano e Dias

python



```
meses_dias = {  
    "janeiro": 31,  
    "fevereiro": 28,  
    "março": 31,  
    "abril": 30,  
    "maio": 31,  
    "junho": 30,  
    "julho": 31,  
    "agosto": 31,  
    "setembro": 30,  
    "outubro": 31,  
    "novembro": 30,  
    "dezembro": 31  
}  
print(meses_dias)
```

Código Limpo – Exemplo e Refatoração

20. Código Original (Pouco Legível)

python



```
def fatorial(n): if n==0 or n==1:return 1  
else:return n*fatorial(n-1)  
numeros=[]  
for i in range(5): numeros.append(i+1)  
print(numeros)  
if numeros[0]<3:print("menor que 3")  
else:print("maior ou igual a 3")  
fatoriais = []  
for i in numeros:fatoriais.append(fatorial(i))  
print(fatoriais)
```

21. Código Limpo e Legível (Refatorado, com Compreensão de Lista)

python



```
def fatorial(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        return n * fatorial(n - 1)  
  
numeros = [i + 1 for i in range(5)]  
print(numeros)  
  
if numeros[0] < 3:
```

```
        print("menor que 3")
else:
    print("maior ou igual a 3")

fatoriais = [fatorial(i) for i in numeros]
print(fatoriais)
```