

Nome: João Geiger Piza

Repositório no Github: <https://github.com/joaogpizza/Lipai>

Códigos das Videoaulas:

aula01.py:

```
""" Aula 01 - Operadores """

# Operadores Aritméticos
n1 = 10.2
n2 = 3.5
resultado = n1 + n2 + 8.5

print('1 + 1', 1 + 1, type(1+1))
print('1.2 + 1.3', 1.2 + 1.3, type(1.2+1.3))
print('resultado', resultado, type(resultado))
print(3-1)
print(3*2, type(3*2))
print(3/2, type(3/2))
print(3 % 2)
print(10 // 3)
print(10 ** 3)

# Operador de atribuição
x = 10.0
print(x)

# Operadores de comparação
resultado = x > 10
print(resultado, type(resultado))

print('10 == 10', 10 == 10, type(10 == 10))
print('10 != 10', 10 != 10, type(10 != 10))
print('10 > 10', 10 > 10, type(10 > 10))
print('10 >= 10', 10 >= 10, type(10 >= 10))
print('10 < 10', 10 < 10, type(10 < 10))
print('10 <= 10', 10 <= 10, type(10 <= 10))

# condicao = x > 10.0 and resultado < 3.0
# if condicao:
#     pass

# Operadores lógicos
print('True and True', True and True, type(True and True))
```

```
print('True and False', True and False, type(True and False))
print('False and True', False and True, type(False and True))
print('False and False', False and False, type(False and False))

print('True or True', True or True, type(True or True))
print('True or False', True or False, type(True or False))
print('False or True', False or True, type(False or True))
print('False or False', False or False, type(False or False))

condicao = False
print('not condicao', not condicao, type(not condicao))

# Operadores especiais

# is
# == comparar se dois valores são iguais
# is verificar se as variáveis apontam para o mesmo
# objeto em memória

a = 10
b = 10.0
c = b

print(a == b, a == c, b == c)
print(a is b, a is c, b is c)

# in
# str, list, tuple, set, dic (chave)

frase = 'Você é um palavrao!'

print('palavrao' in frase, type('palavrao' in frase))
print('Palavrao' in frase, type('Palavrao' in frase))

numeros = {1, 5, 2, 6}
print(1 in numeros)

pessoa = {
    'nome': 'Maria',
    'idade': 22,
    'email': 'maria@email.com'
}
```

```
print(22 in pessoa)
```

aula02.py:

```
""" Aula 02 - instrução if """

# python
# if condicao:
#     instrucao
#     instrucao
#     instrucao

# C, Java, C#, outras
# if() {
#     instrucao
# instrucao
# }

codigo_cliente = 32
valor_desconto = 23.0
DESCONTO_ESPECIAL = valor_desconto >= 20.0

if DESCONTO_ESPECIAL >= 20.0:
    print('Desconto Especial')
    print(codigo_cliente)
else:
    print('Sem desconto especial')

# nome tem que ter pelo menos 3 caracteres
nome = 'Loe'

print(len(nome), type(len(nome)))

NOME_INVALIDO = len(nome) < 3

if NOME_INVALIDO:
    print('Nome deve ter pelo menos 3 caracteres')
else:
    print('Nome válido')

NOME_VALIDO = len(nome) >= 3
if NOME_VALIDO:
```

```

        print('Nome válido')
else:
    print('Nome deve ter pelo menos 3 caracteres')


if not NOME_INVALIDO:
    print('Nome válido')
else:
    print('Nome deve ter pelo menos 3 caracteres')


# nome tem que ter pelo menos 3 caracteres
# idade tem que ser maior ou igual a 18
# exibir todos os erros no final apenas
nome = 'Lo'
idade = 18
erros = []

NOME_INVALIDO = len(nome) < 3
if NOME_INVALIDO:
    erros.append('Nome deve ter pelo menos 3 caracteres')

IDADE_INVALIDA = idade < 18
if IDADE_INVALIDA:
    erros.append('Idade deve ser maior ou igual a 18')


# False: False, None, 0, 0.0, string vazia '', lista, tuple, set vazio
# True: todo o resto
if len(erros) != 0:
    print(erros)
else:
    print('Dados válidos')


# if elif else

# Verifica se um número é positivo ou negativo
numero = -4

#
# _____ - _____
# -N -4 -3 -2 -1 0 1 2 3 4 N
if numero > 0:

```

```

        print('Maior que zero')
elif numero == 0:
    print('Zero')
else:
    print('Menor que zero')

# calcule a média e verifique se as duas notas
# são validas antes do calculo
n1 = 5.6
n2 = 10

# Cuidado com ifs aninhados
if n1 >= 0 and n1 <= 10:
    if n2 >= 0 and n2 <= 10:
        media = (n1 + n2) / 2
        if media >= 6:
            print('Aprovado')
        elif media >= 4:
            print('Recuperacao')
        else:
            print('Reprovado')
    else:
        print('Notas inválidas')
else:
    print('Notas inválidas')

NOTA1_VALIDA = 0 <= n1 <= 10
NOTA2_VALIDA = 0 <= n2 <= 10

if NOTA1_VALIDA and NOTA2_VALIDA:
    media = (n1 + n2) / 2
    if media >= 6:
        print('Aprovado')
    elif media >= 4:
        print('Recuperacao')
    else:
        print('Reprovado')
else:
    print('Notas inválidas')

```

aula03.py:

```
""" Aula 03 - loop for """

# 1. iteração em coleção de elementos
# 2. Repetir instrução


linguagens = ['C', 'Python', 'Java']

print(linguagens[0])
print(linguagens[1])
print(linguagens[2])

# for valor in colecao:
#     instrucao
#     instrucao
#     instrucao


for linguagem in linguagens:
    print(linguagem.upper())


# Comportamento do operador in é
# diferente com base no contexto
print('Java' in linguagens)

nota1 = 10.0
nota2 = 5.5
nota3 = 8.3

media = (nota1 + nota2 + nota3) / 3
print(media)

notas = [10.0, 5.5, 8.3, 2.5]

soma = 0.0
for nota in notas:
    soma = soma + nota

media = soma / len(notas)
print(media)

# range
```

```
# valores = range(10)
# valores = range(0, 10)
# valores = range(0, 10, 2)
valores = range(9, -1, -2)
print(valores)

for valor in valores:
    print(valor)

for i in range(len(linguagens)):
    print(linguagens[i])

for linguagem in linguagens:
    print(linguagem)
```

aula04.py:

```
""" Aula 04 - while """

# while condicao:
#     instrucao
#     instrucao

contador = 0
while contador <= 10:
    print(contador)
    contador +=1

print('fim')
```

aula05.py:

```
""" Aula 05 - break e continue """

for i in range(10):
    if i == 4:
        break
    print(i)

# Encontrar a posicao de um numero
# em uma lista de inteiros. Caso não
# encontre posicao é igual a -1

busca = 5
numeros = [1, 5, 9, 7, 3, 3, 2, 1, 7]
```

```

posicao = -1

contador = 0
for numero in numeros:
    print('Procurando na posicao:', contador)
    if numero == busca:
        posicao = contador
        break
    contador += 1

print(posicao)

# for i in range(len(numeros)):
#     print('Procurando na posicao:', i)
#     if numeros[i] == busca:
#         posicao = i
#         break

# print(posicao)

# continue
# pular a iteração atual
print('-----')
for numero in numeros:
    if numero == 3:
        continue
    print(numero)

```

Exercícios:

ex01.py:

```

LIMITE_SUPERIOR = 10.0
LIMITE_INFERIOR = 0.0
NUMERO_NOTAS = 3

notas = []

for i in range(NUMERO_NOTAS):
    while(True):
        buffer = float(input(f'Digite a nota {i+1}: '))
        if LIMITE_INFERIOR <= buffer <= LIMITE_SUPERIOR:
            notas.append(buffer)
            break

```

```
        print(f'Nota invalida, tem de ser maior ou igual a\n{LIMITE_INFERIOR} e menor ou igual a {LIMITE_SUPERIOR}')
        continue

media = 0.0
for i in range(NUMERO_NOTAS):
    media += notas[i]

media /= NUMERO_NOTAS
print(f'A média aritmética é: {media}')
```

ex02.py:

```
""" Ex 02 """
MEDIA_APROVACAO = 6.0
MEDIA_RECUPERACAO = 4.0

NOTAS_STRING = input('Digite as notas no formato n1, n2, n3, nm: \n')

NOTAS = NOTAS_STRING.split(',')

media = 0.0
for nota in NOTAS:
    media += float(nota)

media /= len(NOTAS)

if media >= MEDIA_APROVACAO:
    print('aprovado')
elif media >= MEDIA_RECUPERACAO:
    print('recuperação')
else:
    print('reprovado')
```

ex03.py:

```
""" Ex 03 """
CODIGO = input('Digite o código: ')
buffer = True

if len(CODIGO) == 7:
    if CODIGO[0] == 'B' and CODIGO[1] == 'R' and CODIGO[6] == 'X':
        for i in range(2, 6):
```

```

        if '0' <= CODIGO[i] <= '9':
            continue
        else:
            buffer = False
            break
    else:
        buffer = False
else:
    buffer = False

if buffer:
    print('O código é válido')
else:
    print('O código é inválido')

```

ex04.py:

```

""" Ex 04 """

CODIGO = input('Digite o identificador: ')
erros = []
buffer = False

if len(CODIGO) != 7:
    erros.append("O identificador tem mais de 7 dígitos")

if len(CODIGO) >= 2:
    if not (CODIGO[0] == 'B' and CODIGO[1] == 'R'):
        erros.append("O identificador não inicia com a sequência BR")
else:
    erros.append("O identificador não inicia com a sequência BR")

if len(CODIGO) >= 4:
    for i in range(len(CODIGO) - 3):
        substring = CODIGO[i:i+4]
        buffer = '0001' <= substring <= '9999'
        for caractere in substring:
            buffer = buffer and ('0' <= caractere <= '9')
        if buffer:
            break
    if not buffer:
        erros.append('O identificador não apresenta número inteiro entre 0001 e 9999')

```

```
if CODIGO[len(CODIGO)-1] != 'X':
    erros.append('O identificador não finaliza com o caractere X')

if erros:
    print('O identificador é invalido\nErros:')
    for erro in erros:
        print(erro)
else:
    print('O identificador é válido')
```