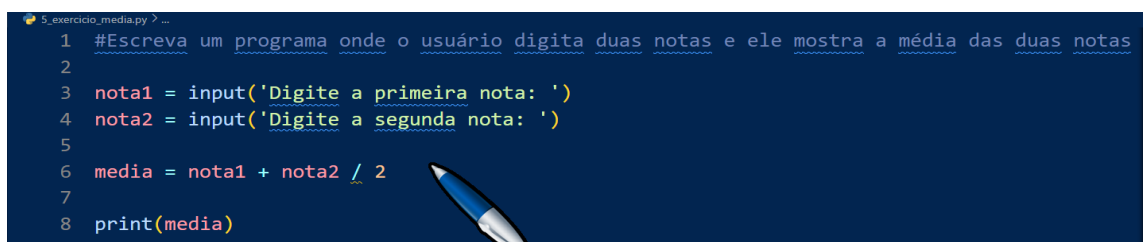
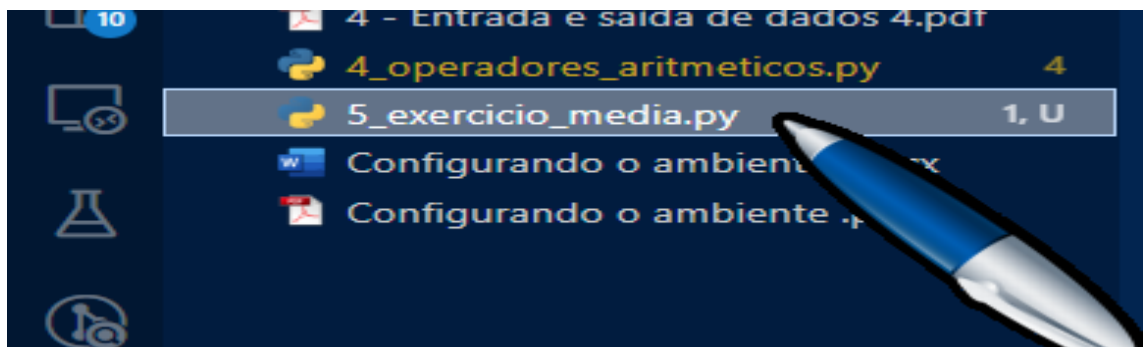
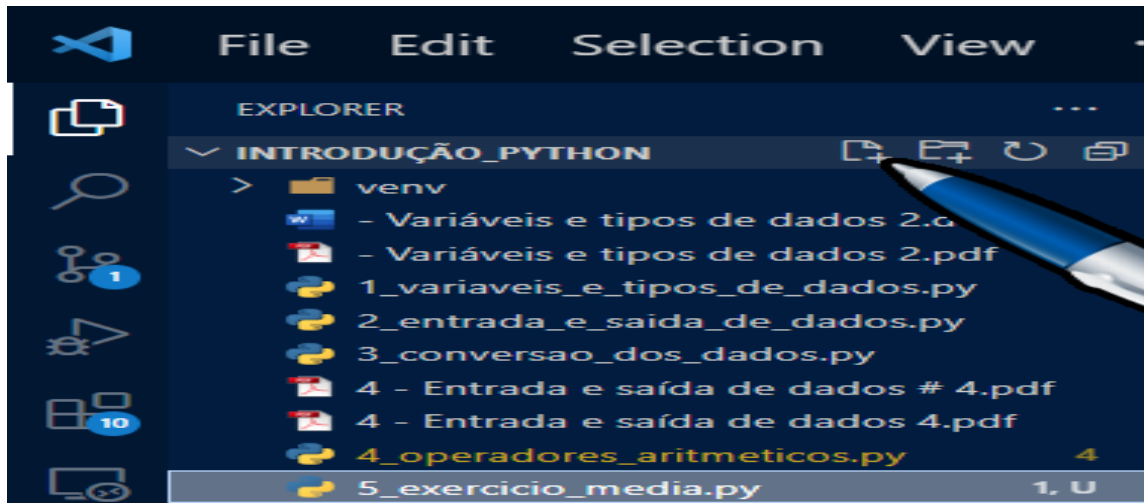


8 - [EX] Exercício média

Criamos uma pasta



Veja que vamos rodar o código e vai dar um erro

Python 5_exercicio_media.py

```
PORTS  AZURE  COMMENTS  OUTPUT  PROBLEMS  54  DEBUG CONSOLE  TERMINAL  SQL CONSOLE  GITLENS

• n/Introdução_Python/venv/Scripts/Activate.ps1
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 5_exercicio_media.py

Digite a primeira nota: 
```

```
• n/Introdução_Python/venv/Scripts/Activate.ps1
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 5_exercicio_media.py

Digite a primeira nota: 10
Digite a segunda nota: 
```

Apresentou erro não é suportado a divisão entre string e inteiro

```
Digite a primeira nota: 10
Digite a segunda nota: 10
Traceback (most recent call last):
  File "C:\Users\Irene\Documents\aulaPython\Introdução_Python\5_exercicio_media.py", line 6, in <module>
    media = nota1 + nota2 / 2
              ~~~~~^~~~~
TypeError: unsupported operand type(s) for /: 'str' and 'int'
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

divisão entre string e inteiro

```
media = nota1 + nota2 / 2
```

Se digitarmos

```
8 #print(media)
9 print(type(nota1))
10
```

Rodar o código

```
python 5_exercicio_media.py

Digite a primeira nota: 
```

Erro

```
~~~~~^~~~~
TypeError: unsupported operand type(s) for /: 'str' and 'int'
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

Roda o código

```
Digite a primeira nota: 10
Digite a segunda nota: 10
```

python 5_exercicio_media.py

Erro

```
TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

Não podemos dividir string por numero inteiro

Podemos converter em float

```
5_exercicio_media.py > ...
1 #Escreva um programa onde o usuário digita duas notas e ele mostra a média das duas notas
2
3 nota1 = input('Digite a primeira nota: ')
4 nota2 = input('Digite a segunda nota: ')
5
6 media = float(nota1) + float(nota2) / 2
7
8 #print(media)
9 #print(type(nota1))
10
11 print(media)
```

Rodando código vai dar resposta errada mas não vai dar erro

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 5_exercicio_media.py
Digite a primeira nota: 
```

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 5_exercicio_media.py
Digite a primeira nota: 10
Digite a segunda nota: 10
15.0
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

Outra forma de fazer

```
1 #Escreva um programa onde o usuário digita duas notas e ele
2
3 nota1 = float(input('Digite a primeira nota: '))
4 nota2 = float(input('Digite a segunda nota: '))
5
6 media = nota1 + nota2 / 2
7
8 #media = float(nota1) + float(nota2) / 2
9
10 #print(media)
11 #print(type(nota1))
12
13 print(media)
```

Roda o código

```
python 5_exercicio_media.py
Digite a primeira nota: 10
Digite a segunda nota: 10
15.0
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

Outra forma de fazer não recomendada

```
1 #Escreva um programa onde o usuário digita duas notas e ele most
2
3 nota1 = input('Digite a primeira nota: ')
4 nota2 = input('Digite a segunda nota: ')
5 nota1 =float(nota1)
6 nota2 =float(nota2)
7
8 #nota1 = float (input('Digite a primeira nota: '))
9 #nota2 = float (input('Digite a segunda nota: '))
10 media = nota1 + nota2 / 2
11
12 #media = float(nota1) + float(nota2) / 2
13
14 #print(media)
15 #print(type(nota1))
16
17 print(media)
18
```

Roda o código

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 5_exercicio_media.py
Digite a primeira nota: 10
Digite a segunda nota: 10
15.0
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

O correto é desta forma

```
3 nota1 = float (input('Digite a primeira nota: '))
4 nota2 = float (input('Digite a segunda nota: '))
5
6 media = nota1 + nota2 / 2
7 print(media)
8
9
10
11
```

Devemos colocar entre parênteses para forçar

```

13 nota1 = float (input('Digite a primeira nota: '))
14 nota2 = float (input('Digite a segunda nota: '))
15
16 media = (nota1 + nota2) / 2
17 print(media)
18
19
20
21

```

Rodar o código

```

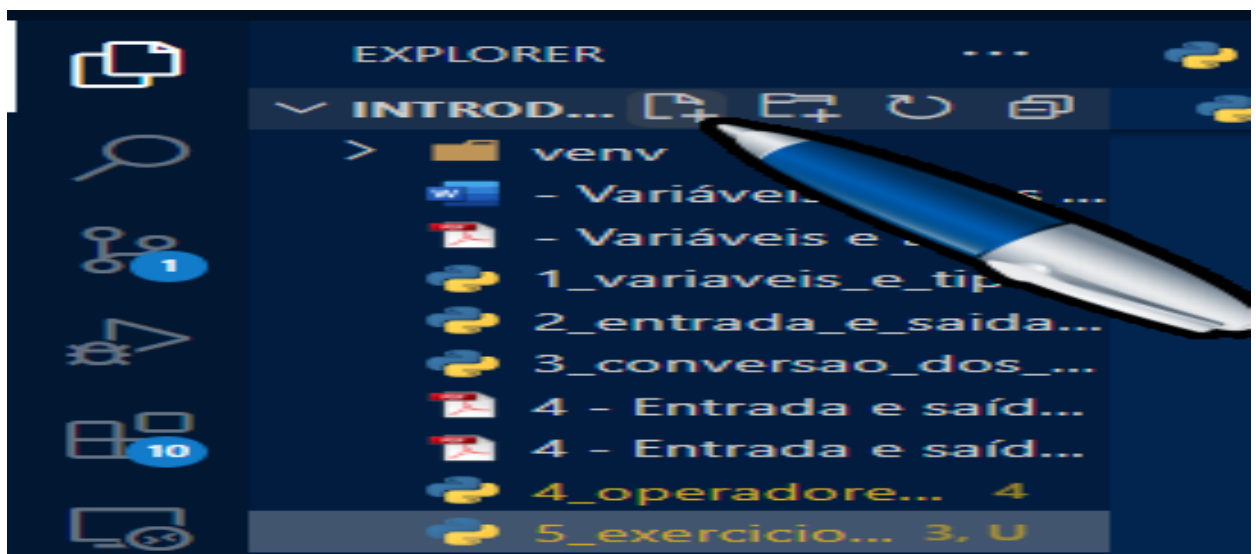
• (venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 5_exercicio_media.py

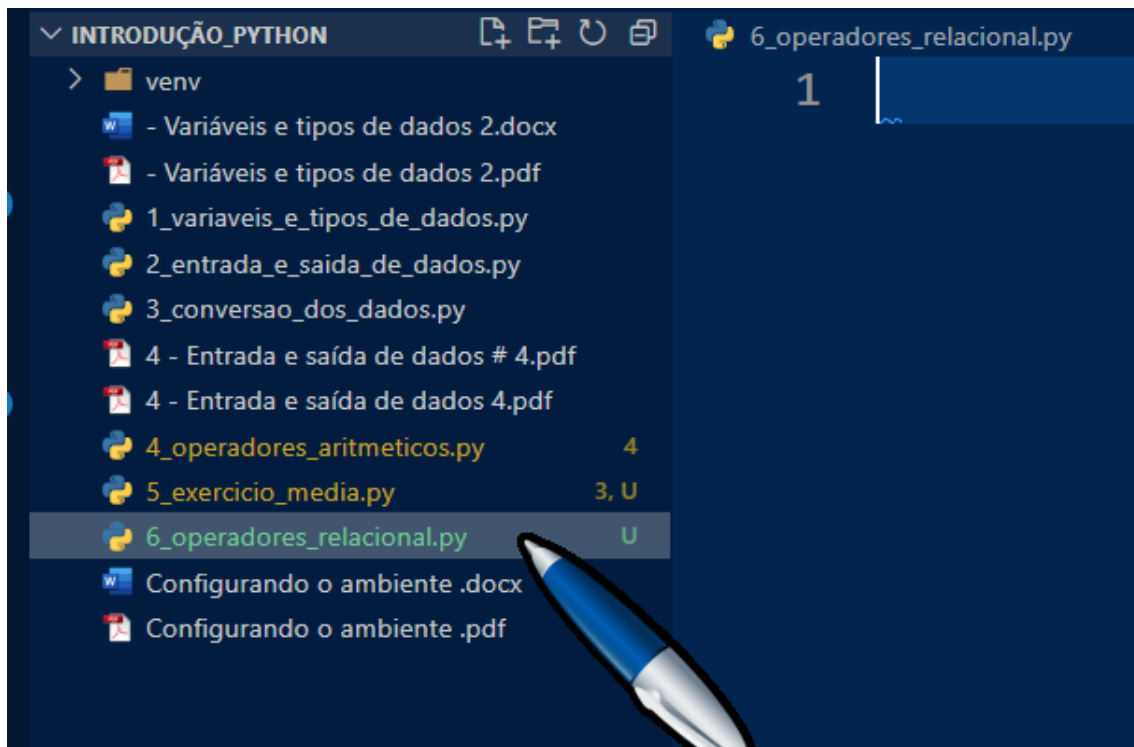
Digite a primeira nota: 10
Digite a segunda nota: 10
10.0
• (venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

```

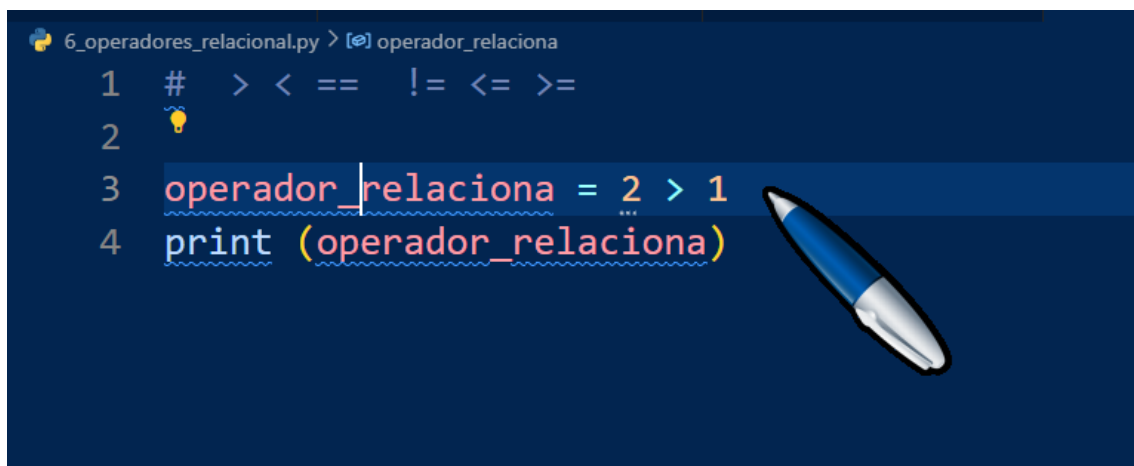
9 - Operadores Relacionais

Criar pasta

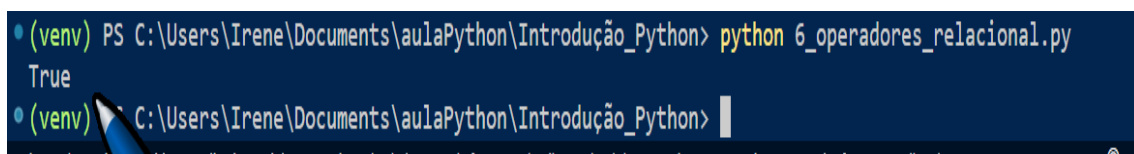




Relação entre dois valores verdade ou falso



Roda o código resultado true verdadeiro



```
6_operadores_relacional.py > ...
1  # > < == != <= >=
2
3  #operador_relacional = 2 > 1
4  operador_relacional = 1 > 1
5  print (operador_relacional)
```

Resultado false

```
True
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 6_operadores_relacional.py
False
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  operador_relacional = 2 < 1
6  print (operador_relacional)
```

```
False
python 6_operadores_relacional.py
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
6_operadores_relacional.py > ...
1  #  > < ==  != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  operador_relacional = 1 < 2
7  print (operador_relacional)
```

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 6_operadores_relacional.py
True
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```


Comparação

```
6_operadores_relacional.py > ...
1  #  > < ==  != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  #operador_relacional = 1 < 2
7  operador_relacional = 2 == 2
8  print (operador_relacional)
```

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 6_operadores_relacional.py
True
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```




```
6_operadores_relacional.py > ...
1  #  > < ==  != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  #operador_relacional = 1 < 2
7  #operador_relacional = 2 == 2
8  operador_relacional = 2 == 3
9  print (operador_relacional)
```



```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
False
python 6_operadores_relacional.py
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
6_operadores_relacional.py > ...
1  #  > < ==  != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  #operador_relacional = 1 < 2
7  #operador_relacional = 2 == 2
8  #operador_relacional = 2 == 3
9  operador_relacional = 2 == 2.0
10 print (operador_relacional)
```



```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
True
python 6_operadores_relacional.py
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

6_operadores_relacional.py > ...

```
1  # > < == != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  #operador_relacional = 1 < 2
7  #operador_relacional = 2 == 2
8  #operador_relacional = 2 == 3
9  #operador_relacional = 2 == 2.0
10 operador_relacional = 2 != 3
11 print (operador_relacional)
```

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

python 6_operadores_relacional.py

True

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

6_operadores_relacional.py > ...

```
1  #  > < == != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  #operador_relacional = 1 < 2
7  #operador_relacional = 2 == 2
8  #operador_relacional = 2 == 3
9  #operador_relacional = 2 == 2.0
10 operador_relacional = 2 != 2
11 print (operador_relacional)
```

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

python 6_operadores_relacional.py

False

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

6_operadores_relacional.py > ...

```
1  # > < == != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  #operador_relacional = 1 < 2
7  #operador_relacional = 2 == 2
8  #operador_relacional = 2 == 3
9  #operador_relacional = 2 == 2.0
10 #operador_relacional = 2 != 2
11 operador_relacional = 2 >= 2
12 print (operador_relacional)
```

python 6_operadores_relacional.py

True

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

6_operadores_relacional.py > ...

```
1  #  > < == != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  #operador_relacional = 1 < 2
7  #operador_relacional = 2 == 2
8  #operador_relacional = 2 == 3
9  #operador_relacional = 2 == 2.0
10 #operador_relacional = 2 != 2
11 #operador_relacional = 2 >= 2
12 operador_relacional = 2 >= 4
13 print (operador_relacional)
```

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

python 6_operadores_relacional.py

False

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

```
6_operadores_relacional.py > ...  
1  #  > < == != <= >=  
2  
3  #operador_relacional = 2 > 1  
4  #operador_relacional = 1 > 1  
5  #operador_relacional = 2 < 1  
6  #operador_relacional = 1 < 2  
7  #operador_relacional = 2 == 2  
8  #operador_relacional = 2 == 3  
9  #operador_relacional = 2 == 2.0  
10 #operador_relacional = 2 != 2  
11 #operador_relacional = 2 >= 2  
12 #operador_relacional = 2 >= 4  
13 operador_relacional = 2 >= 1  
14 print (operador_relacional)
```

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>  
python 6_operadores_relacional.py  
True  
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
6_operadores_relacional.py > ...
1  #  > < == != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  #operador_relacional = 1 < 2
7  #operador_relacional = 2 == 2
8  #operador_relacional = 2 == 3
9  #operador_relacional = 2 == 2.0
10 #operador_relacional = 2 != 2
11 #operador_relacional = 2 >= 2
12 #operador_relacional = 2 >= 4
13 #operador_relacional = 2 >= 1
14 operador_relacional = 2 <= 1
15 print (operador_relacional)
```

```
True
• (venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 6_operadores_relacional.py
False
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

6_operadores_relacional.py > ...

```
1  #  > < == != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  #operador_relacional = 1 < 2
7  #operador_relacional = 2 == 2
8  #operador_relacional = 2 == 3
9  #operador_relacional = 2 == 2.0
10 #operador_relacional = 2 != 2
11 #operador_relacional = 2 >= 2
12 #operador_relacional = 2 >= 4
13 #operador_relacional = 2 >= 1
14 #operador_relacional = 2 <= 1
15 operador_relacional = 2 <= 2
16 print (operador_relacional)
```

True
python 6_operadores_relacional.py
(venv) PS C:\Users\Irene\Documents\aula_python\Introdução_Python>


```
6_operadores_relacional.py > ...
1  #  > < == != <= >=
2
3  #operador_relacional = 2 > 1
4  #operador_relacional = 1 > 1
5  #operador_relacional = 2 < 1
6  #operador_relacional = 1 < 2
7  #operador_relacional = 2 == 2
8  #operador_relacional = 2 == 3
9  #operador_relacional = 2 == 2.0
10 #operador_relacional = 2 != 2
11 #operador_relacional = 2 >= 2
12 #operador_relacional = 2 >= 4
13 #operador_relacional = 2 >= 1
14 #operador_relacional = 2 <= 1
15 #operador_relacional = 2 <= 2
16 operador_relacional = 2 <= 3
17 print (operador_relacional)
```

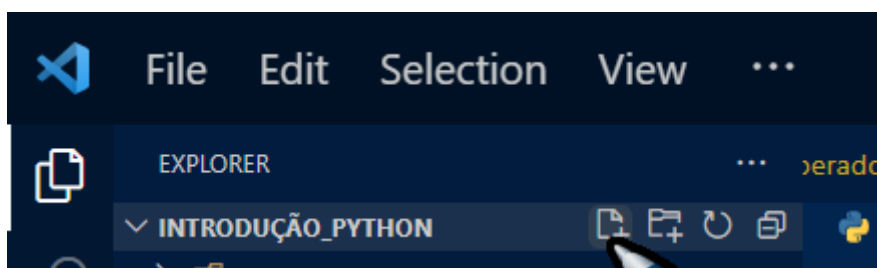
python 6_operadores_relacional.py

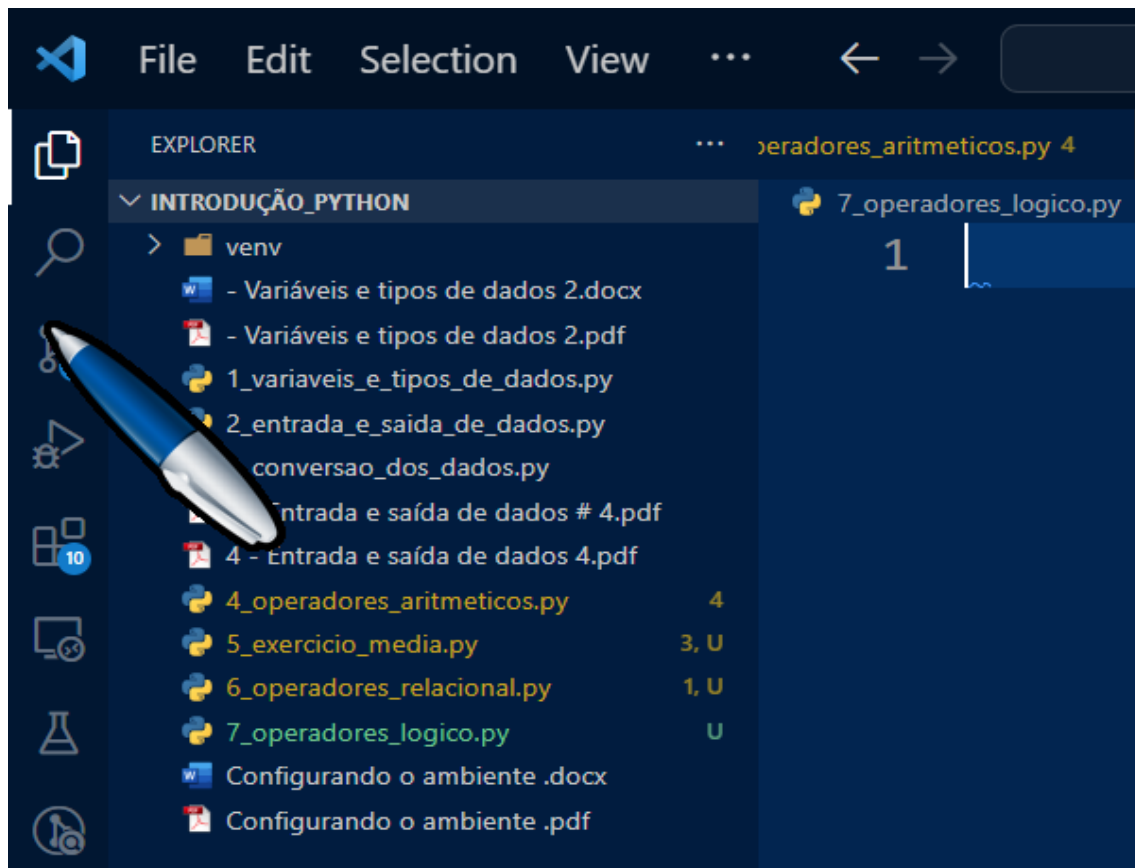
True

(venv) PS C:\Users\I... \Documents\aulaPython\Introdução_Python>

10 - Operadores lógicos

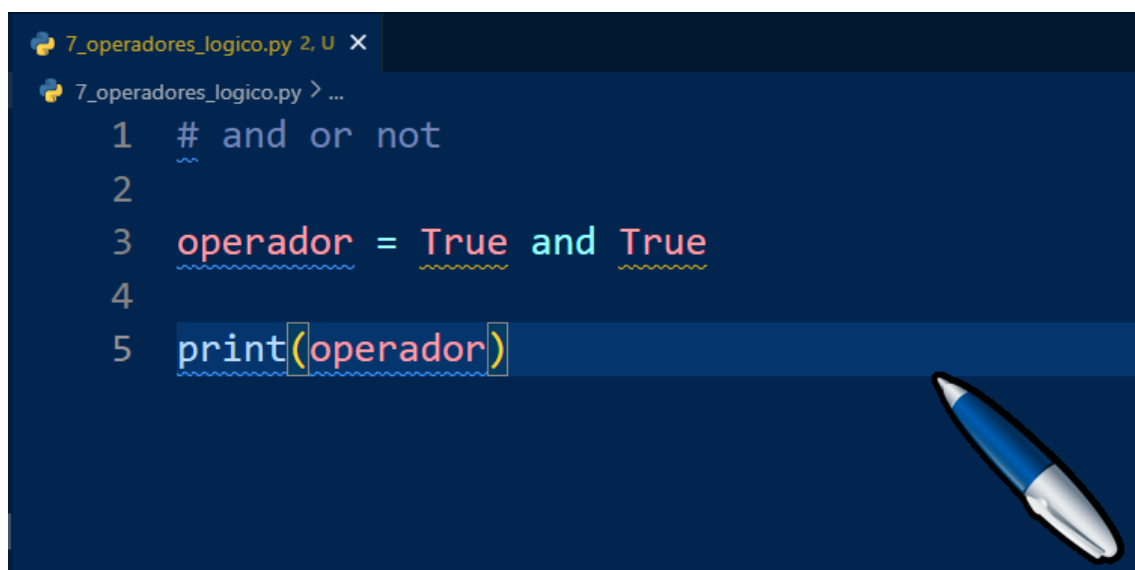
Criar pasta





Sempre vai retornar verdadeiro ou falso ele vai fazer comparação entre verdadeiro e falso

O and vai retornar se os que estiver dos dois lados for verdadeiro ou seja ele retorna verdadeiro



Vamos rodar o código

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 7_operadores_logico.py
True
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
7_operadores_logico.py 2, U x
7_operadores_logico.py > ...
1  # and or not
2
3  #operator = True and True
4  operator = False and True
5
6  print(operator)
```

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 7_operadores_logico.py
False
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

O and só retorna true quando os dois lados for verdadeiro se um dos lados for false vai retornar false

```
7_operadores_logico.py 3, U X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  operador = False and False
6
7  print(operador)
```

Lembre sempre de rodar o código

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 7_operadores_logico.py
False
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

Or se um dos lados for verdadeiro retorna true

```
7_operadores_logico.py 3, U X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  operador = False or False
7
8  print(operador)
```

```
python 7_operadores_logico.py
False
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
7_operadores_logico.py 2, U x
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  operador = False or True
8
9  print(operador)
```

```
python 7_operadores_logico.py
True
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

O not inverte se algo for verdadeiro transforma em falso e vice-versa

```
7_operadores_logico.py 1, U X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  #operador = False or True
8  operador = not False
9
10 print(operador)
```

```
python 7_operadores_logico.py
True
(venv) C:\Users\Irene\Documents\aulaPython\Introdução_Python> |
```


```
7_operadores_logico.py 1, U X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  #operador = False or True
8  #operador = not False
9  operador = not True
10 print(operador)
```

```
python 7_operadores_logico.py
False
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```


```
7_operadores_logico.py 2, U x
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  #operador = False or True
8  #operador = not False
9  #operador = not True
10 operador = 5 == 5 and 3 > 7
11 print(operador)
```

```
python 7_operadores_logico.py
False
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
7_operadores_logico.py 3.0 X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  #operador = False or True
8  #operador = not False
9  #operador = not True
10 operador = 5 == 5 and 3 > 7
11 operador = 5 == 5 and 3 > 2
12 print(operador)
```



```
True
python 7_operadores_logico.py
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> |
```




```
7_operadores_logico.py 2, U X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  #operador = False or True
8  #operador = not False
9  #operador = not True
10 #operador = 5 == 5 and 3 > 7
11 #operador = 5 == 5 and 3 > 2
12 operador = 5 == 5 or 3 > 2
13 print(operador)
```

```
python 7_operadores_logico.py
True
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
7_operadores_logico.py 1, U X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  #operador = False or True
8  #operador = not False
9  #operador = not True
10 #operador = 5 == 5 and 3 > 7
11 #operador = 5 == 5 and 3 > 2
12 #operador = 5 == 5 or 3 > 2
13 operador = 5 == 7 or 3 > 2
14 print(operador)
```

```
python 7_operadores_logico.py
True
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
7_operadores_logico.py 2, U X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  #operador = False or True
8  #operador = not False
9  #operador = not True
10 #operador = 5 == 5 and 3 > 7
11 #operador = 5 == 5 and 3 > 2
12 #operador = 5 == 5 or 3 > 2
13 operador = 5 == 7 or 3 > 3
14 print(operador)
```

```
python 7_operadores_logico.py
False
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
7_operadores_logico.py 2, U X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  #operador = False or True
8  #operador = not False
9  #operador = not True
10 #operador = 5 == 5 and 3 > 7
11 #operador = 5 == 5 and 3 > 2
12 #operador = 5 == 5 or 3 > 2
13 #operador = 5 == 7 or 3 > 3
14 operador = not (5 == 7 or 3 > 3)
15 print(operador)
```

```
python 7_operadores_logico.py
True
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
7_operadores_logico.py 1, U X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  #operador = False or True
8  #operador = not False
9  #operador = not True
10 #operador = 5 == 5 and 3 > 7
11 #operador = 5 == 5 and 3 > 2
12 #operador = 5 == 5 or 3 > 2
13 #operador = 5 == 7 or 3 > 3
14 #operador = not (5 == 7 or 3 > 3)
15 operador = not (5 == 7 or 3 > 2)
16 print(operador)
```

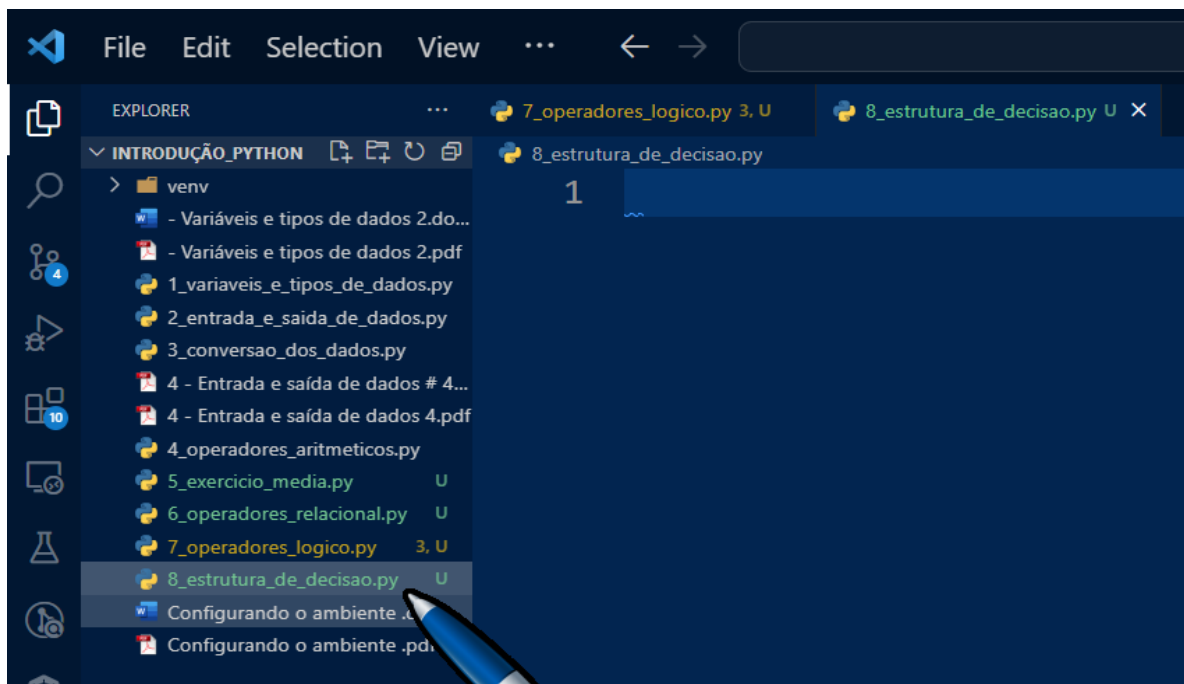
```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 7_operadores_logico.py
False
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
7_operadores_logico.py 3, U X
7_operadores_logico.py > ...
1  # and or not
2
3  #operador = True and True
4  #operador = False and True
5  #operador = False and False
6  #operador = False or False
7  #operador = False or True
8  #operador = not False
9  #operador = not True
10 #operador = 5 == 5 and 3 > 7
11 #operador = 5 == 5 and 3 > 2
12 #operador = 5 == 5 or 3 > 2
13 #operador = 5 == 7 or 3 > 3
14 #operador = not (5 == 7 or 3 > 3)
15 #operador = not (5 == 7 or 3 > 2)
16 operador = not ( (5 == 7 or 3 > 2) and ( 2 == 2 or 5 < 5 ) )
17 print(operador)
```

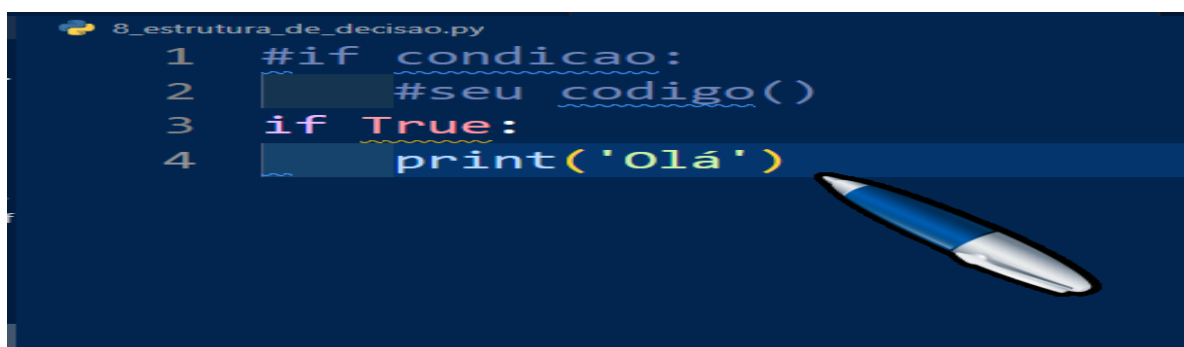
```
python 7_operadores_logico.py
False
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

11 - Estrutura de decisão

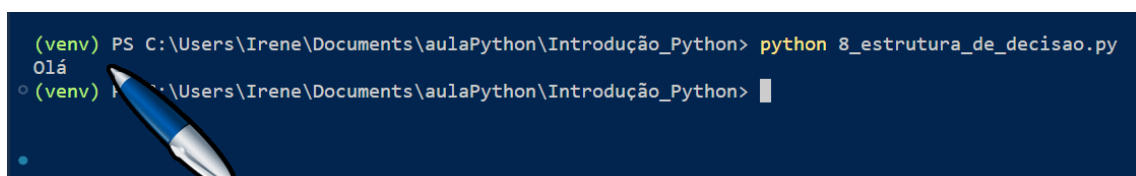
Abrir nova pasta



Podemos executar um bloco de código somente se uma estrutura for verdadeira



Roda o código



Para o print pertencer ao if sempre mantem sempre uma tubulação a frente do contrario não pertencerá

```
8_estrutura_de_decisao.py
1  #if condicao:
2  #seu codigo()
3  #if True:
4  #print('Olá')
5  if False:
6  print('Olá')
```

Quando colocamos false não será executado

```
8_estrutura_de_decisao.py
1  #if condicao:
2  #seu codigo()
3  #if True:
4  #print('Olá')
5  if False:
6  print('Olá')
```

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 8_estrutura_de_decisao.py
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
7_operadores_logico.py 3. U | 8_estrutura_de_decisao.py U X
8_estrutura_de_decisao.py
1  #if condicao:
2  #seu codigo()
3  #if True:
4  #print('Olá')
5  #if False:
6  #print('Olá')
7  if 5 > 3:
8  print('Olá')
```

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 8_estrutura_de_decisao.py
Olá
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

Condição falsa não mostra nada

```
8_estrutura_de_decisao.py
1  #if condicao:
2  #seu codigo()
3  #if True:
4  #print('Olá')
5  #if False:
6  #print('Olá')
7  #if 5 > 3:
8  #print('Olá')
9  if 5 > 7:
10 print('Olá')
```

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 8_estrutura_de_decisao.py

```
8_estrutura_de_decisao.py
1  #if condicao:
2  #seu_codigo()
3  #if True:
4  #print('Olá')
5  #if False:
6  #print('Olá')
7  #if 5 > 3:
8  #print('Olá')
9  #if 5 > 7:
10 #print('Olá')
11
12 if 5 > 7:
13     print('Olá')
14     print('tudo bem')
15
16 print('Até mais!')
```

```
python 8_estrutura_de_decisao.py
Até mais
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
12 '''if 5 > 7:
13     print('Olá')
14     print('tudo bem')
15
16 print('Até mais!')'''
17
18 if 5 > 4:
19     print('Olá')
20     print('tudo bem')
21
22 print('Até mais!')
```

```
python 8_estrutura_de_decisao.py
Olá
tudo bem
Até mais!
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

Para que and seja verdadeiro os dois lados tem que ser verdadeiros

```
18 '''if 5 > 4:
19     print('Olá')
20     print('tudo bem')
21
22 print('Até mais!')'''
23
24 if 5 > 4 and 6 == 7:
25     print('Olá')
26     print('tudo bem')
27
28 print('Até mais!')
```

```
Até mais!
• (venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
python 8 Estrutura de Decisão.py
Até mais!
• (venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

Coloquei condição verdadeira vai mostrar as três linhas

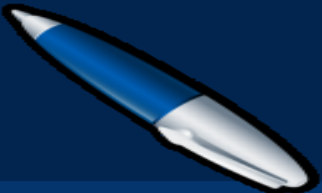
```
'''if 5 > 4 and 6 == 7:
    print('Olá')
    print('tudo bem')
print('Até mais!')'''

if 5 > 4 and 6 == 6:
    print('Olá')
    print('tudo bem')
print('Até mais!')
```

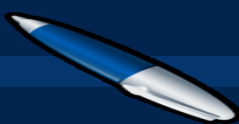
```
• (venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 8 Estrutura de Decisão.py
Olá
tudo bem
Até mais!
• (venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```


Else significa senão


```
35
36 # if else elif
37
38 if 5 == 5:
39     print('Olá')
40
```



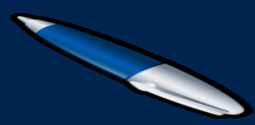
```
36 # if else elif
37
38 media = 8
39
40 if media >= 8:
41     print('APROVADO')
42 else:
43     print('REPROVADO')
44
```




```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 8_estrutura_de_decisao.py
APROVADO
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```



```
45 # if else elif
46
47 media = 4
48
49 if media >= 6:
50     print('APROVADO')
51 else:
52     print('REPROVADO')
```



```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python> python 8_estrutura_de_decisao.py
REPROVADO
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```



Obs: o else é uma opção o else nunca será executado se uma condição se não for verdadeira

ELI F= SE NÃO SE

```
53  
54 # if else elif  
55  
56 media = 4  
57  
58 if media >= 6:  
59     print('APROVADO')  
60 elif media >= 4:  
61     print('RECUPERAÇÃO')
```

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
python 8_estrutura_de_decisao.py
```

• RECUPERAÇÃO

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
# if else elif  
  
media = 2  
  
if media >= 6:  
    print('APROVADO')  
elif media >= 4:  
    print('RECUPERAÇÃO')  
else:  
    print('REPROVADO')
```

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
python 8_estrutura_de_decisao.py
```

REPROVADO

```
(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>
```

```
74 # if else elif
75
76 media = 2
77
78 if media >= 6:
79     print('APROVADO')
80 elif media >= 4:
81     print('RECUPERAÇÃO')
82 elif media >= 2:
83     # comment: :
84     print('REPROVADO')
85 else:
86     print('EXPULSO')
```

python 8_estrutura_de_decisao.py

REPROVADO

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

```
87
88 # if else elif
89
90 media = 1
91
92 if media >= 6:
93     print('APROVADO')
94 elif media >= 4:
95     print('RECUPERAÇÃO')
96 elif media >= 2:
97     # comment: :
98     print('REPROVADO')
99 else:
100     print('EXPULSO')
```

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>

python 8_estrutura_de_decisao.py

EXPULSO

(venv) PS C:\Users\Irene\Documents\aulaPython\Introdução_Python>