

OPERADORES E ESTRUTURAS DE DECISÃO

PROGRAMAÇÃO APLICADA A MATEMÁTICA

Meirylene Avelino

meirylenerea@id.uff.br

OPERADORES ARITMÉTICOS

Operador	Exemplo	Prioridade
(x)	$(1 + 2) * 3 \rightarrow 9$	1
**	$2 ** 3 \rightarrow 8$	2
+x	+15	3
-x	$-(5+3) \rightarrow -8$	3
*	$5 * 3 \rightarrow 15$	4
/	$5 / 3 \rightarrow 1.66$	4
//	$5 // 3 \rightarrow 1$	4
%	$5 \% 3 \rightarrow 2$	4
+	$5 + 3 \rightarrow 8$	5
-	$5 - 3 \rightarrow 2$	5

OPERADORES ARITMÉTICOS

- Operadores com a mesma prioridade (precedência) são analisados da esquerda para a direita
- Divisão de inteiros (//)
 - Resultado é somente a parte inteira da divisão
- Divisão (/)
 - Resultado fracionário

EXEMPLO

► Considerando

$$x = 512$$

$$y = 9.2 - (x // 10 - 14 / 5) + 14 * 0.1$$

► Resolução de y

$$y = 9.2 - (512 // 10 - 14 / 5) + 14 * 0.1$$

$$y = 9.2 - (51 - 14 / 5) + 14 * 0.1$$

$$y = 9.2 - (51 - 2.8) + 14 * 0.1$$

$$y = 9.2 - 48.2 + 14 * 0.1$$

$$y = 9.2 - 48.2 + 1.4$$

$$y = -39 + 1.4$$

$$y = -37.6$$

CONVERSÃO DE TIPOS

Em algumas situações o programador deseja transformar o tipo de uma expressão

- Para isso, basta envolver a variável a ser transformada por "tipo(variável)"

Exemplo: transformar um real em um inteiro

- `a = 5.1`
- `x = int(a)`
- `x` vale 5

Exemplo: transformar um inteiro em um real

- `b = 5`
- `y = float(b);`
- `y` vale 5.0

EXEMPLO

$x = \text{int}(3.3 / (5/2) - 5)$

$y = \text{int}(3.3) / (5/2) - 5$

► Resolução de x

$x = \text{int}(3.3 / (5/2) - 5)$

$x = \text{int}(3.3 / 2.5 - 5)$

$x = \text{int}(1.32 - 5)$

$x = \text{int}(-3.68)$

$x = -3$

► Resolução de y

$y = \text{int}(3.3) / (5/2) - 5$

$y = \text{int}(3.3) / 2.5 - 5$

$y = 3 / 2.5 - 5$

$y = 1.2 - 5$

$y = -3.8$

OPERADORES RELACIONAIS

Operador	Exemplo	Prioridade
$x < y$	$5 < 3 \rightarrow \text{False}$	6
$x \leq y$	$5 \leq 3 \rightarrow \text{False}$	6
$x > y$	$5 > 3 \rightarrow \text{True}$	6
$x \geq y$	$5 \geq 3 \rightarrow \text{True}$	6
$x == y$	$5 == 3 \rightarrow \text{False}$	6
$x \neq y$	$5 \neq 3 \rightarrow \text{True}$	6

Prioridade sempre inferior aos operadores aritméticos
Sempre têm **resultado booleano**

OPERADORES LÓGICOS

Operador	Exemplo	Prioridade
not x	not True → False	7
x and y	True and False → False	8
x or y	True or False → True	9

Prioridade sempre **inferior** aos operadores relacionais
Sempre têm **resultado booleano**

TABELA VERDADE

a	b	not a	a and b	a or b
True	True	False	True	True
True	False	False	False	True
False	True	True	False	True
False	False	True	False	False

Prioridade sempre **inferior** aos operadores relacionais
Sempre têm **resultado** booleano

ATRIBUIÇÃO

- Variável do lado esquerdo, valor ou expressão do lado direito
 $x = 0$
- Pode-se atribuir valor a várias variáveis ao mesmo tempo
 $x = y = z = 0$
x, y e z terão valor 0
- Pode-se também atribuir valores diferentes para variáveis diferentes ao mesmo tempo
 $x, y = 1, 2$
x terá o valor 1, e y terá o valor 2

EXEMPLO

$x = 10$

$y = -2$

$z = 5$

$w = x * y < z / x \text{ or } x / y > z * x \text{ and } z * y < x$

Como o valor de w seria avaliado pelo interpretador Python?

RESOLUÇÃO DE W

`w = x * y < z / x or x / y > z * x and z * y < x`

`w = 10 * -2 < 5 / 10 or 10 / -2 > 5 / 10 and 5 * -2 < 10`

`w = -20 < 5 / 10 or 10 / -2 > 5 / 10 and 5 * -2 < 10`

`w = -20 < 0.5 or 10 / -2 > 5 / 10 and 5 * -2 < 10`

`w = -20 < 0.5 or -5 > 5 / 10 and 5 * -2 < 10`

`w = -20 < 0.5 or -5 > 5 / 10 and 5 * -2 < 10`

`w = -20 < 0.5 or -5 > 0.5 and 5 * -2 < 10`

`w = -20 < 0.5 or -5 > 0.5 and -10 < 10`

`w = True or -5 > 0.5 and -10 < 10`

`w = True or False and -10 < 10`

`w = True or False and True`

`w = True or False`

`w = True`

REFERÊNCIA SOBRE OPERADORES E PRIORIDADES

Tutorial do Python 3

<https://docs.python.org/3.3/reference/expressions.html#operator-precedence>

DECISÃO

Mecanismos de decisão:

- **if ...**
 - Executa algo somente quando uma condição é verdadeira
- **if... else**
 - Bifurca a execução do código em função de uma condição
- **if... elif...**
 - Executa apenas o bloco em que a condição é verdadeira

DECISÃO DO TIPO IF...

Pseudocódigo

```
...  
se CONDIÇÃO então  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```

Python

```
...  
if CONDIÇÃO:  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```

DECISÃO DO TIPO IF...

- Executa o bloco de instruções somente se a condição for verdadeira
- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores
- O bloco de instruções é delimitado por indentação

EXEMPLO DE IF...

- Programa para informar quando um número inteiro é par:

```
numero = int(input("Entre com um numero: "))  
if (numero % 2 == 0):  
    print("O número é par")
```

if com instrução simples

EXEMPLO DE IF...

- Programa somar dois números, se o usuário desejar:

```
op = input("Deseja somar? (S/N) ")
if (op == "S"):
    x = int(input("Digite o primeiro numero:"))
    y = int(input("Digite o segundo numero:"))
    resultado = x + y
    print("O resultado da soma é", resultado)
print("Até a próxima! ")
```

if com bloco de instruções

DECISÃO DO TIPO IF...ELSE

Pseudocódigo

```
...  
Se CONDIÇÃO então  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
Senão  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```

Python

```
...  
if CONDIÇÃO:  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
else:  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```

DECISÃO DO TIPO IF...ELSE

- Executa um ou o outro bloco de instruções em função da condição ser verdadeira ou falsa
- Valem as mesmas regras para **if...**
- Qualquer combinação de instrução individual ou em bloco é aceita no corpo do **if** ou do **else**
- Podem ser aninhados com outras estruturas

EXEMPLO DE IF...ELSE

- Programa para informar se um número é par ou impar:

```
numero = int(input("Entre com um número: "))  
if numero % 2 == 0:  
    print("O número é par.")  
else:  
    print("O número é impar.")
```

EXEMPLO DE IF...ELSE

- Programa para somar ou multiplicar dois números

```
op = input("Deseja somar (S) ou multiplicar (M)?")
x = int(input("Digite o primeiro numero:"))
y = int(input("Digite o segundo numero:"))
if (op == "S"):
    r = x + y
    print("O resultado da soma é", r)
else:
    r = x * y
    print("O resultado da multiplicação é", r)
```

Problema: a multiplicação será realizada mesmo se o usuário digitar algo diferente de M

DECISÃO DO TIPO IF... ELIF...

Pseudocódigo

```
...  
Se CONDIÇÃO então  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
Senão Se CONDIÇÃO então  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```

Python

```
...  
if CONDIÇÃO:  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
elif CONDIÇÃO:  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N
```

DECISÃO DO TIPO IF... ELIF...

- Apenas o bloco no qual a condição é verdadeira é executado
- É possível colocar tantos **elif** quantos forem necessários
- Qualquer combinação de instrução individual ou em bloco é aceita no corpo do **if** ou do **elif**
- É possível adicionar um **else** ao final de tudo
 - Nesse caso, se nenhuma condição for verdadeira, o bloco do **else** será executado

```
...  
if CONDIÇÃO:  
    INSTRUÇÃO 1  
    ...  
    INSTRUÇÃO N  
elif CONDIÇÃO:  
    INSTRUÇÃO 1  
    ...  
    INSTRUÇÃO N  
elif CONDIÇÃO:  
    INSTRUÇÃO 1  
    ...  
    INSTRUÇÃO N  
else:  
    ...
```


EXEMPLO DE IF... ELSE

- Programa para somar ou multiplicar dois números

```
op = input("Deseja somar (S) ou multiplicar (M)?")
x = int(input("Digite o primeiro numero:"))
y = int(input("Digite o segundo numero:"))
if (op == "S"):
    r = x + y
    print("O resultado da soma é", r)
elif (op == "M"):
    r = x * y
    print("O resultado da multiplicação é", r)
else:
    print("Opção inválida")
```

Problema: x e y serão lidos mesmo se a opção for inválida

EXEMPLO DE IF... ELSE

- Programa para somar ou multiplicar dois números

```
op = input("Deseja somar (S) ou multiplicar (M)?")
if (op == "S"):
    x = int(input("Digite o primeiro numero:"))
    y = int(input("Digite o segundo numero:"))
    r = x + y
    print("O resultado da soma é", r)
elif (op == "M"):
    x = int(input("Digite o primeiro numero:"))
    y = int(input("Digite o segundo numero:"))
    r = x * y
    print("O resultado da multiplicação é", r)
else:
    print("Opção inválida")
```

SOLUÇÃO MAIS ELEGANTE, QUE FAZ A LEITURA DE X E Y UMA ÚNICA VEZ

- Programa para somar ou multiplicar dois números

```
op = input("Deseja somar (S) ou multiplicar (M)?")
if (op == "S" or op == "M"):
    x = int(input("Digite o primeiro numero:"))
    y = int(input("Digite o segundo numero:"))
    if (op == "S"):
        r = x + y
        print("O resultado da soma é", r)
    elif (op == "M"):
        r = x * y
        print("O resultado da multiplicação é", r)
    else:
        print("Opção inválida")
```

EXEMPLO: PROGRAMA PARA INFORMAR O NÚMERO DE DIAS DE UM MÊS QUALQUER

```
mes = int(input('Entre com um mês (1 a 12): '))
if (mes==1) or (mes==3) or (mes==5) or (mes==7) or (mes==8) or (mes==10) or (mes==12):
    print('Esse mes tem 31 dias')
elif (mes==4) or (mes==6) or (mes==9) or (mes==11):
    print('Esse mes tem 30 dias')
elif (mes==2):
    ano = int(input('Entre com o ano (4 dígitos): '))
    if (ano % 400 == 0) or (ano % 4 == 0) and (ano % 100 != 0):
        print('Esse mes tem 29 dias')
    else:
        print('Esse mes tem 28 dias')
else:
    print('Mês inválido')
```

USO DE VARIÁVEIS BOOLEANAS

```
imprimeMensagem = True
n = int(input("Digite um numero: "))
if (imprimeMensagem):
    print("O numero digitado foi", n)
else:
    print(n)
```

Note que NÃO usei
if (imprimeMensagem == True):
pois seria redundante!

USO DE NOT

```
imprimeMensagem = True
n = int(input("Digite um numero: "))
if not(imprimeMensagem):
    print(n)
else:
    print("O numero digitado foi", n)
```

CONDIÇÕES SIMPLIFICADAS

- Python permite simplificar condições, adicionando um AND implicitamente

```
a = int(input('Digite um numero: '))
b = int(input('Digite um numero: '))
c = int(input('Digite um numero: '))
if (a == b == c):
    print('Os 3 números são iguais')
else:
    print('Os 3 números não são iguais')
```

CONDIÇÕES SIMPLIFICADAS

- Python permite simplificar condições, adicionando um AND implicitamente

```
a = int(input('Digite um numero: '))
b = int(input('Digite um numero: '))
c = int(input('Digite um numero: '))
if (a == b == c):
    print('Os 3 números são iguais')
else:
    print('Os 3 números
```

Isso equivale à condição
(a == b and b == c)

CONDIÇÕES SIMPLIFICADAS

- Python permite simplificar condições, adicionando um AND implicitamente

```
a = int(input('Digite um numero: '))
b = int(input('Digite um numero: '))
c = int(input('Digite um numero: '))
if (a == b == c):
    print('Os 3 números são iguais')
else:
    print('Os 3 números são diferentes')
```

Isso equivale à condição
(a == b and b == c)
Nada se pode afirmar sobre
a == c

CONDIÇÕES SIMPLIFICADAS

- Para igualdades isso pode ser garantido por **transitividade**, mas operadores não **transitivos** apresentam problema

```
a = int(input('Digite um numero: '))
b = int(input('Digite um numero: '))
c = int(input('Digite um numero: '))
if (a != b != c):
    print('Os 3 números são diferentes')
else:
    print('Os números são iguais')
```

CONDIÇÕES SIMPLIFICADAS

- Para igualdades isso pode ser garantido por **transitividade**, mas operadores não **transitivos** apresentam problema

```
a = int(input('Digite um numero: '))
b = int(input('Digite um numero: '))
c = int(input('Digite um numero: '))
if (a != b != c):
    print('Os 3 números')
else:
    print('Os números são iguais')
```

Operador != não é transitivo.
**Nesse caso não é possível
garantir que a != c**

CONDIÇÕES SIMPLIFICADAS

- Solução: não usar condição simplificada nesses casos

```
a = int(input('Digite um numero: '))
b = int(input('Digite um numero: '))
c = int(input('Digite um numero: '))
if (a != b and b != c and a != c):
    print('Os 3 números são diferentes')
else:
    print('Os números são iguais')
```

ESCOPO DE VARIÁVEIS

- Variável só é visível dentro do seu “escopo”
- Variável declarada (usada pela primeira vez) fora de um bloco
 - *Pode ser acessada e modificada de qualquer lugar*
- Variável declarada (usada pela primeira vez) dentro de um bloco
 - *Só existe se esse bloco for executado*
- Revisitaremos esse assunto mais adiante na disciplina

EXEMPLO COM ERRO

```
nome = input('Digite o nome da pessoa: ')
sexo = input('Digite o sexo da pessoa (F/M): ')
if (sexo == 'M'):
    idade = input('Digite a idade da pessoa: ')
print(nome, 'tem', idade, 'anos')
```

EXEMPLO COM ERRO

```
nome = input('Digite o nome da pessoa: ')\nsexo = input('Digite o sexo da pessoa (F/M): ')\nif (sexo == 'M'):\n    idade = input('Digite a idade da pessoa: ')\nprint(nome, 'tem', idade, 'anos')
```

nome e sexo podem
ser acessadas em
qualquer lugar do
programa

EXEMPLO COM ERRO

```
nome = input('Digite o nome da pessoa: ')\nsexo = input('Digite o sexo da pessoa (F/M): ')\nif (sexo == 'M'):\n    idade = input('Digite a idade da pessoa: ')\nprint(nome, 'tem', idade, 'anos')
```

idade só existe se o código dentro do if for executado

EXEMPLO COM ERRO

```
nome = input('Digite o nome da pessoa: ')\nsexo = input('Digite o sexo da pessoa (F/M): ')\nif (sexo == 'M'):\n    idade = input('Digite a idade da pessoa: ')\nprint(nome, 'tem', idade, 'anos')
```

Se **sexo** for F, esse comando dará erro, pois variável **idade** não terá sido criada pelo Python

VOCÊS JÁ PODEM LER

- Capítulo 1 do livro Use a Cabeça: Programação, até a página 25

REFERÊNCIAS

- Slides baseados no curso da Vanessa Murta

OPERADORES E ESTRUTURAS DE DECISÃO

PROGRAMAÇÃO APLICADA A MATEMÁTICA

Meirylene Avelino

meirylenerea@id.uff.br