

PYTHON

Dia 04A
Prof Flávio Sousa Silva
Recode Pro

Agenda

- ❖ Operadores relacionais
- ❖ Operadores lógicos
- ❖ Expressões Lógicas
- ❖ Teste condicional simples
- ❖ Teste condicional Composto

Operadores Relacionais

- Em python podemos realizar a comparação em dois valores.
- A idéia é ter apenas dois resultados possíveis desta comparação (booleano).
 - Verdadeiro
 - Falso

Operadores Relacionais

- Usamos a comparação entre dois valores utilizados os chamados operadores relacionais.
- Em python são representados pelos símbolos:

Operadores Relacionais

Operador	Operação	Símbolo matemático
==	igualdade	=
>	Maior que	>
<	Menor que	<
!=	diferente	≠
>=	Maior ou igual	≥
<=	Menor ou igual	≤

Operadores Relacionais

- Exemplos do uso de operadores relacionais em python

```
>>> a = 10  
>>> b = 10  
>>> a == b  
True
```

```
>>> a = 10  
>>> b = 10  
>>> a > b  
False
```

```
>>> a = 8  
>>> b = 10  
>>> a < b  
True
```

Operadores Relacionais

- Exemplos do uso de operadores relacionais em python

```
>>> a = 8
>>> b = 10
>>> c = a > b
print(c)
False
```

```
>>> a = 8
>>> b = 10
>>> c = a != b
print(c)
True
```

```
>>> a = 10
>>> b = 10
>>> c = a != b
print(c)
False
```

Operadores Relacionais

- Exemplos do uso de operadores relacionais em python

```
>>> a = 8
>>> b = 10
>>> c = a >= b
print(c)
False
```

```
>>> a = 11
>>> b = 10
>>> c = a >= b
print(c)
True
```

```
>>> a = 10
>>> b = 10
>>> c = a >= b
print(c)
True
```


Operadores Relacionais

- Exemplos do uso de operadores relacionais em python

```
>>> a = 8  
>>> b = 10  
>>> c = a <= b  
print(c)  
True
```

```
>>> a = 11  
>>> b = 10  
>>> c = a <= b  
print(c)  
False
```

```
>>> a = 10  
>>> b = 10  
>>> c = a <= b  
print(c)  
True
```

Operadores Relacionais

- $a = 4$; $b = 10$; $c = 5.0$, $d = 1$ e $f = 5$

Expressão	Resultado
$a == c$	<input type="radio"/> True <input type="radio"/> False
$a < b$	<input type="radio"/> True <input type="radio"/> False
$d > b$	<input type="radio"/> True <input type="radio"/> False
$d = f$	<input type="radio"/> True <input type="radio"/> False
$a == b$	<input type="radio"/> True <input type="radio"/> False
$c < d$	<input type="radio"/> True <input type="radio"/> False
$b > a$	<input type="radio"/> True <input type="radio"/> False
$c >= f$	<input type="radio"/> True <input type="radio"/> False
$f >= c$	<input type="radio"/> True <input type="radio"/> False
$c <= c$	<input type="radio"/> True <input type="radio"/> False
$c <= f$	<input type="radio"/> True <input type="radio"/> False

Operadores Lógicos

- Para agrupar operações relacionais com lógica booleana
- Utilizamos operadores lógicos
- Python suporta três operadores básicos:
 - not (não)
 - and (e)
 - or (ou)

Operadores Lógicos

- Cada operador lógico obedece um conjunto simples de regras.
- Esse conjunto é chamado de tabela verdade.
- A tabela verdade demonstra o resultado de uma operação com um ou dois valores lógicos
 - O operador **not(não)** é um operadores unário.
 - O operador **or(ou)** é um operador binário
 - O operador **and(e)** é um operadores binário

Operadores Lógicos

- Tabela verdade do operador **not(não)**

Valor	not Valor
True	False
False	True

Operadores Lógicos

- Tabela verdade do operador **and(e)**

Valor1	Valor2	Valor1 and Valor2
True	True	True
True	False	False
False	True	False
False	False	False

Operadores Lógicos

- Tabela verdade do operador **or(ou)**

Valor1	Valor2	Valor1 or Valor2
True	True	True
True	False	True
False	True	True
False	False	False

Operadores Lógicos

- a = True
- b = False
- c = True.

Expressão	Resultado
a and a	<input type="radio"/> True <input type="radio"/> False
b and b	<input type="radio"/> True <input type="radio"/> False
not c	<input type="radio"/> True <input type="radio"/> False
not b	<input type="radio"/> True <input type="radio"/> False
not a	<input type="radio"/> True <input type="radio"/> False
a and b	<input type="radio"/> True <input type="radio"/> False
b and c	<input type="radio"/> True <input type="radio"/> False
a or c	<input type="radio"/> True <input type="radio"/> False
b or c	<input type="radio"/> True <input type="radio"/> False
a or c	<input type="radio"/> True <input type="radio"/> False
b or c	<input type="radio"/> True <input type="radio"/> False
c or a	<input type="radio"/> True <input type="radio"/> False
c or b	<input type="radio"/> True <input type="radio"/> False
c or c	<input type="radio"/> True <input type="radio"/> False
b or b	<input type="radio"/> True <input type="radio"/> False

Expressões Lógicas

- Os operadores lógicos podem ser combinados
- Isso forma uma expressão lógica
- Quando uma expressão tiver mais de um operador lógico segue-se a ordem
 - Primeiro o operador **not(não)**
 - Segundo o operador **and(e)**
 - Terceiro o operador **or(ou)**

Expressões Lógicas

- Ex:

```
>>> True or False and not  
True  
True or False and False  
True or False  
True
```

Expressões Lógicas

- Pode-se utilizar também a junção de operadores lógicos com relacionais.
- Neste caso deve-se seguir a seguinte ordem:
 - Primeiro avalia-se os operadores relacionais
 - Segundo avalia-se os operadores lógicos.

Expressões Lógicas

- Ex:

```
>>> a = 10
>>> b = 20
>>> a > 10 and b == 20
False and True
False
```

Testes Condicionais Simples

- Em um programa nem sempre todas as linhas serão executadas
- Muitas vezes haverá uma tomada de decisão
- A tomada de decisão será baseada no resultado de uma condição.
- A base dessas decisões consiste em expressões lógicas.

Testes Condicionais Simples

- As condições servem para selecionar quanto uma parte do programa deve ser ativada
- Ou quando simplesmente ignorada
- Em python, a estrutura de decisão é representada pela palavra **if**.
- **Ex:**

```
>>> if <condição> :  
>>>     bloco verdadeiro
```

Testes Condicionais Simples

- O bloco de código dentro da condição if somente será executado caso:
- O resultado da expressão lógica seja True (VERDADEIRA)
- Caso seja False (FALSA) será ignorada

Testes Condicionais Simples

- Exemplo de código em python

```
>>>a = int(input("valor para a"))
>>>b = int(input("valor para b"))
>>>if a > b:
>>>    print("a maior que b")
>>>if b > a:
>>>    print("a maior que b")
```


Testes Condicionais Simples

```
>>>salario = float(input("Digite o salário para calcular o  
imposto "))  
>>>base = salario  
>>>imposto = 0  
>>>if base > 3000 :  
>>> imposto = imposto + ((base - 3000) * 0.35)  
>>> base = 3000  
>>>if base > 1000 :  
>>> imposto = imposto + ((base - 1000) * 0.20)  
>>>print("Salario: R$%6.2f Imposto: R$%6.2f" % (salario,  
imposto))
```

Testes Condicionais Composto

- Em conjunto com o uso do if, podemos utilizar em python o **else**
- O else funciona em casos onde o bloco de código do if for ignorado
- A expressão lógica do if teve um resultado negativo
- O bloco contido na porção else será sempre executada em caso do teste lógico ser **False(FALSA)**
- Neste caso não precisamos de um outro if para avaliar a porção **FALSA**

Testes Condicionais Composto

- Exemplo de uso do else

```
>>> idade = int(input("Digite a idade de seu carro: "))
>>> if idade <= 3:
>>>     print("Seu carro é novo")
>>> else:
>>>     print("Seu carro é velho")
```

- Preste atenção que o recuo do else está no mesmo nível do if
- Note o uso dos dois pontos :
- Deve-se dar espaço no bloco do else