



Algoritmos e Programação de Computadores

Prof. Edson Borin

Instituto de Computação (IC/Unicamp)

O que vimos na aula passada?

Expressões Relacionais

Tipo bool

- Em Python o tipo **bool** especifica os valores booleanos falso (`False`) e verdadeiro (`True`).
- Podemos criar variáveis associadas a booleanos, mas o uso mais comum é na verificação de resultados de expressões relacionais e lógicas.

```
>>> a = True
>>> type(a)
<class 'bool'>
```

Operadores Relacionais

- Os operadores relacionais da linguagem Python são:
 - == : igualdade
 - != : diferente
 - > : maior que
 - < : menor que
 - >= : maior ou igual que
 - <= : menor ou igual que

Expressões Lógicas

Expressões Lógicas

- Expressões lógicas são aquelas que realizam uma operação lógica (**ou**, **e**, **não**, etc...) e retornam `True` ou `False` (como as expressões relacionais).
- Na linguagem Python temos os seguintes operadores lógicos:
 - **and** : operador E
 - **or**: operador OU
 - **not**: operador NÃO

Comandos Condicionais

Comandos Condicionais

- Uma variação do comando **if** é o **if/else**, cuja sintaxe é:

if expressão relacional ou lógica:

comandos executados se a expressão é verdadeira

else:

comandos executados se a expressão é falsa

Comandos Condicionais

- Uma variação do comando **if** é o **if/else**, cuja sintaxe é:

if expressão relacional ou lógica:

 comandos executados se a expressão é verdadeira


else:

 comandos executados se a expressão é falsa

comandos indentados

Comandos Condicionais

- Uma variação do comando `if` é o `if/else`, cuja sintaxe é:

`if` expressão relacional ou lógica:  **dois pontos**

 comandos executados se a expressão é verdadeira

`else:`  **dois pontos**

 comandos executados se a expressão é falsa

comandos indentados



Algoritmos e Programação de Computadores

Comandos Condicionais

Prof. Edson Borin

Instituto de Computação (IC/Unicamp)

Agenda

- Comandos `if-elif-else`
- Exemplos

Comandos `if-elif-else`

Quando apenas uma de várias alternativas é verdadeira podemos usar a construção `if-elif-else` que em Python é representado por:

```
if condicao_1:  
    comandos_1  
elif condicao_2:  
    comandos_2  
else:  
    comandos_3
```

Comandos `if-elif-else`

Quando apenas uma de várias alternativas é verdadeira podemos usar a construção `if-elif-else` que em Python é representado por:

```
if condicao_1:
    comandos_1
elif condicao_2:
    comandos_2
elif condicao_3:
    comandos_3
elif condicao_4:
    comandos_4
else:
    comandos_5
```

} *n* vezes

Escreva um programa que lê três números e imprime o maior deles.

```
numero1 = int(input("Digite um número: "))
numero2 = int(input("Digite um número: "))
numero3 = int(input("Digite um número: "))

if (numero1 >= numero2) and (numero1 >= numero3):
    print("O maior número é:", numero1)      # numero1 é o maior
else:
    if (numero2 >= numero3):
        print("O maior número é:", numero2) # numero2 é o maior
    else:
        print("O maior número é:", numero3) # numero3 é o maior
```


Escreva um programa que lê três números e imprime o maior deles.

```
numero1 = int(input("Digite um número: "))
numero2 = int(input("Digite um número: "))
numero3 = int(input("Digite um número: "))

if (numero1 >= numero2) and (numero1 >= numero3):
    print("O maior número é:", numero1)      # numero1 é o maior
else:
    if (numero2 >= numero3):
        print("O maior número é:", numero2) # numero2 é o maior
    else:
        print("O maior número é:", numero3) # numero3 é o maior
```

Escreva um programa que lê três números e imprime o maior deles.

```
numero1 = int(input("Digite um número: "))
numero2 = int(input("Digite um número: "))
numero3 = int(input("Digite um número: "))

if (numero1 >= numero2) and (numero1 >= numero3):
    print("O maior número é:", numero1)    # numero1 é o maior
elif (numero2 >= numero3):
    print("O maior número é:", numero2)    # numero2 é o maior
else:
    print("O maior número é:", numero3)    # numero3 é o maior
```

Escreva um programa que lê três números distintos e os imprime em ordem decrescente.

```
numero1 = int(input("Digite um número: "))
numero2 = int(input("Digite um número: "))
numero3 = int(input("Digite um número: "))

if (numero1 >= numero2 >= numero3):
    print(numero1, numero2, numero3)
elif (numero1 >= numero3 >= numero2):
    print(numero1, numero3, numero2)
elif (numero2 >= numero1 >= numero3):
    print(numero2, numero1, numero3)
elif (numero2 >= numero3 >= numero1):
    print(numero2, numero3, numero1)
elif (numero3 >= numero2 >= numero1):
    print(numero3, numero2, numero1)
elif (numero3 >= numero1 >= numero2):
    print(numero3, numero1, numero2)
```

Escreva um programa que simula o jogo conhecido como “Pedra, Papel e Tesoura” de um(a) jogador(a) contra o(a) outro(a).

Ganhador(a) Perdedor(a)

Pedra

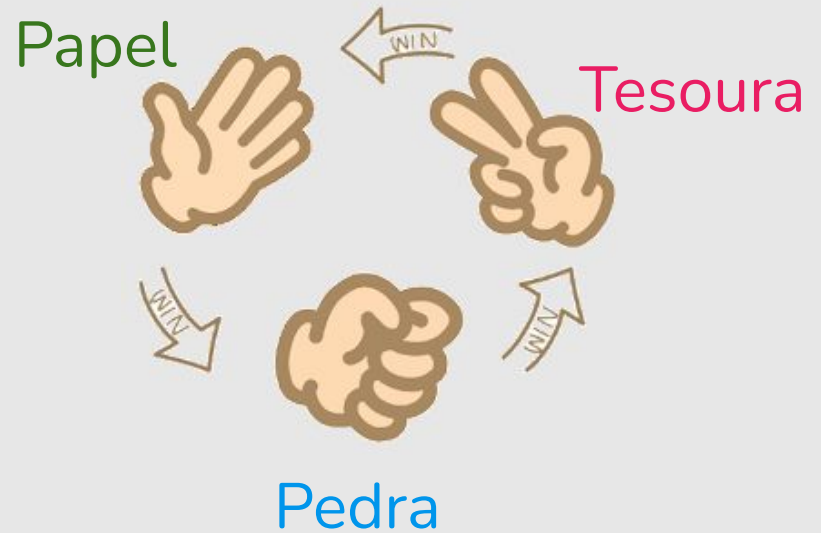
Tesoura

Tesoura

Papel

Papel

Pedra



Pedra, Papel e Tesoura

```
jogador1 = input("Jogador1, digite pedra, papel ou tesoura: ")
jogador2 = input("Jogador2, digite pedra, papel ou tesoura: ")

if (jogador1 == jogador2):
    print("Empate! Ninguém ganhou.") # empate
elif (jogador1 == "pedra" and jogador2 == "tesoura"):
    print("Jogador 1 ganhou.")
elif (jogador2 == "pedra" and jogador1 == "tesoura"):
    print("Jogador 2 ganhou.")
elif (jogador1 == "tesoura" and jogador2 == "papel"):
    print("Jogador 1 ganhou.")
elif (jogador2 == "papel" and jogador1 == "tesoura"):
    print("Jogador 2 ganhou.")
elif (jogador1 == "papel" and jogador2 == "pedra"):
    print("Jogador 1 ganhou.")
else:
    print("Jogador 2 ganhou.")
```

Pedra, Papel e Tesoura

```
jogador1 = input("Jogador1, digite pedra, papel ou tesoura: ")
jogador2 = input("Jogador2, digite pedra, papel ou tesoura: ")

if (jogador1 == jogador2):
    print("Empate! Ninguém ganhou.) # empate
elif (jogador1 == "pedra" and jogador2 == "tesoura"):
    print("Jogador 1 ganhou.")
elif (jogador1 == "tesoura" and jogador2 == "papel"):
    print("Jogador 1 ganhou.")
elif (jogador1 == "papel" and jogador2 == "pedra"):
    print("Jogador 1 ganhou.")
else:
    print("Jogador 2 ganhou.")
```

Pedra, Papel e Tesoura


```
jogador1 = input("Jogador1, digite pedra, papel ou tesoura: ")
jogador2 = input("Jogador2, digite pedra, papel ou tesoura: ")

if (jogador1 == jogador2):
    print("Empate! Ninguém ganhou.") # empate
elif (jogador1 == "pedra" and jogador2 == "tesoura") or \
      (jogador1 == "tesoura" and jogador2 == "papel") or \
      (jogador1 == "papel" and jogador2 == "pedra"):
    print("Jogador 1 ganhou.")
else:
    print("Jogador 2 ganhou.")
```


Pedra, Papel e Tesoura

```
jogador1 = input("Jogador1, digite pedra, papel ou tesoura: ")
jogador2 = input("Jogador2, digite pedra, papel ou tesoura: ")

if (jogador1 == jogador2):
    print("Empate! Ninguém ganhou.") # empate
elif (jogador1 == "pedra" and jogador2 == "tesoura") or \
      (jogador1 == "tesoura" and jogador2 == "papel") or \
      (jogador1 == "papel" and jogador2 == "pedra"):
    print("Jogador 1 ganhou.")
else:
    print("Jogador 2 ganhou.")
```



O comando continua na próxima linha.

Pedra, Papel e Tesoura

- Podemos associar objetos (pedra, papel, tesoura) a números.
- Ou seja, teremos:
 - **pedra** = 0
 - **papel** = 1
 - **tesoura** = 2

Pedra, Papel e Tesoura

```
jogador1 = int(input("Jogador1, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))
jogador2 = int(input("Jogador2, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))

pedra = 0
papel = 1
tesoura = 2

if (jogador1 == jogador2):
    print("Empate! Ninguém ganhou.") # empate
elif (jogador1 == pedra and jogador2 == tesoura) or \
      (jogador1 == tesoura and jogador2 == papel) or \
      (jogador1 == papel and jogador2 == pedra):
    print("Jogador 1 ganhou.")
else:
    print("Jogador 2 ganhou.")
```

Pedra, Papel e Tesoura

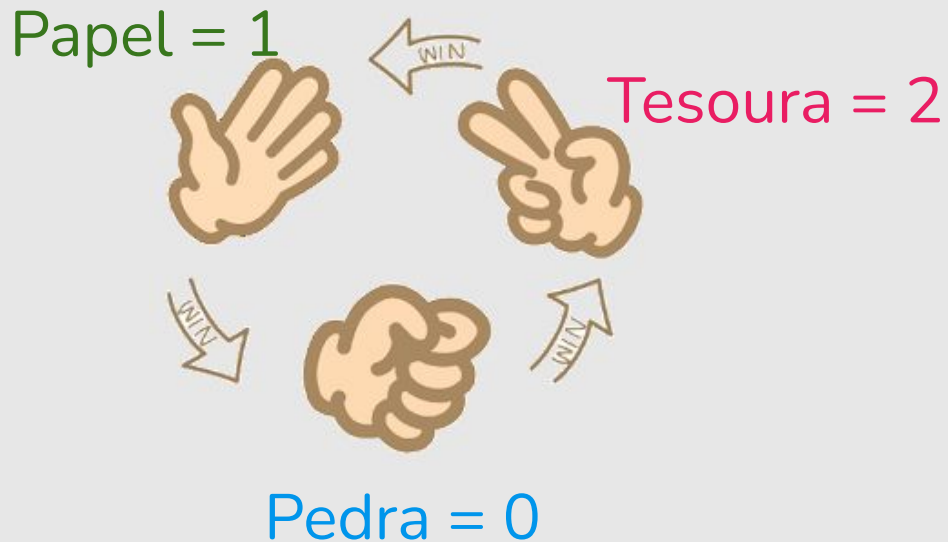
```
jogador1 = int(input("Jogador1, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))
jogador2 = int(input("Jogador2, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))

pedra = 0
papel = 1
tesoura = 2

if (jogador1 == jogador2):
    print("Empate! Ninguém ganhou.") # erro
elif (jogador1 - jogador2) % 3 == 1:
    print("Jogador 1 ganhou.")
else:
    print("Jogador 2 ganhou.")
```

j1	j2	j1-j2	(j1-j2)%3	Vencedor
2	1	1	1	j1
2	0	2	2	j2
1	0	1	1	j1
1	2	-1	2	j2
0	1	-1	2	j2
0	2	-2	1	j1

Pedra, Papel e Tesoura



$$(0 - 2) \% 3 = 1 \text{ (Ganhou)}$$

$$(2 - 1) \% 3 = 1 \text{ (Ganhou)}$$

$$(1 - 0) \% 3 = 1 \text{ (Ganhou)}$$

$$(0 - 1) \% 3 = 2$$

$$(1 - 2) \% 3 = 2$$

$$(2 - 0) \% 3 = 2$$

Referências

- Condicionais encadeados
 - <https://panda.ime.usp.br/pensepy/static/pensepy/06-Selecao/selecao.html#condicionais-encadeados>
 - <https://runestone.academy/runestone/static/thinkcspy/Selection/Chainedconditionals.html>
 - <https://github.com/iviarcio/mc102/blob/master/04.Controle%20de%20Fluxo%20-%20Condicionais.ipynb>

Créditos

— — —

Os *slides* deste curso foram baseados nos slides produzidos e cedidos gentilmente pela Professora Sandra Ávila, do Instituto de Computação da Unicamp.