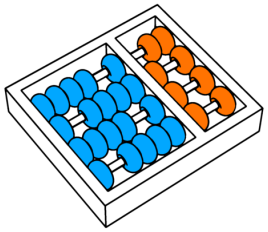


MC102 – Aula 03
Comandos Condicionais

Algoritmos e Programação de Computadores

Zanoni Dias

2019

Instituto de Computação

Roteiro

Expressões Relacionais

Expressões Lógicas

Comandos Condicionais

Expressões Relacionais

Tipo bool

- Já vimos que o tipo **bool** é utilizado para representar os valores booleanos verdadeiro (**True**) e falso (**False**).

```
1 a = True
2 b = False
```

- O uso mais comum dessas variáveis é na verificação de expressões relacionais e lógicas.

Expressões Relacionais

- Expressões relacionais são aquelas que realizam uma comparação entre duas expressões e retornam:
  - **True**, se o resultado for verdadeiro.
  - **False**, se o resultado for falso.
- Os operadores relacionais são:

`==`	igualdade.
`!=`	diferente.
`>`	maior que.
`<`	menor que.
`>=`	maior ou igual que.
`<=`	menor ou igual que.

Expressões Relacionais

- <expressão> `==` <expressão>: retorna verdadeiro quando as expressões forem iguais.

```
a == (10 * 2) # a = 20
# True
b == (10 * 2) # b = 21
# False
```

- <expressão> `!=` <expressão>: retorna verdadeiro quando as expressões forem diferentes.

```
a != (10 * 2) # a = 20
# False
b != (10 * 2) # b = 21
# True
```

Expressões Relacionais

- <expressão> **>** <expressão>: retorna verdadeiro quando a expressão da esquerda tiver valor maior que a expressão da direita.

```
# a = 20 e b = 21
a > b
# False
```

- <expressão> **<** <expressão>: retorna verdadeiro quando a expressão da esquerda tiver valor menor que a expressão da direita.

```
# a = 20 e b = 21
a < b
# True
```

Expressões Relacionais

- <expressão> `>=` <expressão>: retorna verdadeiro quando a expressão da esquerda tiver valor maior ou igual que a expressão da direita.

```
# a = 20 e b = 21
a >= b
# False
```

- <expressão> `<=` <expressão>: retorna verdadeiro quando a expressão da esquerda tiver valor menor ou igual que a expressão da direita.

```
# a = 20 e b = 21
a <= b
# True
```

Expressões Relacionais com Strings

- Ordem considerada para os caracteres do alfabeto:
  - *ABC...XYZabc...xyz*

```
"a" > "b"
# False
"a" == "a"
# True
"a" == "A"
# False
"Z" < "a"
# True
"z" < "a"
# False
```

Expressões Relacionais com Strings

- Ordem considerada para os caracteres do alfabeto:
  - *ABC...XYZabc...xyz*

```
"azzzz" < "zaaaa"
# True
"azzzz" < "Zaaaa"
# False
"3" == 3
# False
3 > "4"
# Traceback (most recent call last):
#   File "<stdin>", line 1, in <module>
# TypeError: '>' not supported between instances of 'int'
    and 'str'
```

Exemplo

O que será impresso pelo código a seguir?

```
1 print((3 * 4) / 2 == (2 * 3))
2 # ?
3 print((4 / 3) <= 1.33)
4 # ?
```

Exemplo

O que será impresso pelo código a seguir?

```
1 print((3 * 4) / 2 == (2 * 3))
2 # True
3 print((4 / 3) <= 1.33)
4 # False
```

Expressões Lógicas

Expressões Lógicas

- Expressões lógicas são aquelas que realizam uma operação lógica e retornam verdadeiro ou falso (como as expressões relacionais).
- Os operadores lógicos são:
  - `and`: operador E.
  - `or`: operador OU.
  - `not`: operador NÃO.

Operador Lógico and

- <expressão1> **and** <expressão2>: retorna verdadeiro quando ambas as expressões são verdadeiras.
- Sua tabela verdade é:

<expressão1>	<expressão2>	resultado
V	V	V
V	F	F
F	V	F
F	F	F

```
a = 0
b = 10
(a == 0) and (b == 0)
# False
(a == 0) and (b != 0)
# True
```

Operador Lógico or

- <expressão1> **or** <expressão2>: retorna verdadeiro quando pelo menos uma das expressões é verdadeira.
- Sua tabela verdade é:

<expressão1>	<expressão2>	resultado
V	V	V
V	F	V
F	V	V
F	F	F

```
1 a = 0
2 b = 10
3 (a == 0) or (b == 0)
4 # True
5 (a != 0) or (b == 0)
6 # False
```

Operador Lógico not

- `not` <expressão>: retorna verdadeiro quando a expressão é falsa e vice-versa.
- Sua tabela verdade é:

<expressão>	resultado
V	F
F	V

```
a = True
b = False
not a
# False
not b
# True
```

Simplificações Úteis

- `not (a == b)` é equivalente a `(a != b)`
- `not (a > b)` é equivalente a `(a <= b)`
- `not (a < b)` é equivalente a `(a >= b)`

Exemplo

O que será impresso pelo código a seguir?

```
1 a = True
2 b = False
3 print(not (a or b))
4 # ?
5 print(not (a and b))
6 # ?
7 print(not (14 > 100) and not (10 > 20))
8 # ?
```

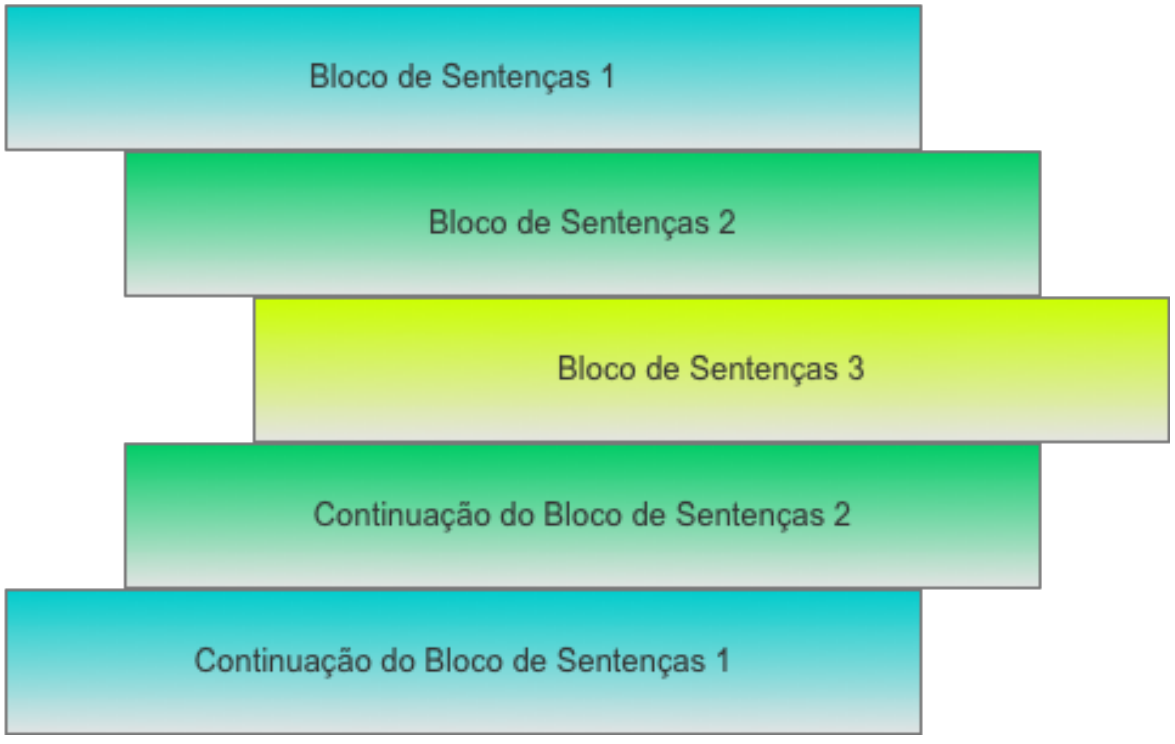
Exemplo

O que será impresso pelo código a seguir?

```
a = True
b = False
print(not (a or b))
# False
print(not (a and b))
# True
print(not (14 > 100) and not (10 > 20))
# True
```

Comandos Condicionais

Blocos de Comandos



- Um bloco de código é um conjunto de comandos agrupados.
- Os programas Python são estruturados através de indentação, ou seja, os blocos de código são definidos pelo seu espaçamento (*tabs*) em relação ao início da linha.

Comandos Condicionais

- O principal comando condicional é o `if`:

```
if <condição>:
# bloco a ser executado se a condição for verdadeira
  <comando1>
  <comando2>
  ...
  <comandoY>
```

- O bloco de comandos é executado somente se a condição (expressão relacional, expressão lógica ou variável booleana) for verdadeira.
- Na estrutura do comando `if` sempre há um “:” após a condição.

Comandos Condicionais

- O programa a seguir verifica se um valor fornecido na entrada é ímpar.

```
1 a = int(input("Digite um número inteiro: "))
2
3 if (a % 2) == 1:
4   print("Número ímpar")
```

Comandos Condicionais

- O programa a seguir verifica se um valor fornecido na entrada é par ou ímpar.

```
1 a = int(input("Digite um número inteiro: "))
2
3 if (a % 2) == 0:
4   print("Número par")
5 if (a % 2) == 1:
6   print("Número ímpar")
```

Comandos Condicionais

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

```
if <condição>:
  # bloco a ser executado se a condição for verdadeira
  <comando>
  ...
  <comando>
else:
  # bloco a ser executado se a condição for falsa
  <comando>
  ...
  <comando>
```

Comandos Condicionais

- O programa a seguir verifica se um valor fornecido na entrada é par ou ímpar.

```
1 a = int(input("Digite um número inteiro: "))
2
3 if (a % 2) == 0:
4   print("Número par")
5 else:
6   print("Número ímpar")
```

Comandos Condicionais

- O programa a seguir verifica o maior entre dois números.

```
1 a = int(input("Digite o primeiro número: "))
2 b = int(input("Digite o segundo número: "))
3
4 if a > b:
5   print("O maior número é", a)
6 else:
7   print("O maior número é", b)
```

Comandos Condicionais

- O programa a seguir compara dois números.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))

if a == b:
  print("Os dois números são iguais")
else:
  if a > b:
    print("O maior número é o primeiro")
  else:
    print("O maior número é o segundo")
```

Comandos Condicionais

- Podemos simplificar o código anterior utilizando o **`elif`**.

```
1 a = int(input("Digite o primeiro número: "))
2 b = int(input("Digite o segundo número: "))
3
4 if a == b:
5   print("Os dois números são iguais")
6 elif a > b:
7   print("O maior número é o primeiro")
8 else:
9   print("O maior número é o segundo")
```

Comandos Condicionais

- O `elif` é utilizado quando queremos fazer o teste de várias alternativas.

```
ra = input("Entre com o RA de um aluno: ")

if ra == "155446":
  print("Gabriel Siqueira")
elif ra == "192804":
  print("Alexsandro Alexandrino")
elif ra == "209823":
  print("Ana Paula Dantas")
elif ra == "188948":
  print("Klairton Brito")
# ...
elif ra == "999999":
  print("...")
```

Comandos Condicionais

- O `elif` é utilizado quando queremos fazer o teste de várias alternativas.

```
ra = input("Entre com o RA de um aluno: ")

if ra == "155446":
  print("Gabriel Siqueira")
elif ra == "192804":
  print("Alexsandro Alexandrino")
elif ra == "209823":
  print("Ana Paula Dantas")
elif ra == "188948":
  print("Klairton Brito")
# ...
elif ra == "999999":
  print("...")
else:
  print("Aluno não encontrado")
```

Exemplo

```
1  a = int(input())
2
3  if a > 3:
4    if a < 7:
5      print("a")
6  else:
7    if a > -10:
8      print("b")
9    else:
10     print("c")
```

- No código acima, o que será impresso...
  - ... quando `a = 7`?
  - ... quando `a = 5`?
  - ... quando `a = 10`?
  - ... quando `a = -5`?
  - ... quando `a = -15`?

Exemplo

```
1  a = int(input())
2
3  if a > 3:
4    if a < 7:
5      print("a")
6  else:
7    if a > -10:
8      print("b")
9    else:
10     print("c")
```

- No código acima, o que será impresso...
  - ... quando `a = 7`? Nada.
  - ... quando `a = 5`? “a”.
  - ... quando `a = 10`? Nada.
  - ... quando `a = -5`? “b”.
  - ... quando `a = -15`? “c”.

Exercícios

1. Escreva um programa que, dados três números inteiros, imprima o menor deles.
2. Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.
3. Escreva um programa que, dadas duas datas, determine qual delas ocorreu cronologicamente primeiro. Para cada um das duas datas, leia três números referentes ao dia, mês e ano, respectivamente.

Exercício 1 - Resposta

- Escreva um programa que, dados três números inteiros, imprima o menor deles.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))
```

Exercício 1 - Resposta

- Escreva um programa que, dados três números inteiros, imprima o menor deles.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c:
  print(a)
```

Exercício 1 - Resposta

- Escreva um programa que, dados três números inteiros, imprima o menor deles.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c:
  print(a)
if b <= a and b <= c:
  print(b)


```

Exercício 1 - Resposta

- Escreva um programa que, dados três números inteiros, imprima o menor deles.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c:
  print(a)
if b <= a and b <= c:
  print(b)
if c <= a and c <= b:
  print(c)
```

Exercício 1 - Resposta

- Este programa tem um comportamento indesejado quando o menor número não é único. Como corrigí-lo?

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c:
  print(a)
if b <= a and b <= c:
  print(b)
if c <= a and c <= b:
  print(c)
```

Exercício 1 - Resposta

- Escreva um programa que, dados três números inteiros, imprima o menor deles.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c:
  print(a)
elif b <= a and b <= c:
  print(b)
elif c <= a and c <= b:
  print(c)
```

Exercício 1 - Resposta

- Escreva um programa que, dados três números inteiros, imprima o menor deles.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c:
  print(a)
elif b <= a and b <= c:
  print(b)
else:
  print(c)
```

Exercício 1 - Resposta

- Escreva um programa que, dados três números inteiros, imprima o menor deles.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c:
  print(a)
elif b <= c:
  print(b)
else:
  print(c)
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and b <= c:
  print(a, b, c)
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and b <= c:
  print(a, b, c)
elif a <= c and c <= b:
  print(a, c, b)
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and b <= c:
  print(a, b, c)
elif a <= c and c <= b:
  print(a, c, b)
elif b <= a and a <= c:
  print(b, a, c)
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and b <= c:
  print(a, b, c)
elif a <= c and c <= b:
  print(a, c, b)
elif b <= a and a <= c:
  print(b, a, c)
elif b <= c and c <= a:
  print(b, c, a)
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and b <= c:
  print(a, b, c)
elif a <= c and c <= b:
  print(a, c, b)
elif b <= a and a <= c:
  print(b, a, c)
elif b <= c and c <= a:
  print(b, c, a)
elif c <= a and a <= b:
  print(c, a, b)


```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and b <= c:
  print(a, b, c)
elif a <= c and c <= b:
  print(a, c, b)
elif b <= a and a <= c:
  print(b, a, c)
elif b <= c and c <= a:
  print(b, c, a)
elif c <= a and a <= b:
  print(c, a, b)
elif c <= b and b <= a:
  print(c, b, a)
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and b <= c:
  print(a, b, c)
elif a <= c and c <= b:
  print(a, c, b)
elif b <= a and a <= c:
  print(b, a, c)
elif b <= c and c <= a:
  print(b, c, a)
elif c <= a and a <= b:
  print(c, a, b)
else:
  print(c, b, a)
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c: # O menor é o primeiro (a)
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c: # O menor é o primeiro (a)
  if b <= c:
    print(a, b, c)
  else:
    print(a, c, b)
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c: # O menor é o primeiro (a)
  if b <= c:
    print(a, b, c)
  else:
    print(a, c, b)
elif b <= c:          # O menor é o segundo (b)
```

Exercício 2 - Resposta

- Escreva um programa que, dados três números inteiros, imprima os números em ordem crescente.

```
a = int(input("Digite o primeiro número: "))
b = int(input("Digite o segundo número: "))
c = int(input("Digite o terceiro número: "))

if a <= b and a <= c: # O menor é o primeiro (a)
  if b <= c:
    print(a, b, c)
  else:
    print(a, c, b)
elif b <= c:          # O menor é o segundo (b)
  if a <= c:
    print(b, a, c)
  else:
    print(b, c, a)
# ...

```

Exercício 2 - Resposta (Continuação)

```
1
2 # ...
3 else:                    # O menor é o terceiro (c)
4
5
6
7
```

Exercício 2 - Resposta (Continuação)

```
1
2 # ...
3 else:                    # O menor é o terceiro (c)
4   if a <= b:
5     print(c, a, b)
6   else:
7     print(c, b, a)
```

Funções `min` e `max`

- Python possui as funções **`min`** (mínimo) e **`max`** (máximo).
- A função **`min`** retorna o menor valor dentre todos os valores passados como parâmetro.

```
a = 10
min(100, 5, a, 7)
# 5
```

- A função **`max`** retorna o maior valor dentre todos os valores passados como parâmetro.

```
max(100, 5, a, 7)
# 100
```

- Refaça os dois exercícios anteriores sem utilizar comandos condicionais. Dica: use as funções **`min`** e **`max`**.

Exercício 3 - Resposta

- Escreva um programa que, dadas duas datas, determine qual delas ocorreu cronologicamente primeiro. Para cada um das duas datas, leia três números referentes ao dia, mês e ano, respectivamente.

```
dia1 = int(input("Digite o dia da primeira data: "))
mes1 = int(input("Digite o mês da primeira data: "))
ano1 = int(input("Digite o ano da primeira data: "))

dia2 = int(input("Digite o dia da segunda data: "))
mes2 = int(input("Digite o mês da segunda data: "))
ano2 = int(input("Digite o ano da segunda data: "))

# ...
```

Exercício 3 - Resposta (Continuação)

```
# ...

if ano1 < ano2:
```

Exercício 3 - Resposta (Continuação)

```
# ...

if ano1 < ano2:
  print(dia1, mes1, ano1, sep="/")
```

Exercício 3 - Resposta (Continuação)

```
# ...

if ano1 < ano2:
  print(dia1, mes1, ano1, sep="/")
elif ano2 < ano1:
  print(dia2, mes2, ano2, sep="/")
```

Exercício 3 - Resposta (Continuação)

```
# ...

if ano1 < ano2:
  print(dia1, mes1, ano1, sep="/")
elif ano2 < ano1:
  print(dia2, mes2, ano2, sep="/")
elif mes1 < mes2:
  print(dia1, mes1, ano1, sep="/")
```

Exercício 3 - Resposta (Continuação)

```
# ...

if ano1 < ano2:
  print(dia1, mes1, ano1, sep="/")
elif ano2 < ano1:
  print(dia2, mes2, ano2, sep="/")
elif mes1 < mes2:
  print(dia1, mes1, ano1, sep="/")
elif mes2 < mes1:
  print(dia2, mes2, ano2, sep="/")
```

Exercício 3 - Resposta (Continuação)

```
# ...

if ano1 < ano2:
  print(dia1, mes1, ano1, sep="/")
elif ano2 < ano1:
  print(dia2, mes2, ano2, sep="/")
elif mes1 < mes2:
  print(dia1, mes1, ano1, sep="/")
elif mes2 < mes1:
  print(dia2, mes2, ano2, sep="/")
elif dia1 < dia2:
  print(dia1, mes1, ano1, sep="/")


```

Exercício 3 - Resposta (Continuação)

```
# ...

if ano1 < ano2:
  print(dia1, mes1, ano1, sep="/")
elif ano2 < ano1:
  print(dia2, mes2, ano2, sep="/")
elif mes1 < mes2:
  print(dia1, mes1, ano1, sep="/")
elif mes2 < mes1:
  print(dia2, mes2, ano2, sep="/")
elif dia1 < dia2:
  print(dia1, mes1, ano1, sep="/")
else:
  print(dia2, mes2, ano2, sep="/")
```

Exercícios

4. Escreva um programa que calcule as raízes de uma equação de segundo grau. O seu programa deve receber três números a, b e c, sendo que a equação é definida como $ax^2 + bx + c = 0$. O seu programa também deve tratar o caso em que $a = 0$.
5. Escreva um programa que simula o jogo conhecido como "Pedra, Papel e Tesoura" de um jogador A contra um jogador B. O programa deve ler a escolha do jogador A e a escolha do jogador B. Por fim, o programa deve indicar quem foi o vencedor.

Exercício 4 - Resposta

- Escreva um programa que calcule as raízes de uma equação de segundo grau. O seu programa deve receber três números a, b e c, sendo que a equação é definida como $ax^2 + bx + c = 0$. O seu programa também deve tratar o caso em que $a = 0$.

```
1  a = float(input("Digite o coeficiente a: "))
2  b = float(input("Digite o coeficiente b: "))
3  c = float(input("Digite o coeficiente c: "))
4
5
6
7
8
9
10
11
```

Exercício 4 - Resposta

- Escreva um programa que calcule as raízes de uma equação de segundo grau. O seu programa deve receber três números a, b e c, sendo que a equação é definida como $ax^2 + bx + c = 0$. O seu programa também deve tratar o caso em que $a = 0$.

```
1  a = float(input("Digite o coeficiente a: "))
2  b = float(input("Digite o coeficiente b: "))
3  c = float(input("Digite o coeficiente c: "))
4
5  if a == 0: # equação do primeiro grau
6
7
8
9
10
11
```

Exercício 4 - Resposta

- Escreva um programa que calcule as raízes de uma equação de segundo grau. O seu programa deve receber três números a, b e c, sendo que a equação é definida como $ax^2 + bx + c = 0$. O seu programa também deve tratar o caso em que $a = 0$.

```
1  a = float(input("Digite o coeficiente a: "))
2  b = float(input("Digite o coeficiente b: "))
3  c = float(input("Digite o coeficiente c: "))
4
5  if a == 0: # equação do primeiro grau
6    if b == 0:
7      print("Não existe raíz.")
8
9
10
11
```

Exercício 4 - Resposta

- Escreva um programa que calcule as raízes de uma equação de segundo grau. O seu programa deve receber três números a, b e c, sendo que a equação é definida como $ax^2 + bx + c = 0$. O seu programa também deve tratar o caso em que $a = 0$.

```
1  a = float(input("Digite o coeficiente a: "))
2  b = float(input("Digite o coeficiente b: "))
3  c = float(input("Digite o coeficiente c: "))
4
5  if a == 0: # equação do primeiro grau
6    if b == 0:
7      print("Não existe raíz.")
8    else:
9      raiz = (-c / b)
10     print("A raiz é:", raiz)
11
```

Exercício 4 - Resposta

- Escreva um programa que calcule as raízes de uma equação de segundo grau. O seu programa deve receber três números a, b e c, sendo que a equação é definida como $ax^2 + bx + c = 0$. O seu programa também deve tratar o caso em que $a = 0$.

```
1  a = float(input("Digite o coeficiente a: "))
2  b = float(input("Digite o coeficiente b: "))
3  c = float(input("Digite o coeficiente c: "))
4
5  if a == 0: # equação do primeiro grau
6    if b == 0:
7      print("Não existe raíz.")
8    else:
9      raiz = (-c / b)
10     print("A raiz é:", raiz)
11 # ...
```

Exercício 4 - Resposta (Continuação)

```
1  # ...
2  else: # equação do segundo grau
3    delta = (b ** 2) - (4 * a * c)
4
5
6
7
8
9
10
11
12
```

Exercício 4 - Resposta (Continuação)

```
# ...
else: # equação do segundo grau
  delta = (b ** 2) - (4 * a * c)
  if delta < 0:
```

Exercício 4 - Resposta (Continuação)

```
# ...
else: # equação do segundo grau
  delta = (b ** 2) - (4 * a * c)
  if delta < 0:
    print("Não existem raízes reais.")
```

Exercício 4 - Resposta (Continuação)

```
# ...
else: # equação do segundo grau
  delta = (b ** 2) - (4 * a * c)
  if delta < 0:
    print("Não existem raízes reais.")
  elif delta != 0:
```

Exercício 4 - Resposta (Continuação)

```
# ...
else: # equação do segundo grau
  delta = (b ** 2) - (4 * a * c)
  if delta < 0:
    print("Não existem raízes reais.")
  elif delta != 0:
    raiz1 = (-b + delta ** (1 / 2)) / (2 * a)
    raiz2 = (-b - delta ** (1 / 2)) / (2 * a)
    print("As raízes são:", raiz1, "e", raiz2)
```

Exercício 4 - Resposta (Continuação)

```
# ...
else: # equação do segundo grau
  delta = (b ** 2) - (4 * a * c)
  if delta < 0:
    print("Não existem raízes reais.")
  elif delta != 0:
    raiz1 = (-b + delta ** (1 / 2)) / (2 * a)
    raiz2 = (-b - delta ** (1 / 2)) / (2 * a)
    print("As raízes são:", raiz1, "e", raiz2)
  else:


```

Exercício 4 - Resposta (Continuação)

```
# ...
else: # equação do segundo grau
  delta = (b ** 2) - (4 * a * c)
  if delta < 0:
    print("Não existem raízes reais.")
  elif delta != 0:
    raiz1 = (-b + delta ** (1 / 2)) / (2 * a)
    raiz2 = (-b - delta ** (1 / 2)) / (2 * a)
    print("As raízes são:", raiz1, "e", raiz2)
  else:
    raiz = -b / (2 * a)
    print("A raiz é:", raiz)
```

Exercício 5 - Resposta

- Escreva um programa que simula o jogo conhecido como “Pedra, Papel e Tesoura” de um jogador A contra um jogador B. O programa deve ler a escolha do jogador A e a escolha do jogador B. Por fim, o programa deve indicar quem foi o vencedor.

```
1  jogadorA = input("Digite a primeira escolha: ")
2  jogadorB = input("Digite a segunda escolha: ")
3
4
5
6
7
8
9
10
11
```

Exercício 5 - Resposta

- Escreva um programa que simula o jogo conhecido como “Pedra, Papel e Tesoura” de um jogador A contra um jogador B. O programa deve ler a escolha do jogador A e a escolha do jogador B. Por fim, o programa deve indicar quem foi o vencedor.

```
1  jogadorA = input("Digite a primeira escolha: ")
2  jogadorB = input("Digite a segunda escolha: ")
3
4  if jogadorA == "pedra":
5
6
7
8
9
10
11
```

Exercício 5 - Resposta

- Escreva um programa que simula o jogo conhecido como "Pedra, Papel e Tesoura" de um jogador A contra um jogador B. O programa deve ler a escolha do jogador A e a escolha do jogador B. Por fim, o programa deve indicar quem foi o vencedor.

```
1  jogadorA = input("Digite a primeira escolha: ")
2  jogadorB = input("Digite a segunda escolha: ")
3
4  if jogadorA == "pedra":
5    if jogadorB == "pedra":
6      print("Empate")
7    elif jogadorB == "tesoura":
8      print("O jogador A ganhou")
9    else:
10     print("O jogador B ganhou")
11 # ...
```

Exercício 5 - Resposta (Continuação)

```
# ...
elif jogadorA == "tesoura":
```

Exercício 5 - Resposta (Continuação)

```
# ...
elif jogadorA == "tesoura":
  if jogadorB == "pedra":
    print("O jogador B ganhou")
  elif jogadorB == "tesoura":
    print("Empate")
  else:
    print("O jogador A ganhou")
```

Exercício 5 - Resposta (Continuação)

```
# ...
elif jogadorA == "tesoura":
  if jogadorB == "pedra":
    print("O jogador B ganhou")
  elif jogadorB == "tesoura":
    print("Empate")
  else:
    print("O jogador A ganhou")
else: # jogadorA == "papel"
```

Exercício 5 - Resposta (Continuação)

```
# ...
elif jogadorA == "tesoura":
  if jogadorB == "pedra":
    print("O jogador B ganhou")
  elif jogadorB == "tesoura":
    print("Empate")
  else:
    print("O jogador A ganhou")
else: # jogadorA == "papel"
  if jogadorB == "pedra":
    print("O jogador A ganhou")
  elif jogadorB == "tesoura":
    print("O jogador B ganhou")
  else:
    print("Empate")
```

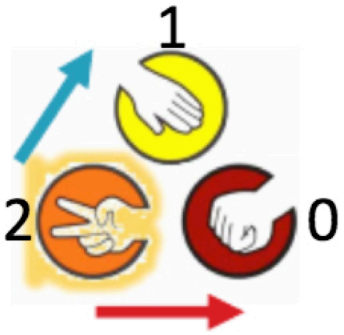
Exercício 5 - Nova Versão

- Associar objetos a números é uma forma de abstração. No código a seguir usamos as seguintes associações:
  - `pedra = 0`
  - `papel = 1`
  - `tesoura = 2`
- O resultado da expressão `(jogadorA - jogadorB % 3)` indica, de forma única, o vencedor da partida.
- Complete o código analisando o resultado da expressão anterior.

Exercício 5 - Nova Versão

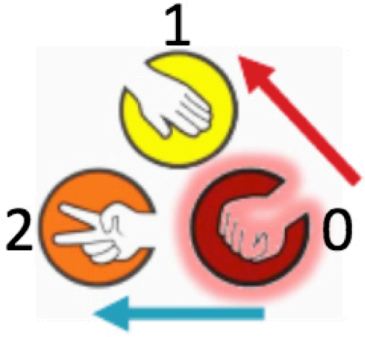
```
1  print("Pedra = 0")
2  print("Papel = 1")
3  print("Tesoura = 2")
4
5  jogadorA = int(input("Digite a primeira escolha: "))
6  jogadorB = int(input("Digite a segunda escolha: "))
7
8  resultado = (jogadorA - jogadorB) % 3
9
10 # Como determinar quem venceu com base na conta acima?
11
12
13
14
15
```

Exercício 5 - Nova Versão




```
(2 - 1) % 3 = 1
(2 - 0) % 3 = 2
```


```
(1 - 0) % 3 = 1
(1 - 2) % 3 = 2
```




```
(0 - 2) % 3 = 1
(0 - 1) % 3 = 2
```

Exercício 5 - Nova Versão

```
1  print("Pedra = 0")
2  print("Papel = 1")
3  print("Tesoura = 2")
4
5  jogadorA = int(input("Digite a primeira escolha: "))
6  jogadorB = int(input("Digite a segunda escolha: "))
7
8  resultado = (jogadorA - jogadorB) % 3
9
10 if resultado == 1:
11   print("O jogador A ganhou")
12
13
14
15
```

Exercício 5 - Nova Versão

```
1  print("Pedra = 0")
2  print("Papel = 1")
3  print("Tesoura = 2")
4
5  jogadorA = int(input("Digite a primeira escolha: "))
6  jogadorB = int(input("Digite a segunda escolha: "))
7
8  resultado = (jogadorA - jogadorB) % 3
9
10 if resultado == 1:
11   print("O jogador A ganhou")
12 elif resultado == 2:
13   print("O jogador B ganhou")
14
15
```

Exercício 5 - Nova Versão

```python
print("Pedra = 0")
print("Papel = 1")
print("Tesoura = 2")

jogadorA = int(input("Digite a primeira escolha: "))
jogadorB = int(input("Digite a segunda escolha: "))

resultado = (jogadorA - jogadorB) % 3

if resultado == 1:
  print("O jogador A ganhou")
elif resultado == 2:
  print("O jogador B ganhou")
else:
  print("Empate")
```

Variável *flag*

- Podemos usar uma variável para armazenar um estado do programa.
- Por exemplo, podemos criar uma variável para indicar se um sistema está funcionando corretamente (ou se apresentou alguma falha).
- Normalmente inicializamos esta variável com um valor padrão (por exemplo, no caso do exemplo acima, `True`) e atualizamos a variável caso uma mudança de estado ocorra (trocando o valor, por exemplo, para `False`).
- Este tipo de variável, que serve para sinalizar uma situação específica, é chamada de *flag*.
- Uma variável *flag* pode simplificar significativamente a escrita, manutenção e o entendimento de um programa.

Exemplo sem *flag*

```
...


if <condição1>:
  print("Falha do tipo 1")

if <condição2>:
  print("Falha do tipo 2")

if <condição3>:
  print("Falha do tipo 3")


...
```

Exemplo sem *flag*

```
...


if <condição4>:
  print("Falha do tipo 4")

...

if <condição100>:
  print("Falha do tipo 100")


if !<condição1> and !<condição2> and ... and !<condição100>:
  print("Sistema funcionando normalmente")
```

Exemplo com *flag*

```
OK = True


if <condição1>:
  print("Falha do tipo 1")
  OK = False
if <condição2>:
  print("Falha do tipo 2")
  OK = False
if <condição3>:
  print("Falha do tipo 3")
  OK = False

...
```

Exemplo com *flag*

```
...


if <condição4>:
  print("Falha do tipo 4")
  Ok = False
...

if <condição100>:
  print("Falha do tipo 100")
  OK = False

if OK:
  print("Sistema funcionando normalmente")
```

Agradecimentos

- Parte desta aula foi baseada em materiais dos seguintes professores:
  - Eduardo C. Xavier: `https://www.ic.unicamp.br/~eduardo`
  - Marcio M. Pereira: `https://iviarcio.wordpress.com`