

FUNDAMENTOS DE PROGRAMAÇÃO

Estruturas condicionais
Prof. Bruno Góis Mateus



Índice

- Introdução
- Expressões booleanas
- Seletor de único caminho
- Indentação ou recuos
- Seletor de dois caminhos
- Seletor de múltiplos caminhos
- Exercícios
- Operadores lógicos
- Exercícios 2

Estrutura condicional

O que é uma condição



Estrutura condicional

- Segundo o dicionário Michaelis:
- con.di.ção
- sf (lat condicione) 1 Classe social a que pertence uma pessoa. 2 Maneira de viver que resulta das circunstâncias em que cada um se acha. 3 Caráter, gênio, índole. 4 Estado, modo de ser (das coisas). 5 Distinção. 6 Categoria elevada. 7 Alguma coisa estabelecida ou combinada como requisito para que outra coisa seja feita ou entre em vigor; estipulação. 8 Obrigação que se impõe ou se aceita como parte essencial de um acordo; cláusula. 9 Circunstância indispensável para um resultado; requisito. 10 Gram Modificação ou circunstância que se junta a uma oração para indicar a condição de que depende o sentido dela.

EXPRESSÕES BOOLEANAS

O tipo Booleano

- Em **Python**, o tipo de dados utilizado para armazenar valores **verdade** e **falso** é chamado de **bool**
- Existem apenas dois valores possíveis
 - True
 - False

```
print(True)
print(type(True))
print(type(False))
print("True")
print(type("True"))
```

```
True
<type 'bool'>
<type 'bool'>
True
<type 'str'>
```

Expressões booleanas

- Essas expressões fazem um pergunta que resultam em uma resposta, sim ou não
- É uma tipo de expressão, cuja avaliação resulta em um valor booleano
- São compostas por um ou mais operadores relacionais

Operadores relacionais

- Operadores de relacionais **consultam as variáveis mas não as altera**

Operador	Descrição
==	Igualdade
!=	Diferença
<	Menor que
<=	Menor que ou igual a
>	Maior que
>=	Maior que ou igual a

Operadores relacionais

- ==

- Operador de igualdade
- Operador simétrico

- =

- Operador de atribuição

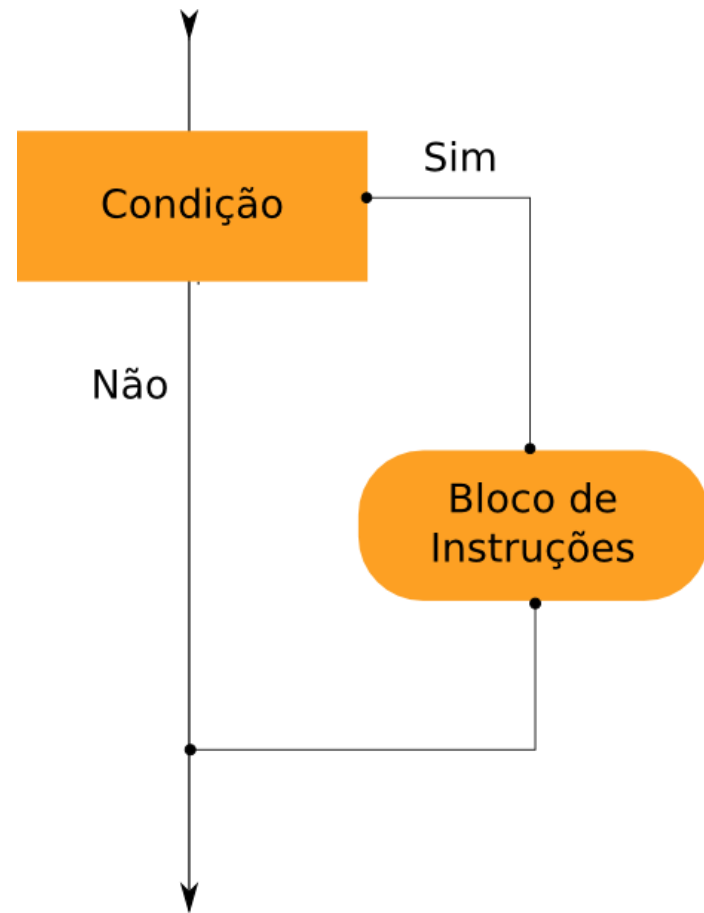
Operadores relacionais

```
x = 5
if x == 5 :
    print 'Igual a 5'          Igual a 5
if x > 4 :
    print 'Maior que 4'       Maior que 4
if x >= 5 :
    print 'Maior que ou Igual a 5'
    Maior que ou Igual a 5
if x < 6 : print 'Menor que 6'  Menor que 6
if x <= 5 :
    print 'Menor que ou Igual a 5'
    Menor que ou Igual a 5
if x != 6 :
    print 'Diferente de 6'     Diferente de 6
```

SELETOR DE CAMINHO ÚNICO

Seletor de caminho único

- Também chamado seletor unário
- Quando a condição é verdadeira
 - Um bloco de instruções é executado
 - Pode possuir uma ou mais sentenças
- Caso contrário
 - A sequência de instruções fora do corpo seletor é executada



Seletor de caminho único

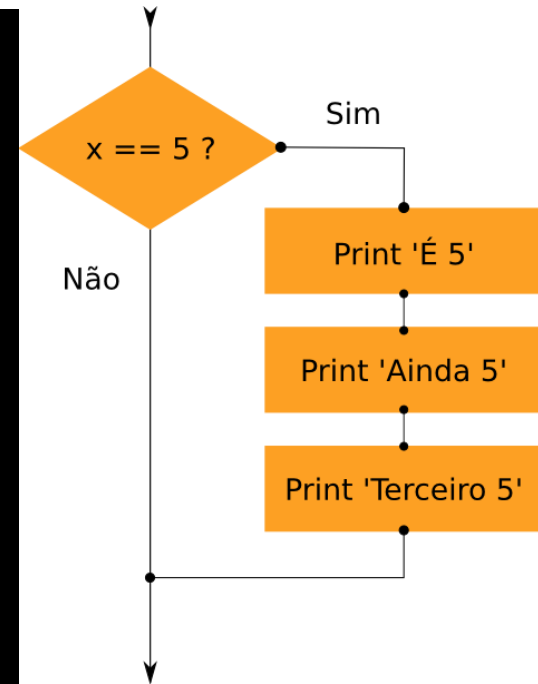
```
x = -10
if x < 0:
    print("O número negativo ", x, " não é válido aqui.")
print("Isso sempre é visualizado")
```

- a) Isso sempre é impresso
- b) O número negativo -10 não é válido aqui
Isso sempre é impresso
- c) O número negativo -10 não é válido aqui

Seletor de caminho único

```
x = 5
print 'Antes é 5'
if x == 5 :
    print 'É 5'
    print 'Ainda é 5'
    print 'Terceiro 5'
print 'Depois do 5'
print 'Antes é 6'
if x == 6 :
    print 'É 6'
    print 'Ainda é 6'
    print 'Terceiro 6'
print 'Depois do 6'
```

Antes é 5
É 5
Ainda é 5
Terceiro 5
Depois do 5
Antes é 6
Depois 6



INDENTAÇÃO OU RECUO

Indentação

- Em Python a indentação tem caráter sintático e semântico
 - Logo, erros sintáticos e semânticos podem ocorrer devido a má indentação
- A indentação altera o escopo das instruções

Indentação

- Quando utilizamos o seletor **if**
 - É necessário **aumentar o recuo** após o **:**
 - **Manter esse recuo** indica quais instruções serão afetadas pelo seletor
 - **Reduzir o recuo** indica o fim desse bloco, fazendo que o nível volte a ser igual ao do seletor
- Linhas em branco são ignoradas
- Comentários também são ignorados

Indentação

• DESLIGUEM
• O TAB!!!

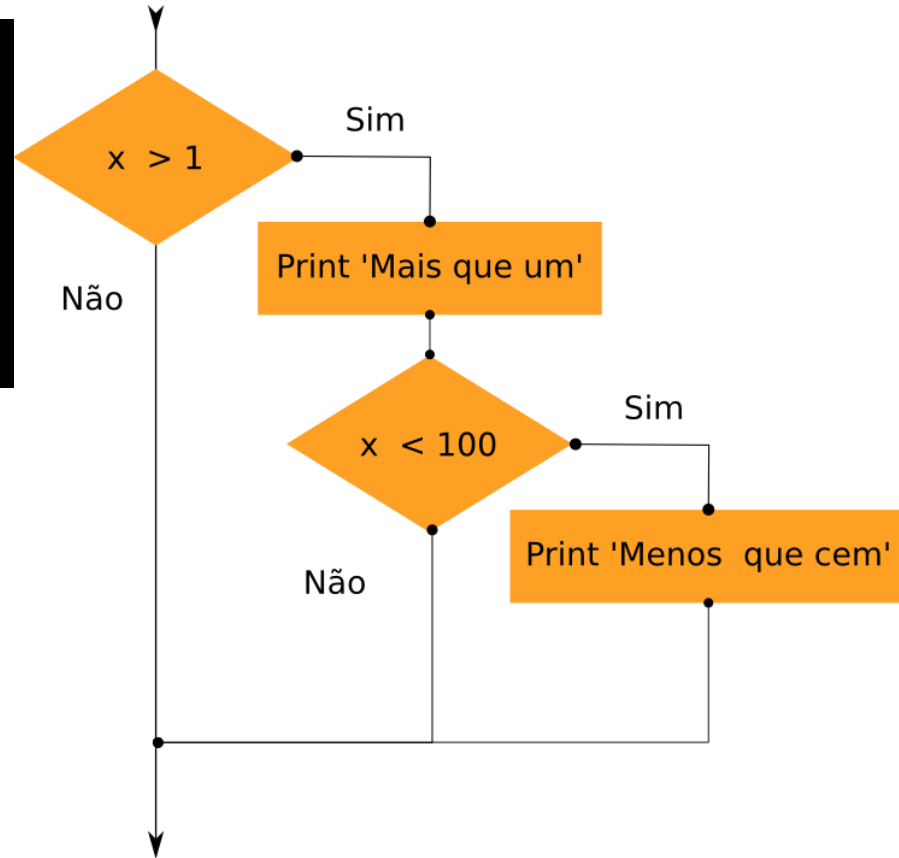
Indentação

```
x = 5
if x > 2 :
    print 'Maior que 2'
    print 'Ainda maior'
print 'Feito com 2'

for i in range(5) :
    print i
    if i > 2 :
        print 'Maior que 2'
    print 'Feito com i', i
print 'Completado'
```

Seletores aninhados

```
x = 42
if x > 1 :
    print 'Mais que um'
    if x < 100 :
        print 'Menos que 100'
print 'Completado'
```



SELETOR DE DOIS CAMINHOS

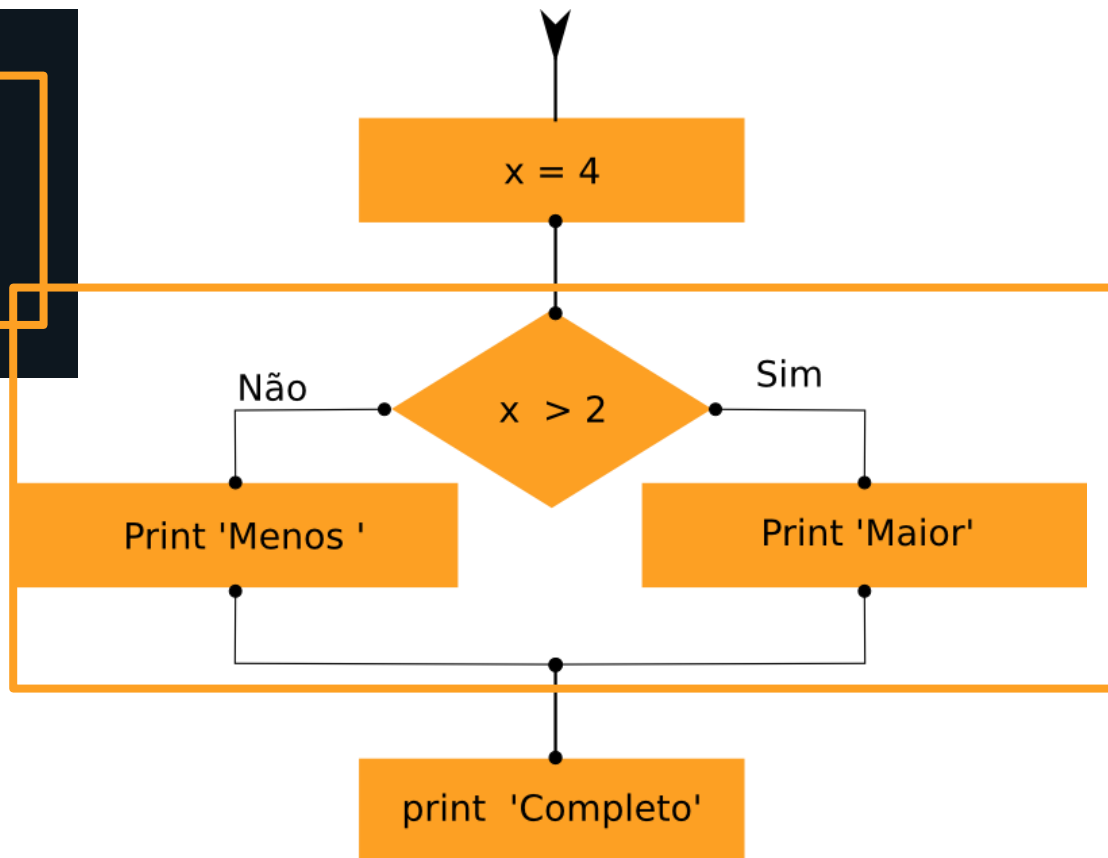
Seletor de dois caminhos

- Às vezes queremos **fazer algo** quando a condição é **verdadeira** e **outra coisa** quando a condição é **falsa**
 - Similar a um entroncamento em uma estrada



Seletor de dois caminhos

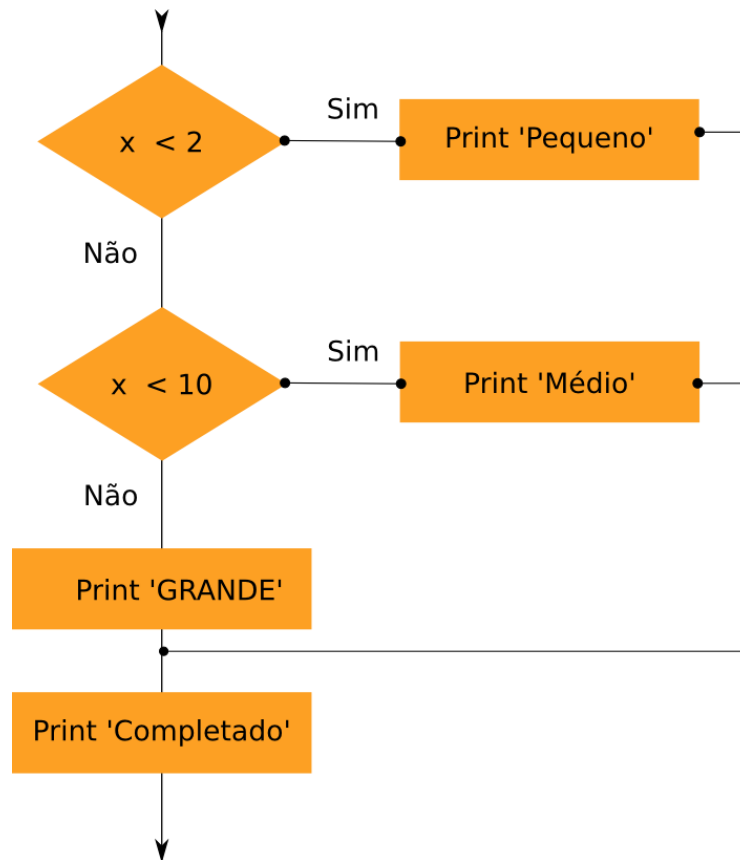
```
x = 4
if x > 2 :
    print 'Maior'
else :
    print 'Menor'
print 'Completo'
```



SELETOR DE MÚLTIPLOS CAMINHOS

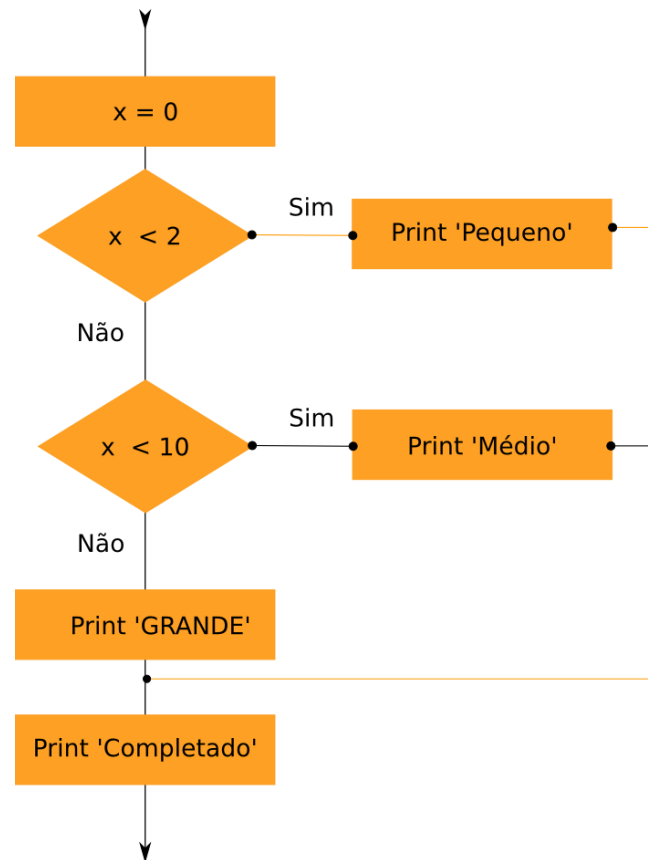
Seletor de múltiplos caminhos

```
if x < 2 :  
    print 'Pequeno'  
elif x < 10 :  
    print 'Médio'  
else :  
    print 'GRANDE'  
print 'Completo'
```



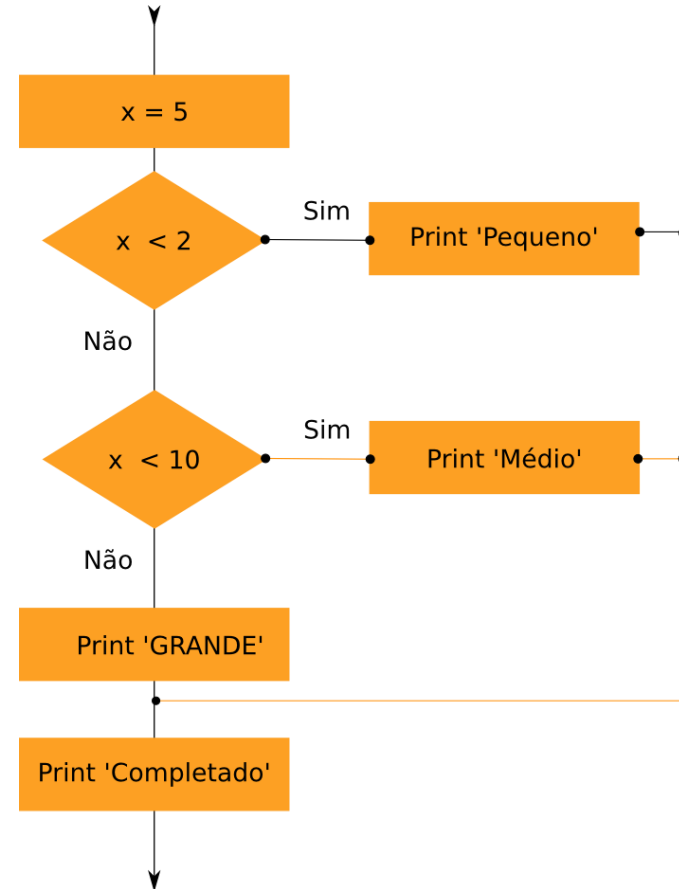
Seletor de múltiplos caminhos

```
x = 0
if x < 2 :
    print 'Pequeno'
elif x < 10 :
    print 'Médio'
else :
    print 'GRANDE'
print 'Completado'
```



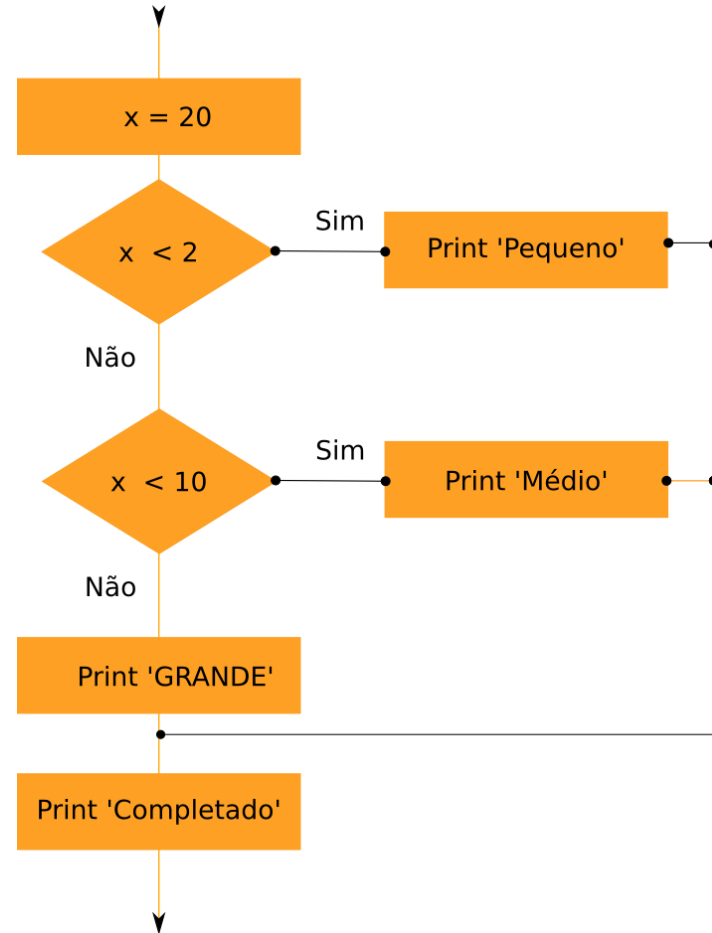
Seletor de múltiplos caminhos

```
x = 5
if x < 2 :
    print 'Pequeno'
elif x < 10 :
    print 'Médio'
else :
    print 'GRANDE'
print 'Completado'
```



Seletor de múltiplos caminhos

```
x = 20
if x < 2 :
    print 'Pequeno'
elif x < 10 :
    print 'Médio'
else :
    print 'GRANDE'
print 'Completo'
```



Seletor de múltiplos caminhos

```
x = 20
if x < 2 :
    print 'Pequeno'
elif x < 10 :
    print 'Médio'

print 'Completado'
```

```
if x < 2 :
    print 'Pequeno'
elif x < 10 :
    print 'Médii'
elif x < 20 :
    print 'Grande'
elif x < 40 :
    print 'Largo'
elif x < 100:
    print 'Enorme'
else :
    print 'Gigantesco!'
```

Teste

- Qual instrução nunca será executada ?

```
if x < 2 :  
    print 'Abaixo de 2'  
elif x >= 2 :  
    print 'Dois ou mais'  
else :  
    print 'Algo mais'
```

```
if x < 2 :  
    print 'Abaixo de 2'  
elif x < 20 :  
    print 'Abaixo de 20'  
elif x < 10 :  
    print 'Abaixo de 10'  
else :  
    print 'Algo mais'
```

EXERCÍCIOS

Hora prática

- Escreva um programa que peça as seguintes informações ao usuário:
 - Salário
 - Carga horária semanal
 - Quantidade de horas trabalhada no mês
- O programa deve mostrar na tela quanto deve ser pago ao funcionário ao final do mês.
 - O valor da hora extra, é igual 1.5 do valor da hora normal
 - A saída deve ter o seguinte formato:
 - O funcionário deve receber 1500.00 reais, sendo 500 referente a horas extras



Hora da prática

- Escreva um programa que determine se um número é par ou ímpar.
 - O número deve ser informado pelo usuário

Hora prática

- Condign game – Primeiro problema

Hora da prática

- Problema: Algoritmo que determina o vencedor de uma partida de par ou ímpar



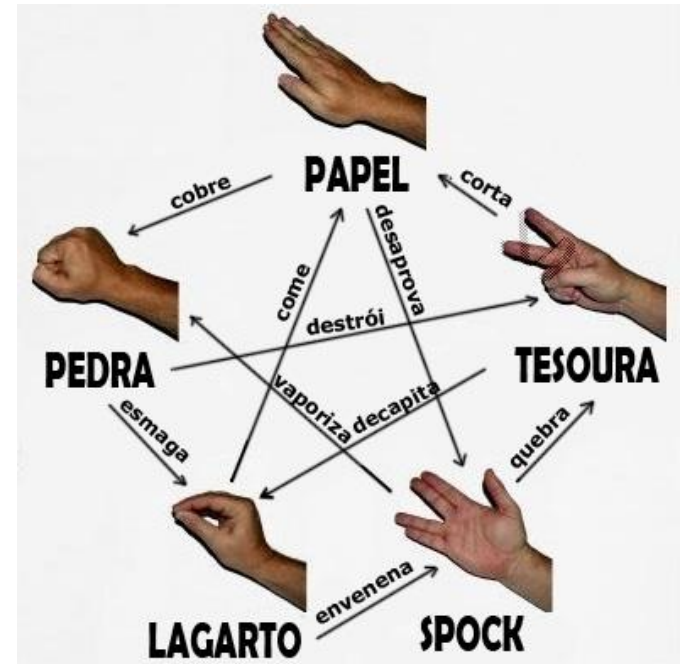
Hora da prática

- Problema: Algoritmo que determina o vencedor de uma pedra, papel e tesoura. SEM OPERADORES LÓGICOS
- Regras:
 - Pedra ganha de tesoura
 - Tesoura ganha de papel
 - Papel ganha de pedra



Hora da prática

- Problema: Algoritmo que determina o vencedor de uma pedra, papel, tesoura, lagarto e spock. SEM OPERADORES LÓGICOS
- Regras:
 - Tesoura corta papel
 - Papel cobre pedra
 - Pedra esmaga lagarto
 - Lagarto envenena Spock
 - Spock esmaga (ou derrete) tesoura
 - Tesoura decapita lagarto
 - Lagarto come papel
 - Papel refuta Spock
 - Spock vaporiza pedra
 - Pedra quebra tesoura



Hora da prática

- Problema: Algoritmo que determina o signo do usuário dado a data de nascimento - SEM OPERADORES LÓGICOS
- Regras:
 - Signo de Aquário (21/1 - 19/2)
 - Signo de Peixes (20/2 - 20/3)
 - Signo de Áries (21/3 - 20/4)
 - Signo de Touro (21/4 - 21/5)
 - Signo de Gêmeos (22/5 - 21/6)
 - Signo de Câncer (21/6 - 23/7)
 - Signo de Leio (24/7 - 23/8)
 - Signo de Virgem (24/8 - 23/9)
 - Signo de Libra (24/9 - 23/10)
 - Signo de Escorpião (24/10 - 22/11)
 - Signo de Sagitário (23/11 - 21/12)
 - Signo de Capricórnio (22/12 - 20/1)



OPERADORES LÓGICOS

Operadores lógicos

- and(e), or(ou) e not(não)
 - Tem o mesmo valor semântico no nosso idioma

Operadores lógicos

- and

```
Expressao1 and Expressao2
```

Tabela verdade

Expressão 1	Expressão 2	Expressão 1 and Expressão 2
true	true	true
true	false	false
false	true	false
false	false	false

Operadores lógicos

- and

- Quando essa condição será verdadeira ?
- Como você descreveria em palavras ?

```
x > 0 and x < 10
```

Tabela verdade

x > 0	x < 10	x > 0 and x < 10
true	true	true
true	false	false
false	true	false
false	false	false

Operadores lógicos

- or

```
Expressao1 or Expressao2
```

Tabela verdade

Expressão 1	Expressão 2	Expressão 1 or Expressão 2
true	true	true
true	false	true
false	true	true
false	false	false

Operadores lógicos

- or
 - Quando essa condição será verdadeira ?
 - Como você descreveria em palavras ?

```
x > 0 or x < 10
```

Tabela verdade

x > 0	x < 10	x > 0 or x < 10
true	true	true
true	false	true
false	true	true
false	false	false

Operadores lógicos

- Or
 - Condição: x ser maior do que 5, 7 ou 8

```
x == 5 or 8
```



```
x == 5 or x == 6 or x == 8
```

Operadores lógicos

- not

```
not Expressao1
```

Tabela verdade

Expressão 1	Not expressão 1
true	false
false	true

Operadores lógicos

- not

```
not x > 5
```

Tabela verdade

X > 5	Not x > 5
true	false
false	true

EXERCÍCIOS

Hora da prática

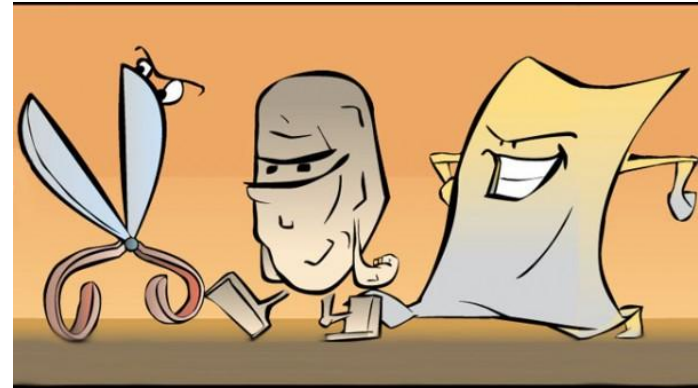
- Qual o valor das seguintes expressões ?
 - $3 == 3$
 - $3 != 3$
 - $3 >= 4$
 - $\text{not } (3 < 4)$

Hora da prática

- Escreva o oposto das seguintes expressões sem utilizar o operador not
 - $a > b$
 - $a \geq b$
 - $a \geq 18$ and $day == 3$
 - $a \geq 18$ or $day != 3$

Hora da prática

- Refazer o pedra, papel e tesoura utilizando os OPERADORES LÓGICOS
- Regras:
 - Pedra ganha de tesoura
 - Tesoura ganha de papel
 - Papel ganha de pedra



Hora da prática

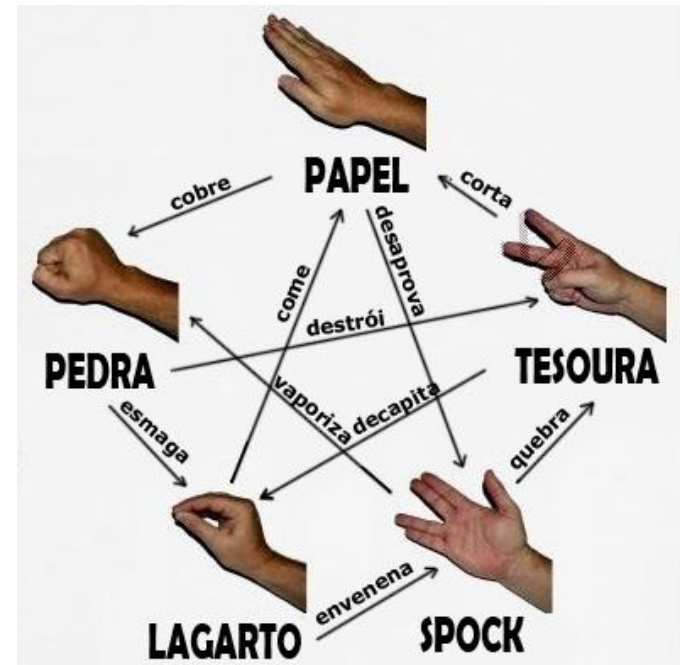
- Escreva um programa que pergunte as notas de um aluno em três provas e determine se ele foi aprovado ou não.
- Para ser aprovado o aluno deve alcançar média 7
- Caso contrário ele deve realizar uma prova final
 - Nesse caso, para ser aprovado o aluno deve ter a média final($(\text{média da três provas} + \text{prova final})/2$) maior do que 5
 - Caso o aluno tire menos que 4 na prova final ele é reprovado

Hora da prática

- Escreva um programa que pergunte a média final do aluno e converta a nota numérica em conceitos seguindo as seguintes regras
 - ≥ 90 -> conceito A
 - [80-90) -> conceito B
 - [70-80) -> conceito C
 - [60-70) -> conceito D
 - < 60 -> conceito F

Hora da prática

- Refazer o pedra, papel, tesoura, lagarto e spock. Utilizando os OPERADORES LÓGICOS
- Regras:
 - Tesoura corta papel
 - Papel cobre pedra
 - Pedra esmaga lagarto
 - Lagarto envenena Spock
 - Spock esmaga (ou derrete) tesoura
 - Tesoura decapita lagarto
 - Lagarto come papel
 - Papel refuta Spock
 - Spock vaporiza pedra
 - Pedra quebra tesoura



Hora da prática

- Refazer o algoritmo que determina o signo do usuário dado a data de nascimento utilizando os OPERADORES LÓGICOS
- Regras:
 - Signo de Aquário (21/1 - 19/2)
 - Signo de Peixes (20/2 - 20/3)
 - Signo de Áries (21/3 - 20/4)
 - Signo de Touro (21/4 - 21/5)
 - Signo de Gêmeos (22/5 - 21/6)
 - Signo de Câncer (21/6 - 23/7)
 - Signo de Leio (24/7 - 23/8)
 - Signo de Virgem (24/8 - 23/9)
 - Signo de Libra (24/9 - 23/10)
 - Signo de Escorpião (24/10 - 22/11)
 - Signo de Sagitário (23/11 - 21/12)
 - Signo de Capricórnio (22/12 - 20/1)



A seguir

- Estruturas de repetição