



# CURSO DE CIÊNCIAS DA COMPUTAÇÃO

## PROGRAMAÇÃO DE COMPUTADORES



TURMA: P1/P4



AULA 05

PROFESSOR: JEOFTON COSTA

# Duvidas da Aula Anterior?





## Objetivos

- Operadores Lógicos;
- Instruções if/elif/else;
- Instruções match/case.





## E se a vida fosse um videogame?

Imagine que você está jogando seu RPG favorito. De repente, você se depara com uma encruzilhada:

- A) Seguir pela estrada principal;
- B) Esperar outros jogadores, e então investigar a trilha misteriosa;
- C) Investigar, sozinho, a trilha misteriosa na floresta.

A escolha é sua! Mas cuidado, cada caminho te leva a um destino diferente.

Na estrada principal, você segue em segurança para a próxima cidade. Mas se escolher a trilha na floresta, prepare-se para enfrentar desafios emocionantes e recompensas incríveis.

Assim como em um jogo, a lógica de programação é feita de escolhas. Usando estruturas condicionais, você decide quais comandos serão executados, criando programas interativos e dinâmicos.

Comando **elif**





```
1 """
2 Escreva um programa que verificando a média de duas notas de um(a) determinado(a)
3 aluno(a), verifica se ele está aprovado, quando obtiver média maior ou igual 7.0,
4 na final, se a média for menor que 7.0 e maior ou igual 4.0, e reprovado, se a média
5 for menor que 4.0
6 """
```

**ATENÇÃO:** as regras de indentação também devem ser observadas aqui.

```
7
8 nota1 = float(input("Informe a primeira nota:"))
9 nota2 = float(input("Informe a segunda nota:"))
10 media = (nota1+nota2)/2
11
12 if media >= 7.0:
13     print(f"Parabéns, você está aprovado, com média {media}.")
14 elif media < 7.0 and media >= 4.0:
15     print("Você está na final. Sua média foi", media, ". ESTUDE!")
16 else:
17     print(f"REPROVADO! Média {media}.")
18 print("*-*-*-*-*")
19 print("Fim do programa.")
```

```
1 """
2 Escreva um programa que verificando a média de duas notas de um(a) determinado(a)
3 aluno(a), verifica se ele está aprovado, quando obtiver média maior ou igual 7.0,
4 na final, se a média for menor que 7.0 e maior ou igual 4.0, e reprovado, se a média
5 for menor que 4.0
6 """
```

Assim, como no comando **if**,  
também precisamos definir  
uma expressão booleana  
para o comando **elif**

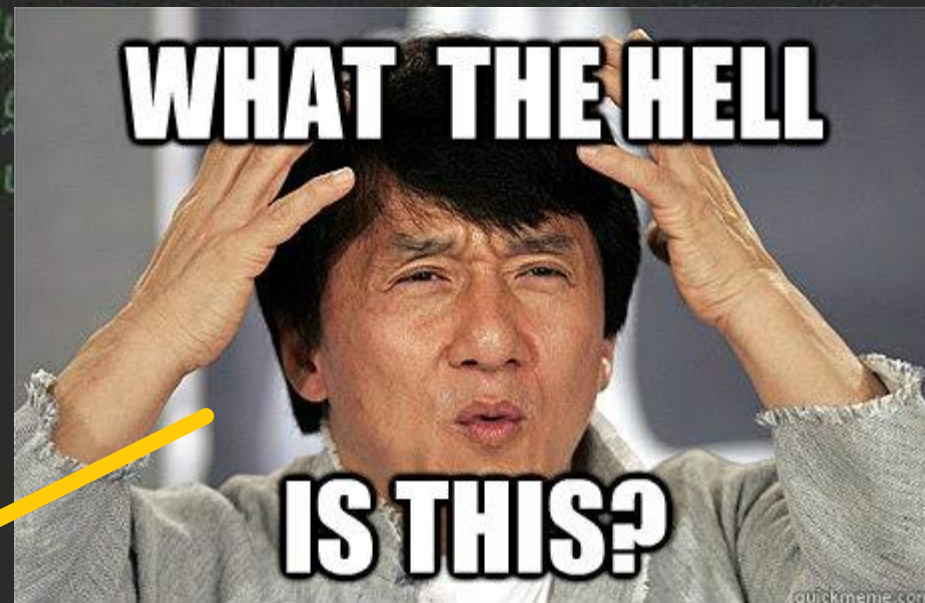
```
8 nota1 = float(input("Informe a primeira nota:"))
9 nota2 = float(input("Informe a segunda nota:"))
10 media = (nota1+nota2)/2
11
12 if media >= 7.0:
13     print(f"Parabéns, você está aprovado, com média {media}.")
14 elif media < 7.0 and media >= 4.0:
15     print("Você está na final. Sua média foi", media, ". ESTUDE!")
16 else:
17     print(f"REPROVADO! Média {media}.")
18 print("*-*-*-*-*")
19 print("Fim do programa.")
```



```
1 """
2 Escreva um programa que verificando a média de de
3 aluno(a), verifica se ele está aprovado, quando
4 na final, se a média for menor que 7.0 e maior ou
5 for menor que 4.0
6 """
```

```
7
8 nota1 = float(input("Informe a primeira nota:"))
9 nota2 = float(input("Informe a segunda nota:"))
10 media = (nota1+nota2)/2
```

```
11
12 if media >= 7.0:
13     print(f"Parabéns, você está aprovado, com média {media}.")
14 elif media < 7.0 and media >= 4.0:
15     print("Você está na final. Sua média foi", media, ". ESTUDE!")
16 else:
17     print(f"REPROVADO! Média {media}.")
18 print("*-*-*-*-*")
19 print("Fim do programa.")
```





```
1 """
2 Escreva um programa que verificando a média de duas notas de um(a) determinado(a)
3 aluno(a), verifica se ele está aprovado, quando obtiver média maior ou igual 7.0),
4 na final, se a média for menor que 7.0 e maior ou igual 4.0, e reprovado, se a média
5 for menor que 4.0
6 """
```

```
7
8 nota1 = float(input("Informe a primeira nota: "))
9 nota2 = float(input("Informe a segunda nota: "))
10 media = (nota1+nota2)/2
```

```
11
12 if media >= 7.0:
13     print(f"Parabéns, você está aprovado, com média {media}.")
14 elif media < 7.0 and media >= 4.0:
15     print("Você está na final. Sua média foi", media, ". ESTUDE!")
16 else:
17     print(f"REPROVADO! Média {media}.")
18 print("*-*-*-*-*")
19 print("Fim do programa.")
```

Calma, Xovem! Isto é  
apenas o operador lógico  
'e', que em python se  
escreve **and**



# Operadores Lógicos

Operador	Função
<b>and</b>	lógico E
<b>or</b>	lógico OU
<b>not</b>	lógico de negação



# Tabela Verdade (Python)

## and (&)

Entrada 1	Entrada 2	Saída
V	V	V
V	F	F
F	V	F
F	F	F

## or (|)

Entrada 1	Entrada 2	Saída
V	V	V
V	F	V
F	V	V
F	F	F

## not

Entrada	Saída
V	F
F	V

# Operadores Matemáticos em Python

Python	Operação	Exemplo	Resultado
+	Somar	$2 + 3$	5
-	Subtrair	$5 - 4$	1
*	Multiplicar	$3 * 4$	12
/	Dividir	$11 / 2$	5.5
//	Quociente da divisão inteira	$11 // 2$	5
%	Resto da divisão inteira	$11 \% 2$	1
**	Exponenciação	$3 ** 2$	9



# Construindo de condições booleanas

Operador	Comparação	Exemplo	Resultado
==	Igual	“senha” == “senha”	Verdadeiro
>	Maior que	9 > 21	Falso
>=	Maior ou igual	7 >= 7	Verdadeiro
<	Menor que	001 < 010	Verdadeiro
<=	Menor ou igual	342 <= 231	Falso
!=	Diferente	“Lindinhos” != “lindinhos”	Verdadeiro

# Operadores Lógicos

Operador	Função
and	lógico E
or	lógico OU
not	lógico de negação



# Exercitando

Escreva um programa que solicita ao usuário um número inteiro e que se ele é divisível por 2, 3, 5, 6 ou 7, caso contrário, informe que o número não é divisível por nenhum destes valores.

Escreva um programa que calculando a média de duas notas de um(a) determinado(a) aluno(a), verifica se ele está aprovado, quando obtiver média maior ou igual 7.0, na final, se a média for menor que 7.0 e maior ou igual 4.0, e reprovado, se a média for menor que 4.0.



# Exercitando

Escreva um programa que solicita ao usuário que digite um número de 1 a 7. Após informe qual o dia da semana (1 para domingo, 2 para segunda, 3 para terça, sucessivamente, até chegar no 7, para o sábado). Se o usuário digitar qualquer outro valor numérico, o programa deverá informar uma mensagem de erro.

---

Desenvolva um programa que solicita ao usuário um número e verifica se ele é positivo, negativo ou igual a zero. O programa deve imprimir uma mensagem correspondente à condição do número.

---

Escreva um programa que pede ao usuário para inserir seu peso (em kg) e sua altura (em metros). Com base nesses valores, o programa deve calcular o IMC do usuário e informar em qual categoria ele se encontra de acordo com a tabela abaixo:

IMC abaixo de 18.5: Abaixo do peso

IMC entre 18.5 e 24.9: Peso normal

IMC entre 25 e 29.9: Sobrepeso

IMC entre 30 e 34.9: Obesidade grau 1

IMC entre 35 e 39.9: Obesidade grau 2

IMC acima de 40: Obesidade grau 3



# Exercitando

Crie um programa que solicita ao usuário o valor total de uma compra. Em seguida, o programa deve calcular o valor do desconto com base no valor total da compra, de acordo com a seguinte tabela:

- Se o valor total for maior que R\$ 100, aplique um desconto de 10%.
- Se o valor total estiver entre R\$ 50 e R\$ 100, aplique um desconto de 5%.
- Se o valor total for menor ou igual a R\$ 50, não aplique nenhum desconto.

O programa deve então imprimir o valor total da compra após o desconto ser aplicado.

---

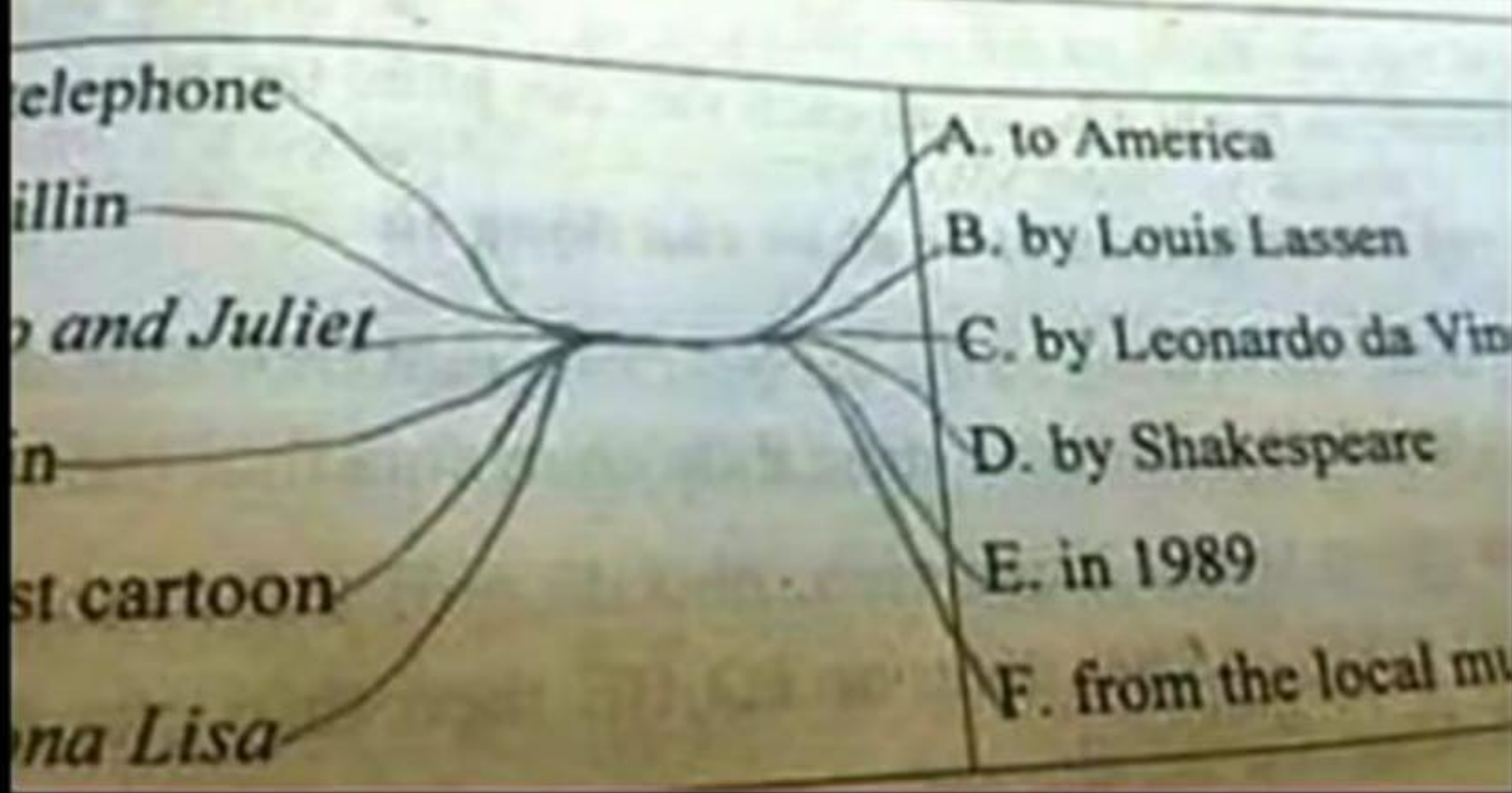
Crie um programa que solicita ao usuário sua idade e, com base nesse valor, informa em qual faixa etária ele se encontra, de acordo com as seguintes categorias:

- Até 12 anos: Criança
- De 13 a 19 anos: Adolescente
- De 20 a 59 anos: Adulto
- 60 anos ou mais: Idoso

# Exercitando

Crie um programa que pede ao usuário para inserir três comprimentos de lado e verifica se esses comprimentos podem formar um triângulo. Se os comprimentos puderem formar um triângulo, o programa deve exibir uma mensagem indicando o tipo de triângulo (equilátero, isósceles ou escaleno); caso contrário, deve informar que não é possível formar um triângulo com esses comprimentos.





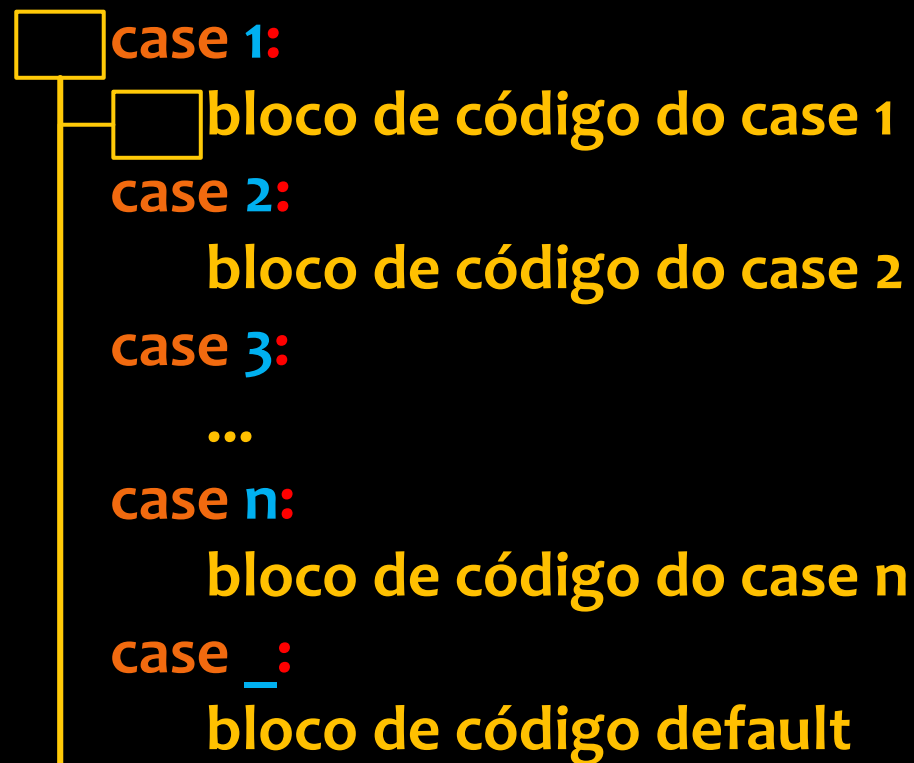
**ESTRUTURA  
MATCH/CASE**

**Give this kid a  
medal!**



A partir da versão 3.10, a linguagem python “ganhou” a estrutura match/case, a qual permite implementar algoritmos com múltiplas escolhas:


**match variável\_de\_controle:**



```
match variável_de_controle:
    case 1:
        bloco de código do case 1
    case 2:
        bloco de código do case 2
    case 3:
        ...
    case n:
        bloco de código do case n
    case _:
        bloco de código default
```

The diagram illustrates the match/case structure with indentation levels. A box is placed next to the 'match' keyword. A line connects this box to another box placed next to the first 'case' statement. This pattern repeats for subsequent 'case' statements, showing how the indentation level increases for each case block. The 'case \_:' block represents the default case.

Aqui também devem ser observadas as regras de indentação. Tanto na declaração do comando **case** em relação do comando **match**, como do **bloco de código** em relação ao comando **case**.



An arrow points from the text box to the 'case \_:' block in the code structure, indicating that the indentation rules mentioned in the text apply to this block as well.



A partir da versão 3.10, a linguagem python “ganhou” a estrutura match/case, a qual permite implementar algoritmos com múltiplas escolhas:

**match variável\_de\_controle:**

**case 1:**

bloco de código do case 1

**case 2:**

bloco de código do case 2

**case 3:**

...

**case n:**

bloco de código do case n

**case \_:**

bloco de código default

Esta é uma prática (convenção) utilizada por programadores python para representar os valores default, ou seja, se nenhuma das entradas esperadas for inserida, esta sentença case exibirá/retornará uma mensagem padrão.

# Demonstrando o match/case

```
1 """
2 Utilizando a estrutura match/case, escreva um programa que solicita ao usuário que digite um número de 1 a 7.
3 Após informe qual o dia da semana (1 para domingo, 2 para segunda, 3 para terça, sucessivamente, até chegar
4 no 7, para o sábado). Se o usuário digitar qualquer outro valor numérico, o programa deverá informar uma
5 mensagem de erro.
6 """
7
8 numero_digitado = int(input("Digite um número inteiro de 1 a 7: (Depois tecla enter)"))
9
10 match numero_digitado:
11     case 1:
12         print(f"O número digitado foi {numero_digitado}, que corresponde ao domingo.")
13     case 2:
14         print(f"O número digitado foi {numero_digitado}, que corresponde à segunda-feira.")
15     case 3:
16         print(f"O número digitado foi {numero_digitado}, que corresponde à terça-feira.")
17     case 4:
18         print(f"O número digitado foi {numero_digitado}, que corresponde à quarta-feira.")
19     case 5:
20         print(f"O número digitado foi {numero_digitado}, que corresponde à quinta-feira.")
21     case 6:
22         print(f"O número digitado foi {numero_digitado}, que corresponde à sexta-feira.")
23     case 7:
24         print(f"O número digitado foi {numero_digitado}, que corresponde ao sábado.")
25     case _:
26         print(f"O número digitado foi {numero_digitado}, o qual não tem dia da semana correspondente.")
27
28 print("Fim do programa")
```



# Exercitando

Implemente uma função em Python que receba uma letra como entrada e retorne "Vogal" se a letra for uma vogal (a, e, i, o, u), "Consoante" se for uma consoante ou "Caractere inválido" se não for uma letra.

---

Crie uma função em Python que receba o código de um produto (1 para Produto A, 2 para Produto B, etc.) e retorne o preço correspondente. Considere os seguintes preços: Produto A - R\$ 10, Produto B - R\$ 20, Produto C - R\$ 30



Escreva que recebendo um número de 1 a 12, informe a quantidade de dias do mês. Adote 1 será relativo ao mês de janeiro, 2 o mês de fevereiro e assim sucessivamente.

---

Implemente um calculadora em python que, capturando o número digitado pelo usuário e a operação matemática escolhida, retorne a respectiva tabuada.





## USO DE IMAGENS DE PERSONAGENS DE QUADRINHOS PARA FINS ESTRITAMENTE ACADÊMICOS

"As imagens de personagens de quadrinhos utilizadas neste material têm o propósito exclusivo de enriquecer o conteúdo acadêmico da apresentação. O uso está em conformidade com as leis de direitos autorais, pois se enquadra no contexto educacional. Respeitamos os direitos dos detentores dessas propriedades intelectuais."