



Programação de Computadores

- ✓ Estrutura de decisão aninhadas
- ✓ Exemplos de aplicação

Na aula passada...

Tomadas de decisão são importantes na programação, na atividade profissional, na vida...



www.peytonbolin.com



<https://www.sbcoaching.com.br/>



<https://consultormarketing.digital/>

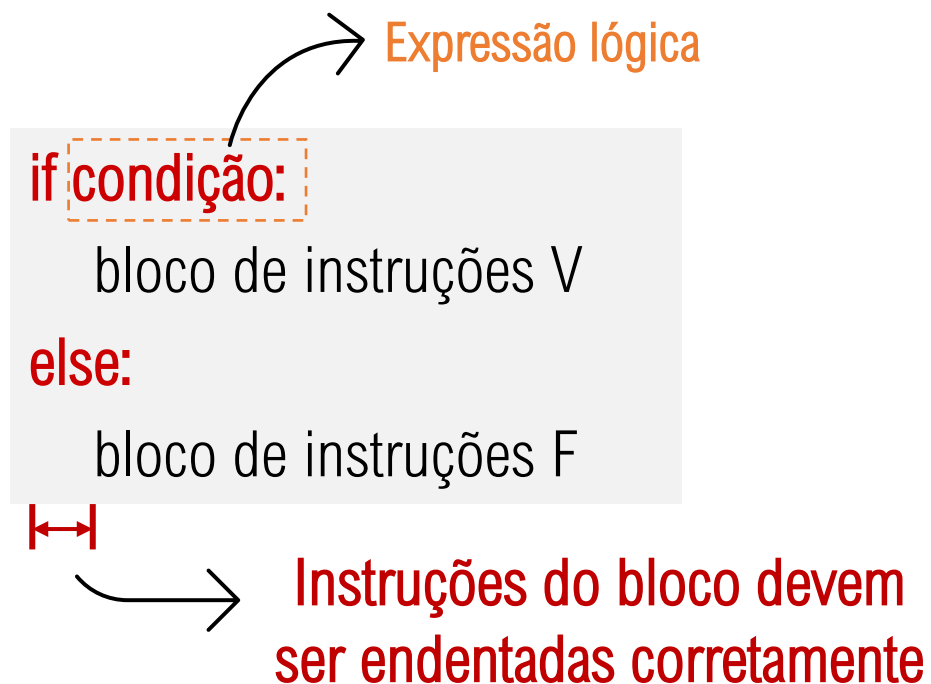


Na programação, as decisões devem estar bem definidas: não podem ser ambíguas.

Na aula passada...

Estrutura Condicional Composta:

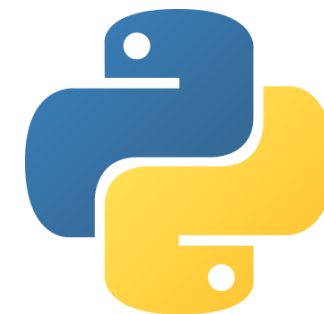
Nela é avaliada uma expressão lógica e se o resultado da avaliação for verdadeiro (true) será executado o primeiro bloco de código, caso contrário o bloco do senão.



Na aula passada...

Crie um programa em Python que solicite ao usuário a sua idade e mostre se o mesmo pode ou não ter CNH.

```
main.py
1  # Exemplo 04: programa que solicita a idade
2  # do usuário e mostra se ele pode ter CNH
3
4  idade = int(input('Digite a sua idade: '))
5
6  ▼ if idade >=18:
7      print('Parabéns! Você pode ter CNH')
8  ▼ else:
9      print('Você NÃO pode ter CNH')
```



Na aula passada...

Operadores ternários:

- ✓ São todos os operadores que podem receber três operandos.
- ✓ Como as expressões condicionais costumam ser os operadores ternários mais populares nas linguagens em que aparecem, acabamos por associar estes nomes e considerar que são a mesma coisa.
- ✓ No Python, a estrutura de uma expressão condicional ternária é algo bem simples:

```
paz = True if vencer_o_thanos else False  
resposta = "Par" if x % 2 == 0 else "ímpar"
```

Na aula passada...

Normalmente, quando executamos algum código no VSCode, o terminal fica “poluído”.



se
Liga
Aí

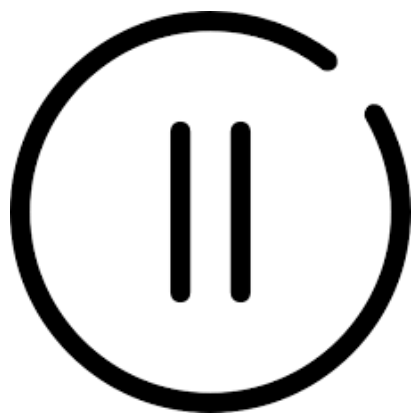
Podemos utilizar a biblioteca `os` para limpar o terminal e exibir somente as informações que fazem parte da execução do programa.



```
limparterminal.py > ...
1 import os
2 os.system('cls' if os.name == 'nt' else 'clear')
3
4 nome = input('Digite o seu nome: ')
5 print(f'Olá {nome}!!Este é um exemplo de terminal limpo!!!')
6 input()
```

Antes de continuarmos...

Pausa para reflexão!!!



antes de continuar...

Dúvidas quanto aos exercícios da última aula?



Conceitos abordados nesta aula

A proposta desta aula é apresentar estruturas utilizadas quando várias condições precisam ser testadas.



<https://vidaprogramador.com.br/>



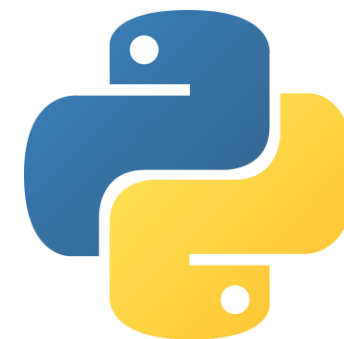
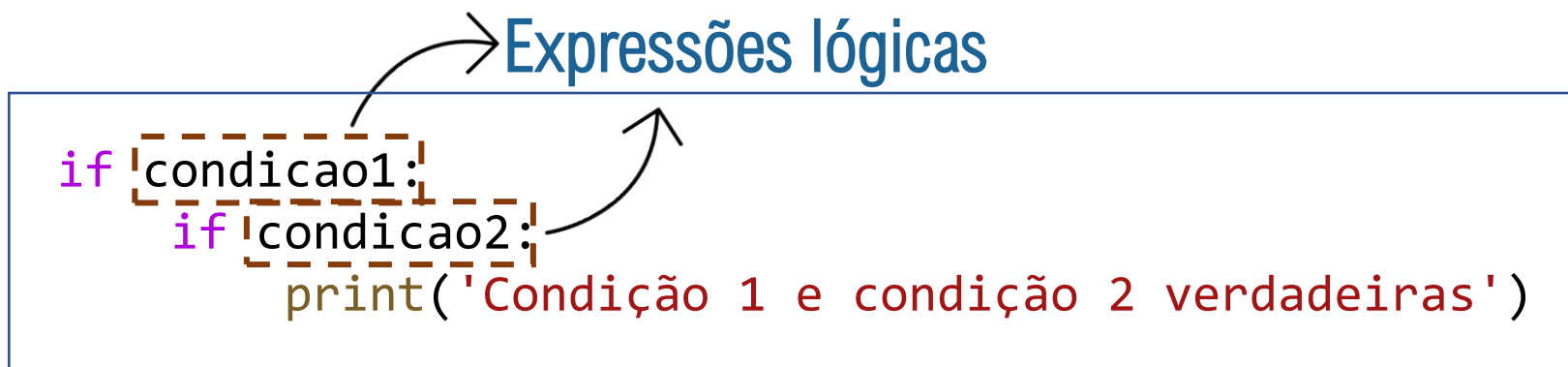
Estruturas condicionais aninhadas

Estruturas condicionais podem ser postas uma dentro da outra de forma a garantir uma melhor maneira de executar os comandos → estruturas aninhadas.

Estrutura condicional aninhada

Expressões lógicas

```
if condicao1:  
    if condicao2:  
        print('Condição 1 e condição 2 verdadeiras')
```



Estruturas condicionais aninhadas

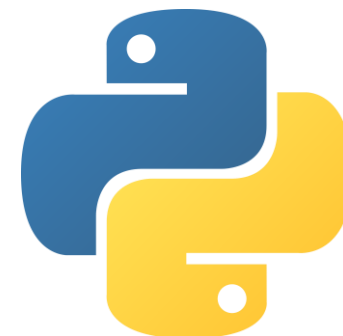
A estrutura aninhada pressupõem uma correspondência de operadores lógicos compostos:

```
main.py
1 if condicao1:
2     if condicao2:
3         print("As condições 1 e 2 são verdadeiras!")
```

Corresponde a:

```
main.py
1 if condicao1 and condicao2:
2     print("As condições 1 e 2 são verdadeiras!")
3
```

Condicional
simples



Estrutura if-else aninhada - Python

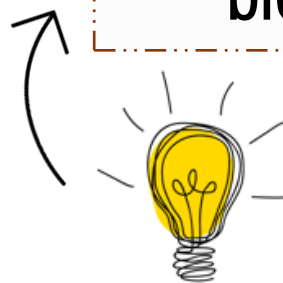
Usada quando várias condições devem ser testadas.

Sintaxe:

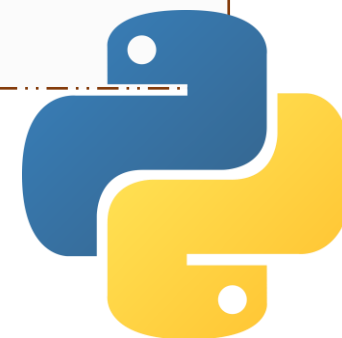
```
if condição 1:  
    bloco de instruções 1  
else:  
    if condição 2:  
        bloco de instruções 2  
    else:  
        bloco de instruções 3
```

ou

```
if condição 1:  
    bloco de instruções 1  
elif condição 2:  
    bloco de instruções 2  
else:  
    bloco de instruções 3
```



Mais fácil!!!!



Exemplos

1- Escreva um programa em Python que solicite ao usuário a média de um aluno e o percentual de frequência e mostre a sua situação, conforme a tabela abaixo:

Frequência	Media	Situação
< 75		Reprovado por falta
>= 75	< 6	Reprovado por nota
	>= 6	Aprovado

Exemplo 1 – resolução incorreta!

```
algoritmo resultadoFinal
inicio
    real m, f
    escreva("Digite a média")
    leia(m)
    escreva("Digite o percentual de frequência")
    leia(f)
    se (f < 75) {
        escreva("Reprovado por falta")
    }
    se (m < 6) {
        escreva("Reprovado por nota")
    }
    senão{
        escreva("Aprovado")
    }
fim
```

O que será impresso se a média do aluno for igual a 7,0 e a frequência for 60?



Exemplo 1 – resolução correta!

```
algoritmo resultadoFinal
inicio
    real m, f
    escreva("Digite a média")
    leia(m)
    escreva("Digite o percentual de frequência")
    leia(f)
    se (f < 75) {
        escreva("Reprovado por falta")
    }
    senão se (m < 6) {
        escreva("Reprovado por nota")
    }
    senão{
        escreva("Aprovado")
    }
fim
```

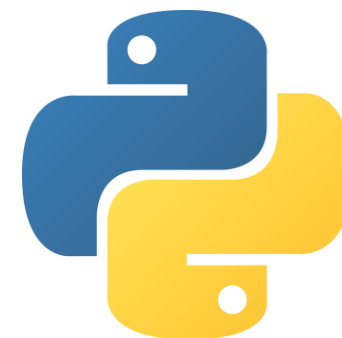
O que será impresso se a média do aluno for igual a 7,0 e a frequência for 60?



Exemplo 1 – Python (if-else-if)

```
media = float(input('Digite a média do aluno: '))
freq = float(input('Digite o percentual de frequência: '))

if freq < 75:
    print('Reprovado por faltas!')
else:
    if media < 6:
        print('Reprovado por nota!!!')
    else:
        print('Aprovado')
```



Exemplo 1 – Python(if-elif-else)

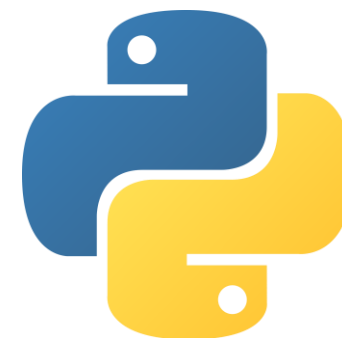
```
media = float(input('Digite a média do aluno: '))
freq = float(input('Digite o percentual de frequência: '))

if freq < 75:
    print('Reprovado por faltas!')
elif media < 6:
    print('Reprovado por nota!!!')
else:
    print('Aprovado')
```



se
Liga
Aí

Utilizar o **elif** deixa o código mais organizado, além de implementar uma solução mais simples!!!



Exemplos

2- Faça um programa em Python que solicite um código referente ao tipo da diária de hospedagem e também a quantidade de diárias desejada por um cliente. Calcule e mostre, usando estrutura condicional aninhada, o valor total a pagar pelo cliente, conforme a tabela abaixo:

Tipo da diária	Quarto	Valor da diária
S	Simples	R\$ 255,50
D	Duplo	R\$ 305,50
T	Triplo	R\$ 360,50

Caso seja digitado um código diferente dos apresentados na tabela acima deve ser mostrada a mensagem “Tipo de diária inválido”.

Exemplo 2

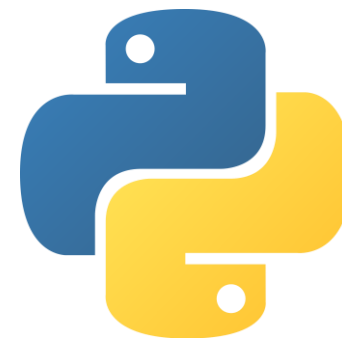
```
algoritmo hospedagem
inicio
    inteiro qtdeDiarias
    real valor
    caracter tipo
    escreva("Digite a quantidade de diárias")
    leia(qtdeDiarias)
    escreva("Digite o tipo de hospedagem")
    leia(tipo)
    se tipo=='s' ou tipo=='S'
    |   escreva ("Valor a pagar", (qtdeDiarias*255.5))
senão
    |   se tipo=='d' ou tipo=='D'
    |   |   escreva ("Valor a pagar", (qtdeDiarias*305.5))
senão
    |   se tipo=='t' ou tipo=='T'
    |   |   escreva ("Valor a pagar", (qtdeDiarias*360.5))
senão
    |   escreva("Tipo de hospedagem inválida")
fim
```



Exemplo 2

```
qDiarias = int(input('Digite a quantidade de diárias: '))
tipo = input('Digite o tipo de hospedagem (s|d|t): ')

if tipo == 's' or tipo == 'S':
    print(f'Valor a pagar R$ {qDiarias * 255.5}')
elif tipo == 'd' or tipo == 'D':
    print(f'Valor a pagar R$ {qDiarias * 305.5}')
elif tipo == 't' or tipo == 'T':
    print(f'Valor a pagar R$ {qDiarias * 360.5}')
else:
    print('Tipo de hospedagem inválida!!!!')
```



Exemplo 2 – Uso do operador **in**

```
qDiarias = int(input('Digite a quantidade de diárias: '))
tipo = input('Digite o tipo de hospedagem (s|d|t): ')

if tipo in 'sS':
    print(f'Valor a pagar R$ {qDiarias * 255.5}')
elif tipo in 'dD':
    print(f'Valor a pagar R$ {qDiarias * 305.5}')
elif tipo in 'tT':
    print(f'Valor a pagar R$ {qDiarias * 360.5}')
else:
    print('Tipo de hospedagem inválida!!!!')
```



se
Liga
Aí

O operador **in** retorna verdadeiro se o valor está contido no objeto.



Exemplos

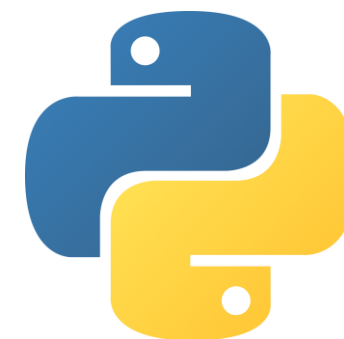
3- Faça um programa em Python que solicite ao usuário três números inteiros distintos e apresente o maior deles (desconsidere a possibilidade deles serem iguais)

Testes	Mensagem
$n1 > n2$ e $n1 > n3$	n1 é o maior
$n2 > n1$ e $n2 > n3$	n2 é o maior
$n3 > n1$ e $n3 > n2$	n3 é o maior

Exemplos

3- Faça um programa em Python que solicite ao usuário três números inteiros distintos e apresente o maior deles (desconsidere a possibilidade deles serem iguais)

```
main.py
1  n1 = int(input('Digite o primeiro número: '))
2  n2 = int(input('Digite o segundo número: '))
3  n3 = int(input('Digite o terceiro número: '))
4
5  ▼ if n1 > n2 and n1 > n3:
6      print('O primeiro número é o maior!')
7  ▼ elif n2 > n1 and n2 > n3:
8      print('O segundo número é o maior!')
9  ▼ else:
10     print('O primeiro número é o maior!')
```



Exemplos

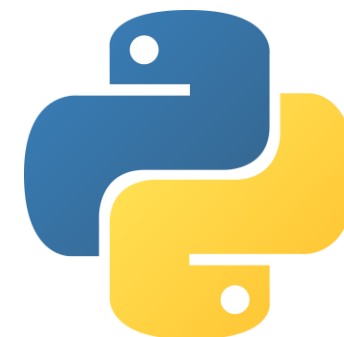
4- Crie um programa em Python que solicite ao usuário o peso e a altura, calcule o Índice de Massa Corpórea: $IMC = \text{peso} / \text{altura}^2$ e mostre em qual categoria o usuário se encontra, conforme a tabela abaixo:

Categoria	IMC
Abaixo do peso	< 20
Peso Normal	$\geq 20 \text{ e } < 25$
Sobrepeso	$\geq 25 \text{ e } < 30$
Obeso	$\geq 30 \text{ e } < 40$
Obeso Mórbido	≥ 40

Exemplo 4

exemplo4.py > ...

```
1 peso = float(input('Digite o seu peso em kg: '))
2 altura = float(input('Digite a sua altura em m: '))
3
4 imc = peso / altura**2
5
6 if imc < 20:
7     print('Abaixo do peso.')
8 elif imc < 25:
9     print('Peso normal.')
10 elif imc < 30:
11     print('Sobrepeso.')
12 elif imc < 40:
13     print('Obeso.')
14 else:
15     print('Obeso mórbido.')
```



Exemplos

5- Criar programa em Python que solicite ao usuário o valor total da compra e a quantidade de parcelas a financiar e o sistema deve imprimir o valor de cada parcela de acordo com os juros da tabela abaixo:

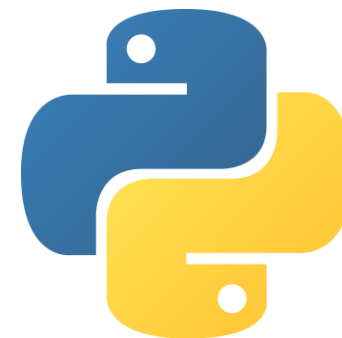
Nº Parcelas	% de juros
2	3
4	7
6	9
8	12

Exemplo 5

```
valor = float(input('Digite o valor da compra R$ '))
parcelas = int(input('Digite a quantidade de parcelas (2-4-6-8) '))

if parcelas == 2:
    valor = valor * 1.03
elif parcelas == 4:
    valor = valor * 1.07
elif parcelas == 6:
    valor = valor * 1.09
elif parcelas == 8:
    valor = valor * 1.12
else:
    valor = 0

if valor == 0:
    print('0 número de parcelas é inválido!!!')
else:
    print(f'0 valor de cada parcela é: R$ {valor/parcelas}')
```

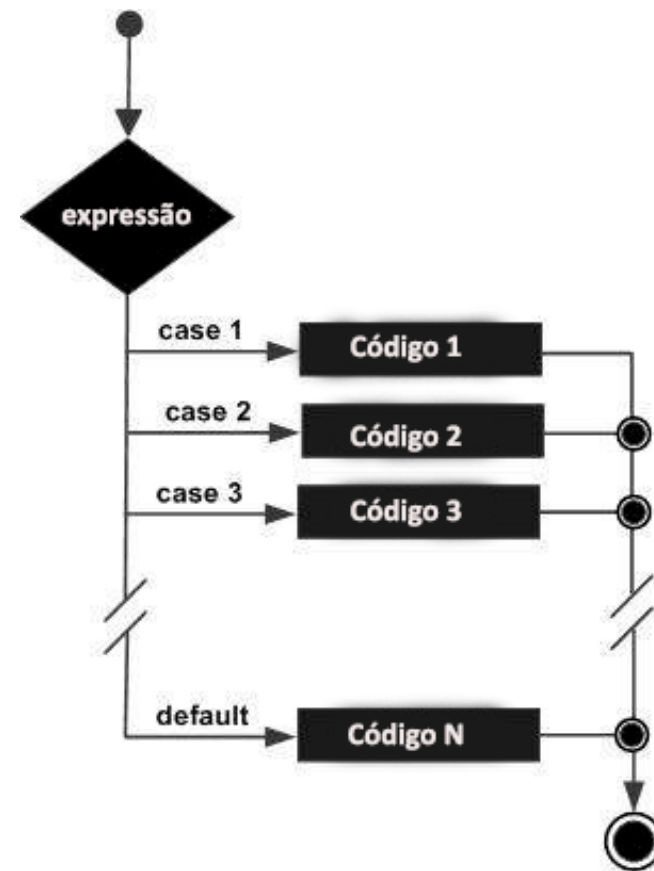


Pattern Matching ou “Correspondência de Padrões”

A partir da versão 3.10, o Python passou a oferecer uma maneira simples e eficaz de testar vários valores e realizar ações condicionais: a instrução match-case.

Similar ao switch-case, presente em diversas linguagens de programação, a match-case:

- ✓ É uma forma simples para se definir diversos desvios no código a partir de uma única variável.
- ✓ Pode ser usada quando se tem várias seleções com muitas alternativas.
- ✓ Equivale a comparação de igualdades.



Pattern Matching ou “Correspondência de Padrões”

A partir da versão 3.10, o Python passou a oferecer uma maneira simples e eficaz de testar vários valores e realizar ações condicionais: a instrução **match-case**.

```
match {elemento}:  
    case {padrão 1}:  
        {ação 1}  
    case {padrão 2}:  
        {ação2}  
    case {padrão 3}:  
        {ação 3}  
# Outros padrões...  
    case _:  
        {ação padrão ou default}
```

- ✓ **Elemento:** pode ser um inteiro, um decimal, uma lista, um objeto e assim por diante: é ele quem vai ser “testado”.
- ✓ O bloco **case padrão: ação** define o que fazer quando o padrão foi correspondido.
- ✓ Caso não haja correspondência, o **case_** executa a ação padrão

Exemplos

5- Criar programa em Python que solicite ao usuário o valor total da compra e a quantidade de parcelas a financiar e o sistema deve imprimir o valor de cada parcela de acordo com os juros da tabela abaixo:

Nº Parcelas	% de juros
2	3
4	7
6	9
8	12

Exemplo 5 – com o uso do match... case

exemplo5(match).py > ...

```
1 valor = float(input('Digite o valor da compra R$ '))
2 parcelas = int(input('Digite a quantidade de parcelas (2-4-6-8) '))
3
4 ~ match(parcelas):
5     case 2: valor = valor * 1.03
6     case 4: valor = valor * 1.07
7     case 6: valor = valor * 1.09
8     case 8: valor = valor * 1.12
9     case _: valor = 0
10
11 ~ if valor == 0:
12     print('0 número de parcelas é inválido!!!')
13 ~ else:
14     print(f'0 valor de cada parcela é: R$ {valor/parcelas}')
```



Exemplos

6- Na cidade de São Paulo a circulação de veículos é restrita a determinados horários no chamado “Centro Expandido”. Com base no dígito final da placa, o motorista sabe se deve ou não circular no horário proibido. Considere a tabela abaixo:

Dígito final da placa	Não pode circular
1 ou 2	Segunda-feira
3 ou 4	Terça-feira
5 ou 6	Quarta-feira
7 ou 8	Quinta-feira
9 ou 0	Sexta-feira

Faça um programa que receba o número inteiro da placa de um automóvel, verifique e imprima qual o dia da semana o veículo não pode circular no “Centro Expandido”

Exemplo 6

exemplo6.py > ...

```
1 placa = int(input('Digite os quatro dígitos da placa do veículo: '))
2
3 final = placa % 10
4
5 if final == 1 or final == 2:
6     print('O veículo não pode circular às segundas-feiras')
7 elif final == 3 or final == 4:
8     print('O veículo não pode circular às terças-feiras')
9 elif final == 5 or final == 6:
10    print('O veículo não pode circular às quartas-feiras')
11 elif final == 7 or final == 8:
12    print('O veículo não pode circular às quintas-feiras')
13 else:
14    print('O veículo não pode circular às sextas-feiras')
```



Exemplo 6 - com o uso do match... case

```
exemplo6(match).py > ...
1  placa = int(input('Digite os quatro dígitos da placa do veículo: '))
2
3  final = placa % 10
4
5  match(final):
6      case 1 | 2:
7          print('O veículo não pode circular às segundas-feiras')
8      case 3 | 4:
9          print('O veículo não pode circular às terças-feiras')
10     case 5 | 6:
11         print('O veículo não pode circular às quartas-feiras')
12     case 7 | 8:
13         print('O veículo não pode circular às quintas-feiras')
14     case 9 | 0:
15         print('O veículo não pode circular às sextas-feiras')
16     case _:
17         print('Placa inválida!!!')
```



Alguma dúvida????



Python 3.10

Python 3.10 - novidades da versão:

- ✔ <https://pythonacademy.com.br/blog/python-3-10-o-que-ha-de-novo>
- ✔ <https://docs.python.org/pt-br/3/whatsnew/3.10.html>



Exercícios de aplicação



Observações sobre exercícios

- ✓ Todos os exercícios devem ser resolvidos em Python.
- ✓ O código Python pode ser feito no IDLE, no Repl.it, ou na ferramenta que você ache mais adequada e deve ser salvo um arquivo por exercício com a extensão .py
- ✓ Após finalizar todos os exercícios da aula, coloque-os em uma pasta com o nome da aula, compacte a pasta e envie no Blackboard.



<https://youtu.be/BuxuUbgKwCg>



Exercícios

1- Criar um programa em Python que leia a idade de uma pessoa e informe sua classe eleitoral:

- não-eleitor (abaixo de 16 anos)
- eleitor obrigatório (entre 18 e 65 anos)
- eleitor facultativo (entre 16 e 18 anos e maior de 65 anos)

2- Faça um programa em Python que leia três valores inteiros (variáveis a , b e c) e efetue o cálculo da equação de segundo grau, apresentando: (i) as duas raízes, quando for possível efetuar o cálculo (delta positivo ou zero); (ii) a mensagem "Não há raízes reais", se não for possível fazer o cálculo (delta negativo); (iii) a mensagem "Não é equação do segundo grau", se o valor de a for igual a zero.

Exercícios

3- Um comerciante calcula o valor da venda, tendo em vista a tabela a seguir:

Valor de compra	Valor de venda
valor < R\$10,00	lucro de 70%
R\$ 10,00 <= valor < R\$ 30,00	lucro de 50%
R\$ 30,00 <= valor < R\$ 50,00	lucro de 40%
valor >= R\$50,00	lucro de 30%

Crie uma programa em Python que permita digitar o nome do produto e valor da compra, e imprimindo o nome do produto e o valor da venda.

Exercícios

4- Elabore um programa em Python que implemente uma calculadora com as funções de somar, subtrair, multiplicar e dividir. O programa deverá solicitar ao usuário os dois valores, e perguntar qual a operação pretendida ('+', '-', '*' ou '/') e a seguir calcular e mostrar o resultado.

Créditos

Esta aula foi elaborada com base no material produzido e cedido gentilmente pelos **Professores Alcides, Lédon, Amilton e Cristiane.**





That's all Folks!