

Comandos Condicionais Avançados

CIÊNCIA DA
COMPUTAÇÃO

ATITUS
EDUCAÇÃO



ATITUS
EDUCAÇÃO



- Aula 04 -
Pensamento Computacional

Prof. Me. Lucas R. C. Pessutto

Na aula de hoje...

01

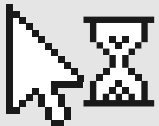
Condicionais Múltiplos

Comando match - case

Motivação e funcionamento

02

Condicionais inline



Exercício

- * Fazer o trecho de um programa que, dependendo do conteúdo de “codigo”, atribua o conteúdo especificado na tabela ao lado à variável “resultado”. Suponha que a variável resultado inicie com o valor 20

Código	Resultado
1	resultado + 5
2	Inalterado
3	resultado – 2
4	Inalterado
5	resultado – 2
6	$2 * (\text{resultado} + 1)$
7	Inalterado
8	0
9	0
10	0
11	123

Exercício

```
codigo = int(input("Informe o código: "))

resultado = 20

if codigo == 1:
    resultado += 5
elif codigo == 3 or codigo == 5:
    resultado -= 2
elif codigo == 6:
    resultado = 2 * (resultado + 1)
elif codigo == 8 or codigo == 9 or codigo == 10:
    resultado = 0
elif codigo == 11:
    resultado = 123

print(f"RESULTADO = {resultado}")
```

Código	Resultado
1	resultado + 5
2	Inalterado
3	resultado - 2
4	Inalterado
5	resultado - 2
6	2 * (resultado + 1)
7	Inalterado
8	0
9	0
10	0
11	123

Comando match-case

```
match expressao:  
    case <padrao1>:  
        <comandos 1>  
    case <padrao2>:  
        <comandos 2>  
  
    ...  
    case <padraon>:  
        <comandos n>  
    case _:  
        <comandos _>
```

Caso padrão:
executado quando não
satisfaz nenhuma
condição anterior

Comando match-case

- * Expressão deve ser do mesmo tipo dos padrões
- * Compara o valor retornado pela expressão a uma sequência de conteúdos constantes, na sequência em que aparecem
- * Se a expressão == conteúdo, o bloco de comandos associado é executado
- * Não podem existir padrões repetidos

```
match expressao:  
    case <padrao1>:  
        <comandos 1>  
    case <padrao2>:  
        <comandos 2>  
    ...  
    case <padraon>:  
        <comandos n>  
    case _:  
        <comandos _>
```

```
codigo = int(input("Informe o código: "))
```

```
resultado = 20
```

```
match codigo:
```

```
    case 1:
```

```
        resultado += 5
```

```
    case 3:
```

```
        resultado -= 2
```

```
    case 5:
```

```
        resultado -= 2
```

```
    case 6:
```

```
        resultado = 2 * (resultado + 1)
```

```
    case 8:
```

```
        resultado = 0
```

```
    case 9:
```

```
        resultado = 0
```

```
    case 10:
```

```
        resultado = 0
```

```
    case 11:
```

```
        resultado = 123
```

```
print(f"RESULTADO = {resultado}")
```

Código	Resultado
1	resultado + 5
2	Inalterado
3	resultado - 2
4	Inalterado
5	resultado - 2
6	2 * (resultado + 1)
7	Inalterado
8	0
9	0
10	0
11	123

Padrão ou

- * O operador pipe pode ser usado quando há um conjunto de padrões que disparam um mesmo comportamento

- * Sua sintaxe é:

```
case <padrao1> | <padrao2> | <padrao3>:
```

- * O bloco de comandos é executado caso a expressão seja igual a um dos padrões


```
codigo = int(input("Informe o código: "))

resultado = 20
match codigo:
    case 1:
        resultado += 5
    case 3 | 5:
        resultado -= 2
    case 6:
        resultado = 2 * (resultado + 1)
    case 8 | 9 | 10:
        resultado = 0
    case 11:
        resultado = 123

print(f"RESULTADO = {resultado}")
```

Código	Resultado
1	resultado + 5
2	Inalterado
3	resultado – 2
4	Inalterado
5	resultado – 2
6	2 * (resultado + 1)
7	Inalterado
8	0
9	0
10	0
11	123

Problema 01: Calculadora



Enunciado de um problema:

Fazer um programa que simule uma calculadora que executa as quatro operações aritméticas (soma, subtração, produto e divisão).

O programa deve ler a operação (char) que deverá ser efetuada e os dois valores (operandos-float)

Emitir mensagem de erro caso não seja uma operação válida (operações válidas: '+', '-', '*', '/')

Após o cálculo, dar a resposta.

Problema 01: Calculadora

```
operando1, operador, operando2 = input("Informe a operação: ").split(" ")
```

```
operando1, operando2 = float(operando1), float(operando2)
```

```
erro = False
```

```
match operador:
```

```
    case "+":
```

```
        resultado = operando1 + operando2
```

```
    case "-":
```

```
        resultado = operando1 - operando2
```

```
    case "*":
```

```
        resultado = operando1 * operando2
```

```
    case "/":
```

```
        resultado = operando1 / operando2
```

```
    case _:
```

```
        print("Operação desconhecida!")
```

```
        erro = True
```

```
if not erro:
```

```
    print(f"{operando1} {operador} {operando2} = {resultado}")
```

Exercício: Estado Civil

Fornecido um código caractere, correspondente a um estado civil, devolver por extenso o estado civil correspondente.

Devem ser aceitas na entrada tanto maiúsculas quanto minúsculas.

Codificação dos estados civis considerados:

S - Solteiro

C - Casado

D - Divorciado

V – Viúvo

Qualquer outro caractere de entrada deve ser acusado como estado civil incorreto.

Comando Condicional inline

- * Permite executar um comando if usando uma única linha de código
- * Também chamado de Operador Ternário
- * Esse comando permite avaliar uma condição e retornar um dentre dois valores possíveis
- * Sintaxe:

```
<variavel> = <valor A> if <expressão_logica> else <valor B>
```

Comando Condicional inline

Exemplos:

```
x = True
```

```
if x:  
    y = 10  
else:  
    y = 20
```

```
print(y)
```

```
x = True  
y = 10 if x else 20  
print(y)
```

Comando Condicional inline

Exemplos:

```
x = 3
```

```
if x == 1:
    y = 10
elif x == 2:
    y = 20
else:
    y = 30
```

```
print(y)
```

```
x = 3
y = 10 if x == 1 else (20 if x == 2 else 30)
print(y)
```

Exemplos

```
media = 10
if media >= 6:
    res = "aprovado"
else:
    res = "reprovado"
```

```
res = "aprovado" if media >= 6 else "reprovado"
```

```
a = 10
num = 8
a += 1 if num > 10 else -1
print(a)
```

O que é impresso por esse código?

9