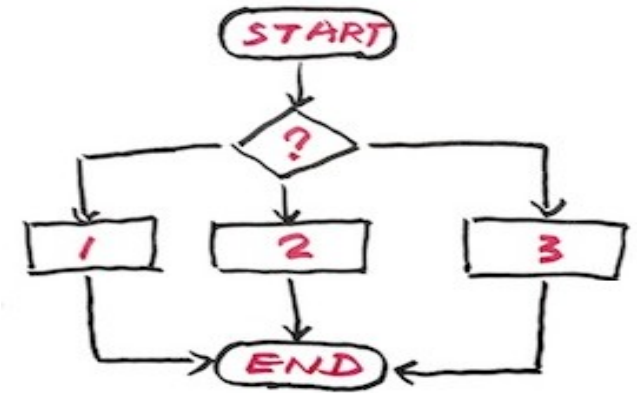


Introdução à Programação

Aula 10 *Switch case*



Prof. Dr. Ivan José dos Reis Filho

Aulas anteriores

Estrutura de repetição

- For
- While
- Do While

Aula de hoje

Comandos

- Break
- Continue
- Switch – case – default

Comando *break*

Comando **break** é utilizado em conjunto com duas estruturas:

- Repetição (for, while e do while)
- Seletiva (switch / match)

O break é utilizado para sair abruptamente da estrutura.

Ex: Se estivermos em uma interação (repetição) e um tal resultado ocorrer, o programa deve sair da interação.

Exemplo

```
1 for i in range(100):  
2 |  
3 |     print(i)  
4 |  
5 |     if i == 10:  
6 | |     break  
7 |
```

↳ 0
1
2
3
4
5
6
7
8
9
10

Quando o **i** for igual 10 o comando **break** interrompe a estrutura de repetição.

Comando *continue*

Comando **continue** se parece com o **break**, mas a diferença é que ele apenas ignora o resto do bloco de dados de uma interação.

Exemplo: Vamos imaginar um programa deve retornar apenas os múltiplos de 5 de um sequência. Então, temos que verificar número por número e devemos ignorar todos os números que não são múltiplos de 5.

próximo slide →

Exemplo

```
1 for i in range(100):  
2 |  
3 |   if i%5 != 0:  
4 | |   continue  
5 |  
6 |   print(i)  
7
```

```
0  
5  
10  
15  
20  
25  
30  
35  
40  
45  
50  
55  
60  
65  
70  
75  
80  
85  
90  
95
```

Quando o **i** não for múltiplo de 5, o comando **continue** continua a interação sem executar o restante do código da estrutura de repetição.

Atividade 01

Faça uma estrutura de repetição que percorre o valor de 1 até 100. Quando a interação tiver no número 25 o algoritmo deve encerrar a estrutura de repetição.

Atividade 02

Faça uma estrutura de repetição que imprime todos os número de 0 a 50, exceto os que são múltiplos de 3 ou 4. Use o comando continue.

O comando match

Quase que **if elif else** com várias possibilidades, mas com algumas diferenças importantes.

- **1ª diferença:** Os cases não aceitam operadores lógicos. Portanto, não é possível fazer uma comparação.
- **2ª diferença:** O match executa seu bloco em cascata. Ou seja, se a variável indicar para o primeiro case e dentro do switch tiver 5 cases, o switch executará todos os outros 4 cases a não ser que utilizemos o comando para sair do switch. (Nos referimos ao **BREAK**).

Estrutura

match **variável**

CASE **valor1:**

Dados a serem executados

BREAK;

CASE **valor2:**

Dados a serem executados

BREAK;

CASE **_:**

Nenhuma das opções anteriores são válidas

BREAK

Exemplo

A estrutura match (semelhante ao switch-case de outras linguagens) foi introduzida no **Python 3.10** como parte da "structural pattern matching".

```
1 print("Menu de Opções:")
2 print("1 - Ver saldo")
3 print("2 - Sacar dinheiro")
4 print("3 - Depositar dinheiro")
5 print("4 - Sair")
6
7 opcao = int(input("Escolha uma opção: "))
8
9 match opcao:
10 |     case 1:
11 |         print("Seu saldo é de R$ 1.000,00")
12 |     case 2:
13 |         print("Você escolheu sacar dinheiro.")
14 |     case 3:
15 |         print("Você escolheu depositar dinheiro.")
16 |     case 4:
17 |         print("Saindo do sistema...")
18 |     case _:
19 |         print("Opção inválida.")
20
```

Comando default

O case _ (**default**) é ativado caso não tenha achado nenhum case definido.

Exemplo: Seu programa pede para que o usuário digite apenas duas opções (S ou N) para reiniciar o programa.

Mas, propositalmente ou por engano, o usuário digita uma opção totalmente diferente. E agora? O que seu programa deve fazer? É aqui que o **default** entra.

Exercício 01

Faça um algoritmo que recebe dois valores inteiros. Faça um **menu** para o usuário escolher se ele deseja somar (1), subtrair(2), multiplicar(3) ou dividir (4). Caso usuário digite uma informação que não pertence aos “case”, informe opção errada.

```
print("=== Menu de Operações ===")
print("1 - Soma")
print("2 - Subtração")
print("3 - Multiplicação")
print("4 - Divisão")

opcao = int(input("Escolha uma operação (1 a 4): "))
a = float(input("Digite o primeiro número: "))
b = float(input("Digite o segundo número: "))
```

Exercício 02

Crie um algoritmo que simula um menu bancário. O sistema deve permanecer em execução até que o usuário escolha a opção de sair. Use while para manter o menu ativo e match para tratar as opções.

```
saldo = 1000.0 # saldo inicial
opcao = 0

while opcao != 4:
    print("\n--- MENU BANCÁRIO ---")
    print("1 - Ver saldo")
    print("2 - Depositar")
    print("3 - Sacar")
    print("4 - Sair")
```

Exercício 03

Crie um programa que simule o cálculo de salários de funcionários com base nas horas trabalhadas. O sistema deve ter um menu com as seguintes opções:

1. Calcular salário
2. Sair

O programa deve permanecer em execução até que o usuário escolha sair. O usuário deve digitar:

- o valor da hora trabalhada;
- a quantidade de horas normais
- a quantidade de horas extras

$$\text{salário} = (\text{horas_normais} * \text{valor_hora}) + (\text{horas_extras} * \text{valor_hora} * 1.5)$$

Exercício 04