



Engenharia de Software
Computacional thinking with Python
Aula 06 – Estrutura de Condição aninhada e
estrutura de repetição

Prof. Dr. Francisco Elânio

Objetivo

- **Estrutura condicional aninhada**
- **Exemplos / Exercícios**
- **Estrutura de repetição while**
- **Exemplos / Exercícios**

Estrutura de Condição Aninhada

Serve para organizar e realizar tomadas de decisão complexas com base em múltiplas condições. Ela permite que você avalie várias condições dentro de outras condições, criando uma hierarquia de decisões.

Essa estrutura é útil quando você precisa lidar com situações em que uma decisão depende de várias condições ou critérios diferentes.

Estrutura de controle aninhada – Ex. 1

```
minutos = int(input("Minutos:"))
```

```
if minutos < 200:  
    preco = 0.2
```

```
else:  
    if minutos <= 400:  
        preco = 0.18  
    else:  
        preco = 0.15
```

```
print(f'R$ {preco * minutos}')
```


Elegibilidade para desconto – Ex. 2

```
idade = int(input("Digite sua idade: "))
total_compra = float(input("Digite o total da compra: "))

if idade >= 65:
    if total_compra > 100:
        print("Você tem direito a um desconto de 10%!")
    else:
        print("Desculpe, o desconto é válido apenas para compras  
acima de $100.")
else:
    print("Você não tem direito ao desconto.")
```

Estrutura de controle aninhada – Ex. 3

```
ano_carro = int(input("Digite o ano do carro: "))
tipo_carro = input("Digite o tipo do carro (mecânico ou automático): ")
marca_carro = input("Digite a marca do carro: ")

if 2013 < ano_carro <= 2015:
    if marca_carro == "Ford":
        print("Carro da marca Ford fabricado entre 2013 e 2015.")
    else:
        print("Carro fabricado entre 2013 e 2015, mas não é da marca Ford.")
else:
    print("Carro fora do intervalo de anos especificado.")
```

Ex. 4: Verificação do horário do dia e temperatura para sugerir uma atividade ao ar livre

```
hora = int(input("Digite a hora atual (0-23): "))
temperatura = float(input("Digite a temperatura atual (em Celsius): "))

if 6 <= hora < 12:
    if temperatura > 20:
        print("Hora de fazer uma caminhada matinal!")
    else:
        print("Hora de tomar um café da manhã quente!")
elif 12 <= hora < 18:
    if temperatura > 25:
        print("Que tal um piquenique no parque?")
    else:
        print("Hora de um almoço leve!")
else:
    print("Hora de relaxar em casa.")
```

Ex. 5: Verificação do status de um pedido com base na idade do pedido e no tipo de cliente

```
idade_pedido = int(input("Digite a idade do pedido (em dias): "))
tipo_cliente = input("Digite o tipo de cliente (regular ou VIP): ")

if idade_pedido < 7:
    if tipo_cliente == "VIP":
        print("Seu pedido está sendo preparado para entrega expressa!")
    else:
        print("Seu pedido será entregue em breve.")
elif 7 <= idade_pedido < 14:
    if tipo_cliente == "VIP":
        print("Seu pedido está em processo de embalagem especial!")
    else:
        print("Seu pedido está sendo preparado.")
else:
    print("Seu pedido está a caminho.")
```


Ex.6 : Sistema de Recomendação de Restaurantes com Base na Preferência de Culinária e no Orçamento

```
preferencia_culinaria = input("Digite sua preferência culinária (italiana, japonesa, mexicana, brasileira): ")
orcamento = float(input("Digite seu orçamento disponível (em reais): "))

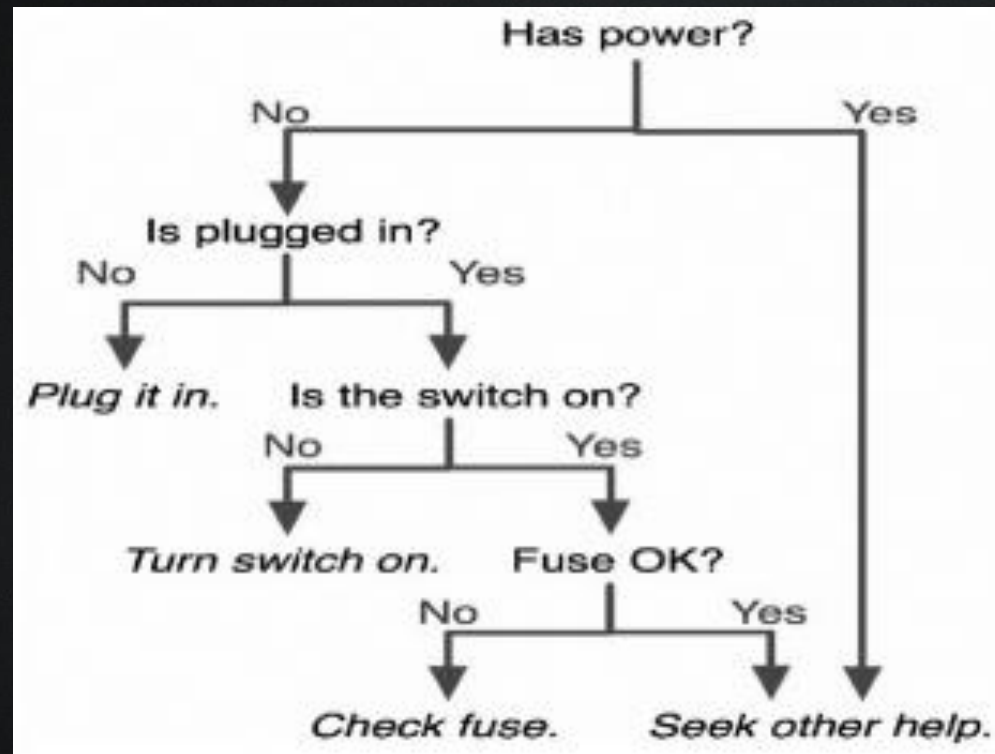
if preferencia_culinaria == "italiana":
    if orcamento >= 100:
        print("Recomendação: Restaurante italiano de alta gastronomia")
    else:
        print("Recomendação: Pizzaria tradicional")
elif preferencia_culinaria == "japonesa":
    if orcamento >= 150:
        print("Recomendação: Restaurante japonês com menu degustação")
    else:
        print("Recomendação: Rodízio de sushi")
elif preferencia_culinaria == "mexicana":
    if orcamento >= 80:
        print("Recomendação: Taqueria autêntica")
    else:
        print("Recomendação: Restaurante de comida mexicana casual")
else:
    if orcamento >= 60:
        print("Recomendação: Churrascaria rodízio")
    else:
        print("Recomendação: Buffet de comida brasileira")
```

Ex. 7 - Verificação de acesso a um site

```
usuario = input("Nome de usuário: ")
senha = input("Senha: ")

if usuario == "admin":
    if senha == "password":
        print("Bem-vindo, admin! Acesso concedido.")
    else:
        print("Senha incorreta. Tente novamente.")
else:
    print("Nome de usuário incorreto. Tente novamente.")
```

Ex. 8 - Crie um algoritmo para resolver o problema de energia



Estrutura de repetição

Looping é um programa ou procedimento de script que executa tarefas repetitivas (ou iterativas).

O loop é uma sequência de instruções (um bloco) que é executada repetidamente enquanto ou até que alguma condição seja atendida ou satisfeita. Cada repetição é chamada de iteração.

Estrutura de repetição

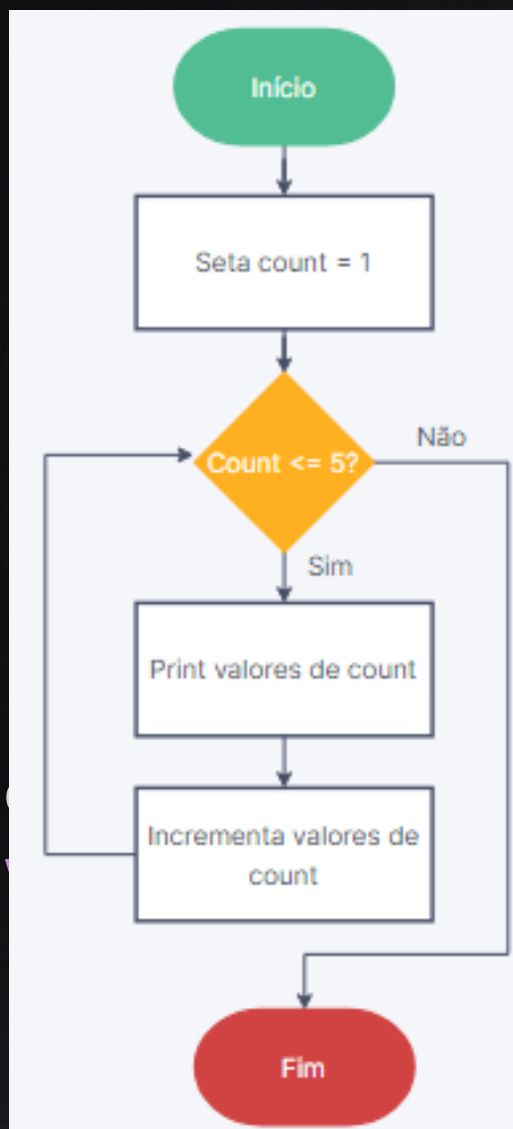
- A palavra-chave `while` inicia a instrução `while`.
- A condição determina se o corpo será (ou continuará sendo) executado. Um `cólon (:)` deve seguir a condição.
- `block` é um bloco de uma ou mais instruções a serem executadas desde que a condição seja verdadeira. Todas as instruções que compõem o bloco devem ser recuadas o mesmo número de espaços a partir da esquerda.

```
count = 1
while count <= 5:
    print(count)
    count += 1
```

A palavra-chave `while` no Programa faz com que a variável `'count'` seja repetidamente impressa e incrementada até que seu valor atinja cinco.

Estrutura de repetição

Fluxograma



```
count = 1
while count <= 5:
    print (count)
    count += 1
```

Estrutura de repetição

Ex. 2

Permita que os usuários insiram um valor inteiro positivo. Quando o usuário insere um inteiro negativo, o loop termina e o código imprime a soma dos inteiros positivos inseridos até aquele ponto. A soma é zero se o primeiro valor for negativo

```
soma = 0
```

```
print("Entre com um número para soma:")  
entrada = int(input())
```

```
while entrada > 0:  
    soma += entrada  
    print("A soma é =", soma)  
    entrada = int(input())
```

Estrutura de repetição

Ex. 3 – Estrutura booleana

```
realizado = False
while not realizado:
    entrada = eval(input())
```

```
if entrada == 999:
    realizado = True
else:
    print(entrada)
```

- `realizado = False`: Inicializa a variável realizado como False.
- `while not realizado::` Inicia um loop while que continuará enquanto a variável realizado for False.
- `entrada = eval(input())`: Solicita ao usuário que insira uma entrada e avalia a entrada usando a função `eval()`.
- `if entrada == 999::` Verifica se a entrada do usuário é igual a 999.
- Se a entrada for 999, a variável realizado é definida como True, o que termina o loop while.
- Se a entrada não for 999, o programa imprime a entrada.

Estrutura de repetição

Ex. 4 – Soma números dentro do intervalo

```
soma = 0
num = 3
while num <= 5:
    soma += num
    num += 1
print("A soma dos números de 1 a 5 é:", soma)
```

- **soma = 0:** Inicializa a variável soma com o valor zero para acumular a soma dos números.
- **num = 3:** Inicializa a variável num com o valor 3, que é o primeiro número do intervalo que queremos somar.
- **while num <= 5::** Inicia um loop while que continua enquanto o valor de num for menor ou igual a 5.
- **soma += num:** Adiciona o valor atual de num à variável soma.
- **num += 1:** Incrementa o valor de num em 1 a cada iteração do loop.
- **print("A soma dos números de 1 a 5 é:", soma):** Após o loop, imprime a soma dos números dentro do intervalo [3, 4, 5].

Estrutura de repetição

Ex. 5 – Soma números dentro do intervalo

- `num = int(input("Digite um número para calcular o fatorial: ")):`
- `fat = 1:` Inicializa a variável `fat` com o valor 1.
- `i = 1:` Inicializa a variável `i` com o valor 1.
- `while i <= num::` Inicia um loop `while` que continuará enquanto o valor de `i` for menor ou igual a `num`.
- `fat *= i:` Multiplica o valor atual de `fat` pelo valor atual de `i` e atribui o resultado de volta à variável `fat`.
- `i += 1:` Incrementa o valor de `i` em 1 a cada iteração do loop.
- `print("O fatorial de", num, "é:", fat):`

```
num = int(input("Digite um número para calcular o fatorial: "))
fat = 1
i = 1
while i <= num:
    fat *= i
    i += 1
print("O fatorial de", num, "é:", fat)
```

*O importante é não parar
de questionar (Einstein)*

Referências

ASCENCIO, A. F. G, CAMPOS, E. A. V. Fundamentos da Programação de Computadores: algoritmos, Pascal, C/C++ e Java, 2ª Edição, São Paulo: Pearson 2007.

SALVETTI, Dirceu Douglas; BARBOSA, Lisbete Madson. Algoritmos. São Paulo: Pearson, 2004.