

# Estruturas Condicionais em Python

Seu Nome

## 1 Introducao

As estruturas condicionais sao fundamentais na programacao, permitindo que os programas tomem decisoes baseadas em condicoes especificas. Em Python, as principais estruturas condicionais sao `if`, `else` e `elif`. Este documento explora detalhadamente cada uma dessas estruturas, fornecendo exemplos praticos e casos de uso.

## 2 Estrutura if

A estrutura `if` e a base das declaracoes condicionais em Python. Ela permite que um bloco de codigo seja executado apenas se uma condicao especifica for avaliada como verdadeira.

### 2.1 Sintaxe basica

```
1 if condicao:
2     # codigo a ser executado se a condicao for verdadeira
```

### 2.2 Exemplos

#### 2.2.1 Exemplo 1: Verificacao de idade

```
1 idade = 18
2 if idade >= 18:
3     print("Voce e maior de idade.")
4     print("Pode entrar na festa.")
```

#### 2.2.2 Exemplo 2: Verificacao de numero par

```
1 numero = 10
2 if numero % 2 == 0:
3     print(f"{numero} e um numero par.")
```

#### 2.2.3 Exemplo 3: Verificacao de string vazia

```
1 nome = ""
2 if not nome:
3     print("Por favor, insira um nome valido.")
```

## 3 Estrutura else

A estrutura `else` e usada em conjunto com `if` para executar um bloco de codigo quando a condicao do `if` nao e verdadeira. Isso permite criar alternativas quando a condicao principal nao e atendida.

### 3.1 Sintaxe basica

```
1 if condicao:
2     # codigo a ser executado se a condicao for verdadeira
3 else:
4     # codigo a ser executado se a condicao for falsa
```

### 3.2 Exemplos

#### 3.2.1 Exemplo 1: Verificacao de idade com alternativa

```
1 idade = 16
2 if idade >= 18:
3     print("Voce e maior de idade.")
4     print("Pode entrar na festa.")
5 else:
6     print("Voce e menor de idade.")
7     print("Nao pode entrar na festa.")
```

#### 3.2.2 Exemplo 2: Verificacao de login

```
1 usuario = "admin"
2 senha = "senha123"
3
4 if usuario == "admin" and senha == "senha123":
5     print("Login bem-sucedido!")
6 else:
7     print("Usuario ou senha incorretos.")
```

#### 3.2.3 Exemplo 3: Verificacao de numero positivo/negativo

```
1 numero = -5
2 if numero >= 0:
3     print(f"{numero} e positivo ou zero.")
4 else:
5     print(f"{numero} e negativo.")
```

## 4 Estrutura elif

A estrutura `elif` (abreviacao de "else if") e usada para verificar multiplas condicoes em sequencia. Ela e particularmente util quando ha varias alternativas possiveis.

### 4.1 Sintaxe basica

```
1 if condicao1:
2     # codigo para condicao1
3 elif condicao2:
4     # codigo para condicao2
5 elif condicao3:
6     # codigo para condicao3
7 else:
8     # codigo se nenhuma condicao for verdadeira
```

## 4.2 Exemplos

### 4.2.1 Exemplo 1: Sistema de notas

```
1 nota = 75
2 if nota >= 90:
3     print("A")
4 elif nota >= 80:
5     print("B")
6 elif nota >= 70:
7     print("C")
8 elif nota >= 60:
9     print("D")
10 else:
11     print("F")
```

### 4.2.2 Exemplo 2: Categorias de IMC

```
1 imc = 27.5
2 if imc < 18.5:
3     print("Abaixo do peso")
4 elif 18.5 <= imc < 25:
5     print("Peso normal")
6 elif 25 <= imc < 30:
7     print("Sobrepeso")
8 elif 30 <= imc < 35:
9     print("Obesidade grau 1")
10 elif 35 <= imc < 40:
11     print("Obesidade grau 2")
12 else:
13     print("Obesidade grau 3")
```

### 4.2.3 Exemplo 3: Dias da semana

```
1 dia = 3
2 if dia == 1:
3     print("Segunda-feira")
4 elif dia == 2:
5     print("Terça-feira")
6 elif dia == 3:
7     print("Quarta-feira")
8 elif dia == 4:
9     print("Quinta-feira")
10 elif dia == 5:
11     print("Sexta-feira")
12 elif dia == 6:
13     print("Sabado")
14 elif dia == 7:
15     print("Domingo")
16 else:
17     print("Dia invalido")
```

## 5 Combinando estruturas condicionais

As estruturas condicionais podem ser combinadas e aninhadas para criar logicas mais complexas.

### 5.1 Exemplo: Sistema de desconto

```

1 idade = 25
2 membro = True
3 compra = 150
4
5 if idade >= 60:
6     if membro:
7         desconto = 0.20 # 20% de desconto
8     else:
9         desconto = 0.15 # 15% de desconto
10 elif membro:
11     if compra >= 100:
12         desconto = 0.10 # 10% de desconto
13     else:
14         desconto = 0.05 # 5% de desconto
15 else:
16     desconto = 0 # Sem desconto
17
18 valor_final = compra * (1 - desconto)
19 print(f"Valor final da compra: R$ {valor_final:.2f}")

```

## 6 Operadores logicos em condicoes

Python suporta os operadores logicos `and`, `or` e `not`, que podem ser usados para criar condicoes mais complexas.

### 6.1 Exemplo: Verificacao de elegibilidade para emprestimo

```

1 idade = 30
2 renda = 5000
3 score_credito = 700
4
5 if (idade >= 21 and idade <= 65) and (renda >= 3000) and (score_credito >= 650):
6     print("Elegivel para emprestimo")
7 elif idade < 21 or idade > 65:
8     print("Idade fora da faixa permitida")
9 elif renda < 3000:
10    print("Renda insuficiente")
11 else:
12    print("Score de credito baixo")

```

## 7 Estrutura match-case (Python 3.10+)

A partir do Python 3.10, foi introduzida a estrutura `match-case`, que e semelhante ao `switch-case` de outras linguagens como C e Java. Esta estrutura fornece uma maneira mais elegante de lidar com multiplas condicoes, especialmente quando se esta comparando uma variavel com varios valores possiveis.

### 7.1 Sintaxe basica

```

1 match variavel:
2     case padrao1:
3         # codigo para padrao1
4     case padrao2:
5         # codigo para padrao2
6     case padrao3:
7         # codigo para padrao3
8     case _:
9         # codigo padrao (similar ao 'default' em switch)

```

## 7.2 Exemplos

### 7.2.1 Exemplo 1: Dias da semana

```
1 dia = 3
2 match dia:
3     case 1:
4         print("Segunda-feira")
5     case 2:
6         print("Terca-feira")
7     case 3:
8         print("Quarta-feira")
9     case 4:
10        print("Quinta-feira")
11    case 5:
12        print("Sexta-feira")
13    case 6:
14        print("Sabado")
15    case 7:
16        print("Domingo")
17    case _:
18        print("Dia invalido")
```

### 7.2.2 Exemplo 2: Tipos de dados

```
1 def tipo_dado(valor):
2     match valor:
3         case int():
4             print("E um inteiro")
5         case float():
6             print("E um numero de ponto flutuante")
7         case str():
8             print("E uma string")
9         case list():
10            print("E uma lista")
11        case dict():
12            print("E um dicionario")
13        case _:
14            print("Tipo nao reconhecido")
15
16 tipo_dado(42)
17 tipo_dado("Python")
18 tipo_dado([1, 2, 3])
```

### 7.2.3 Exemplo 3: Padroes complexos

O match-case tambem suporta padroes mais complexos, incluindo desempacotamento de sequencias e correspondencia de estruturas.

```
1 def analisar_ponto(ponto):
2     match ponto:
3         case (0, 0):
4             print("Origem")
5         case (0, y):
6             print(f"No eixo Y, y={y}")
7         case (x, 0):
8             print(f"No eixo X, x={x}")
9         case (x, y) if x == y:
10            print(f"Na diagonal, x=y={x}")
11        case (x, y):
12            print(f"Ponto generico ({x}, {y})")
13        case _:
```

```
14         print("Nao e um ponto valido")
15
16 analisar_ponto((0, 0))
17 analisar_ponto((0, 5))
18 analisar_ponto((3, 3))
19 analisar_ponto((2, 4))
```

### 7.3 Vantagens do match-case

- Mais legível e conciso que múltiplos `elif`
- Suporta padrões complexos e desempacotamento
- Permite correspondência de tipos e estruturas
- Mais eficiente em termos de desempenho para múltiplas comparações

### 7.4 Considerações

É importante notar que a estrutura `match-case` está disponível apenas a partir do Python 3.10. Para versões anteriores, é necessário usar as estruturas `if-elif-else` tradicionais.

## 8 Conclusão

Neste documento, exploramos detalhadamente as estruturas condicionais em Python:

- `if`: para executar código com base em uma condição
- `else`: para fornecer uma alternativa quando a condição do `if` não é atendida
- `elif`: para verificar múltiplas condições em sequência
- `match-case`: uma alternativa moderna ao `switch-case` (Python 3.10+)

Também vimos como essas estruturas podem ser combinadas e usadas com operadores lógicos para criar lógicas de decisão complexas. O domínio dessas estruturas é essencial para criar programas Python flexíveis e responsivos.

## 9 Exercícios Práticos

1. Escreva um programa que solicite a idade do usuário e imprima se ele pode ou não votar (a idade mínima para votar é 16 anos).
2. Crie um programa que calcule o IMC (Índice de Massa Corporal) de uma pessoa e classifique o resultado em: Abaixo do peso, Peso normal, Sobrepeso ou Obeso.
3. Desenvolva um programa que solicite três números ao usuário e imprima o maior deles.
4. Implemente um programa que determine se um ano fornecido pelo usuário é bissexto ou não.
5. Crie um programa que simule um sistema de notas escolar, onde o usuário insere uma nota de 0 a 100 e o programa retorna o conceito correspondente (A, B, C, D ou F).
6. Desenvolva um programa que simule um caixa eletrônico. O programa deve perguntar o valor do saque e informar quantas notas de cada valor serão fornecidas, considerando que há notas de R\$100, R\$50, R\$20 e R\$10.
7. Crie um programa que determine o preço de um produto com base em seu código. Use a estrutura `elif` para lidar com diferentes códigos de produto.