

Departamento de Engenharia de Sistemas e Computação

Estruturas de Decisão

Seleção simples, composta e em sequência: if, else, elif

João Araujo

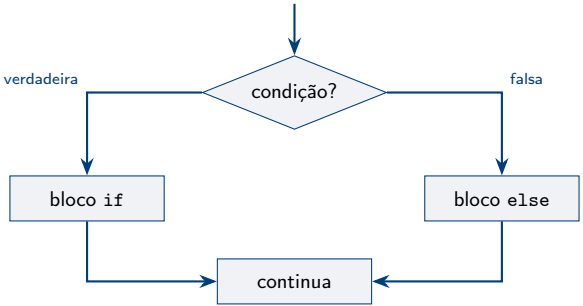
Algoritmos Computacionais I

Aula 07

Roteiro

- Tomando uma decisão: `if` e `else` (Cap. 3.2)
- Indentação: a estrutura do bloco em Python
- Testes em sequência: `elif` (3.3)
- Condições compostas: `and`, `or`, `not` (3.4)
- Decisões aninhadas

Seleção: o programa escolhe o caminho



A condição é uma expressão cujo valor é `True` ou `False` (comparações da Aula 5).

`if` simples e composto

```
numero = int(input("Digite um número inteiro: "))
if numero % 2 == 0:
    print(numero, "é par")
else:
    print(numero, "é ímpar")
```

- Os dois-pontos `:` abrem o bloco; o *recuo* (4 espaços) define o que pertence a ele
- `else` é opcional: sem ele, nada acontece quando a condição é falsa
- `numero % 2 == 0`: o resto da divisão por 2 identifica números pares

Fonte: Exercício 3.4 do livro

Indentação é sintaxe

```
lado = float(input("Lado: "))
   area = lado * lado      # recuo inesperado
print("Área:", area)
```

```
IndentationError: unexpected indent
```

- Em C ou Java as chaves { } delimitam blocos; em Python é a **indentação**
- Misturar tabulação e espaços gera erros difíceis de ver: configure o editor para usar 4 espaços
- Exercício 3.3: provoque propositalmente erros de indentação e leia as mensagens

Múltiplo de outro número

```
numero1 = int(input("Digite o primeiro número: "))
numero2 = int(input("Digite o segundo número: "))
if numero2 % numero1 == 0:
    print(numero2, "é múltiplo de", numero1)
else:
    print(numero2, "não é múltiplo de", numero1)
```

E se `numero1` for 0?

`ZeroDivisionError`. Como evitar? Com outra decisão antes do cálculo, ou tratando a exceção (Aula 13).

Fonte: Exercício 3.5 do livro

Testes em sequência: `elif`

```
numero = int(input("Digite um número: "))
if numero % 3 == 0:
    print(numero, "é divisível por 3")
elif numero % 5 == 0:
    print(numero, "é divisível por 5")
elif numero % 7 == 0:
    print(numero, "é divisível por 7")
else:
    print(numero, "não é divisível por 3, 5 ou 7")
```

- Os testes são feitos em ordem; ao primeiro verdadeiro, os demais são ignorados
- Para 15 o programa só diz "divisível por 3". Como mostrar *todos*? (Exercício 3.6)

Aninhamento versus `elif`

```
nota1 = float(input("Qual é a primeira nota? "))
nota2 = float(input("E a segunda nota? "))
media = (nota1 + nota2) / 2
print("A média é ", media)
if media < 7:
    if media < 4:
        print("Ficou reprovado :(")
    else:
        print("Ficou para prova final")
else:
    print("Passou direto!")
```

Fonte: code/Cap3/prog.3.3_calcula_final_reprova.py

Condições compostas: and, or, not

A	B	A and B	A or B	not A
V	V	V	V	F
V	F	F	V	F
F	V	F	V	V
F	F	F	F	V

```
letra = input('Digite H para Homem e M para Mulher ')
if letra == 'H' or letra == 'h':
    print("Você escolheu Homem.")
elif letra == 'M' or letra == 'm':
    print("Você escolheu Mulher.")
else:
    print("Você escolheu outra opção.")
```

Fonte: code/Cap3/prog.3.5_h_e_m.py

Comparação encadeada: o desconto por idade

```
idade = int(input('Qual a idade? '))
if idade >= 0 and idade < 12:
    print('Desconto é de 100%')
elif idade >= 12 and idade < 26:
    print('Desconto é de 50%')
else:
    print('Sem Desconto')
```

Em Python é possível escrever de forma mais próxima da matemática:

```
if 0 <= idade < 12:
    print('Desconto é de 100%')
elif 12 <= idade < 26:
    print('Desconto é de 50%')
```

Fonte: `code/Cap3/prog.3.9, prog.3.10`

Desafio de lógica: o ano bissexto

Um ano é bissexto se for divisível por 4, **exceto** os divisíveis por 100, **a menos que** também sejam divisíveis por 400.

Ano	Bissexto?
2024	sim (div. por 4, não por 100)
1900	não (div. por 100, não por 400)
2000	sim (div. por 400)
2023	não

Tente: escreva a condição composta antes de ver a próxima página.

Ano bissexto: três versões

```
ano = int(input('Digite um ano: '))
if (ano % 4 == 0 and not(ano % 100 == 0)) or ano % 400 == 0:
    print('Ano é bissexto')
```

Versão mais legível (simplificando `not(...== 0)`) e com `else`:

```
ano = int(input('Digite um ano: '))
if (ano % 4 == 0 and ano % 100 != 0) or ano % 400 == 0:
    print('Ano é bissexto')
else:
    print('Ano não é bissexto')
```

Os parênteses explicitam a precedência: `and` é avaliado antes de `or`.

Fonte: `code/Cap3/prog.3.6_bissexto.py`, `prog.3.7`, `prog.3.8`

Exercícios resolvidos no site: leitura guiada

- **3.7** Aumento diferenciado de salários por faixa
- **3.8** Classificação do triângulo (equilátero, isósceles, escaleno)
- **3.9** Classificação por faixa etária
- **3.10** Enunciado atualizado: confira a versão do site

Roteiro para resolver qualquer exercício

1. Quais são as **entradas** e as **saídas**?
2. Quais são os **casos** (faixas, exceções)? Faça uma tabela.
3. Escreva o pseudocódigo, depois o Python.
4. **Teste** com um valor de cada caso e com os *limites*.

Para praticar: exercícios do livro

Site do livro – Capítulo 3

- Decisão simples e composta: **3.4, 3.5**
- Testes em sequência: **3.6**
- Condições compostas: **3.7, 3.8, 3.9, 3.10**

Enunciados e soluções comentadas em
`https://www.araujo.eng.uerj.br/livro/exercicios/cap3/`

Exercícios adicionais (sem solução): `https://www.araujo.eng.uerj.br/livro/extras/`

Resumo da aula

- `if/else` escolhe entre dois caminhos; `elif` encadeia vários
- O bloco é definido pela **indentação** (4 espaços)
- `and`, `or`, `not` combinam condições; use parênteses
- Teste sempre os valores nos *limites* das faixas
- Próxima aula: iteração (`while` e `for`)