

# Computação I - Python

## Aula 7 - Estrutura de Repetição com Teste de Parada: while

### Introdução

Apresentado por: Bernardo F. Costa

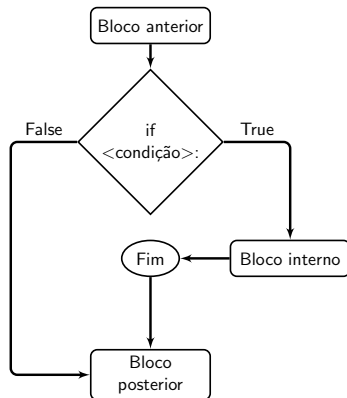
Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>



# Estrutura de Repetição *while*

- Lembra do **if**?
- Versão mínima (sem **else**):  
fluxograma ao lado
- Executa (ou não) o trecho interno
- Expressão booleana avalia condição e decide o fluxo da execução
- Decisão é tomada uma única vez



# Estrutura de Repetição *while*

- Uso específico: checagem simples para tarefa fixa com esforço conhecido
- Exemplos: conversão para valor absoluto, evitar divisão por zero
- E se quiséssemos repetir uma comparação semelhante?
- Computadores existem para fazer tarefas repetitivas

```
def ...  
    ...  
    # converter para valor absoluto  
    if numero < 0:  
        numero = -1 * numero  
    ...  
    # evitar divisão por zero  
    if divisor != 0:  
        quociente = dividendo // divisor  
        resto = dividendo % divisor  
    ...
```

# Estrutura de Repetição *while*

- Exemplo: cálculo de saldo em poupança
- Juros e quantidade de depósitos determinam saldo acumulado
- E se quiséssemos a quantidade de depósitos dado saldo acumulado?
- Como calcular essa quantidade?

```
def poupanca_3m(deposito, juros):  
    '''Calcula poupanca para tres meses  
    entrada: float, float  
    saida: float  
    '''  
    poupanca = 0  
    # tres depositos em sequencia  
    poupanca = poupanca * (1 + juros / 100) + deposito  
    poupanca = poupanca * (1 + juros / 100) + deposito  
    poupanca = poupanca * (1 + juros / 100) + deposito  
    return poupanca
```

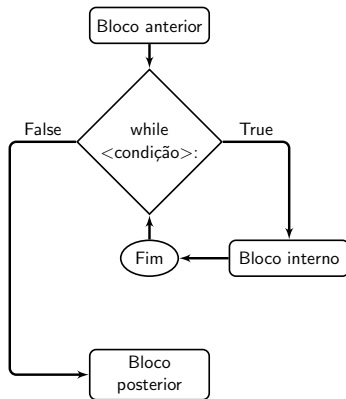
# Estrutura de Repetição *while*

- **if** pode testar uma condição de parada
- Teste limitado, solução exige repetição contínua
- Não resolve o problema

```
def total_meses_lim3(deposito, juros, saldo):  
    '''Calcula total de meses (ate 3) para atingir saldo desejado.  
    entrada: float, float, float  
    saida: int  
    '''  
  
    poupanca = 0  
    mes = 0  
    if (poupanca >= saldo):  
        return mes  
    mes = mes + 1  
    poupanca = poupanca * (1 + juros / 100) + deposito  
    # poupanca em um mes?  
    if (poupanca >= saldo):  
        return mes  
    mes = mes + 1  
    poupanca = poupanca * (1 + juros / 100) + deposito  
    # poupanca em dois meses?  
    if (poupanca >= saldo):  
        return mes  
    mes = mes + 1  
    poupanca = poupanca * (1 + juros / 100) + deposito  
    return mes
```

# Estrutura de Repetição *while*

- **if** não é adequado para repetições
- **while** muda um dos fluxos do **if**
- Expressão booleana avalia condição que decide o fluxo da execução
- Decisão é tomada na entrada e ao final de cada trecho interno
- Saída só é possível quando a condição for falsa
- Iteração: uma passagem dentro do **while**



# Estrutura de Repetição *while*

- **while**: vários **if**'s em sequência
- Implementa a continuidade de uma condição
- Executa qualquer quantidade de repetições
- Código se torna enxuto, sem duplicação

```
def total_meses_saldo(deposito, juros, saldo):  
    '''Calcula total de meses para atingir saldo desejado  
    entrada: float, float, float  
    saida: int  
    '''  
    poupanca = 0  
    mes = 0  
    while (poupanca < saldo):  
        mes = mes + 1  
        poupanca = poupanca * (1 + juros / 100) + deposito  
    return mes
```

## Autores

- **João C. P. da Silva** ▶ Lattes
- **Carla Delgado** ▶ Lattes
- **Ana Luisa Duboc** ▶ Lattes

## Colaboradores

- **Anamaria Martins Moreira** ▶ Lattes
- **Fabio Mascarenhas** ▶ Lattes
- **Leonardo de Oliveira Carvalho** ▶ Lattes
- **Charles Figueiredo de Barros** ▶ Lattes
- **Fabício Firmino de Faria** ▶ Lattes