

Estruturas de Repetição



Estruturas de repetição

- Permitem que um bloco de comandos seja executado diversas vezes
- **Dois tipos de Repetição:**
 - **Repetição condicional:** executa um bloco de código enquanto uma condição lógica for verdadeira (*while*)
 - **Repetição contável:** executa um bloco de código um número predeterminado de vezes (*for*)



Repetição condicional

Pseudocódigo

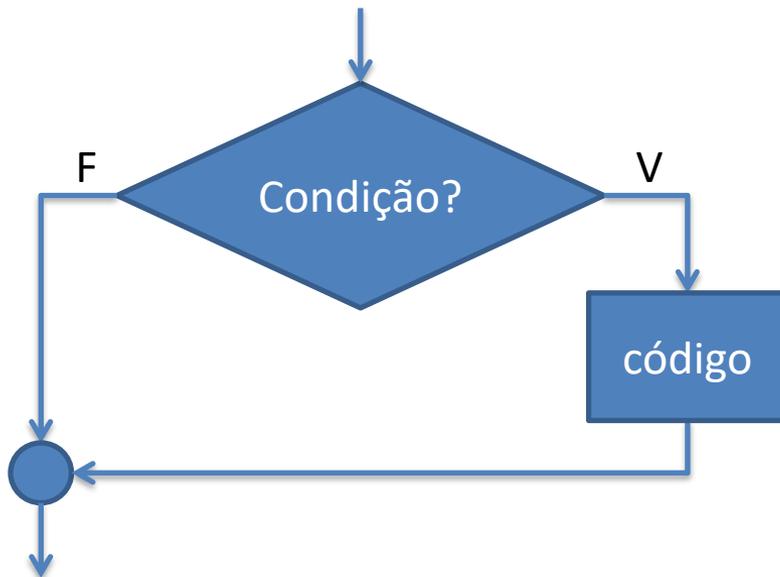
```
...  
enquanto CONDIÇÃO  
faça  
    INSTRUÇÃO 1;  
    INSTRUÇÃO 2;  
    ...  
    INSTRUÇÃO N;  
...  
...
```

Python

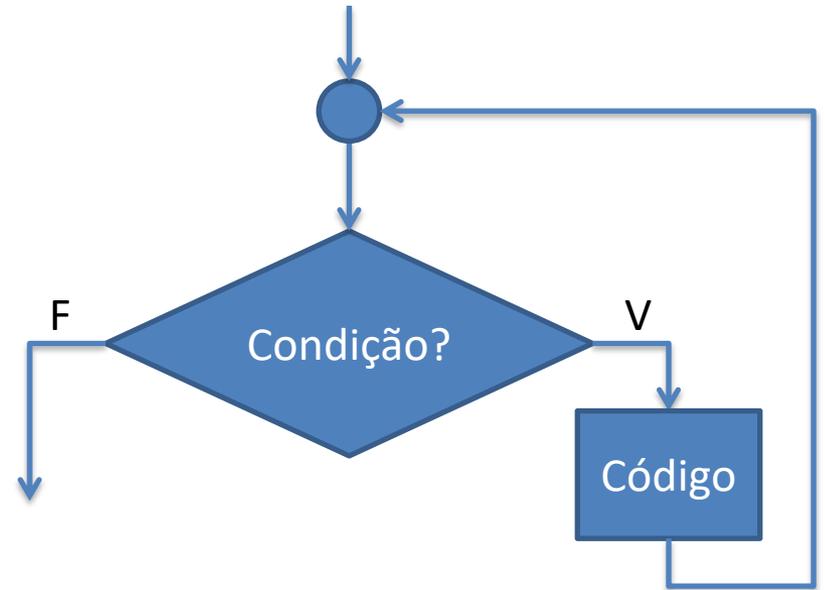
```
...  
while CONDIÇÃO:  
    INSTRUÇÃO 1;  
    INSTRUÇÃO 2;  
    ...  
    INSTRUÇÃO N;  
...
```

Repetição condicional

Se



Enquanto



Repetição condicional

- Executa o bloco de instruções **enquanto a condição for verdadeira**
- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores
- O bloco de instruções pode conter um ou mais comandos
- O início e o fim do bloco são definidos de acordo com a indentação

Repetição condicional

- Executa o bloco de instruções **enquanto a condição for verdadeira**
- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores
- O bloco de instruções pode conter um ou mais comandos
- O início e o fim do bloco são definidos de acordo com a indentação

Nenhuma novidade: igual ao if!!!

Repetição condicional

- A estrutura de repetição é chamada de **loop** porque continua-se voltando ao início da instrução até que a condição se torne falsa
- Deve haver alguma instrução dentro do bloco de comandos que torne a condição falsa para que a **repetição seja encerrada**
- Quando a condição se torna falsa, a **próxima instrução** após o bloco do while é executada
- Se a condição do while for falsa desde o início, o bloco de instruções **nunca é executado**.

Exemplo 1 – Bomba

```
numero = int(input('Digite um número: '))  
while numero > 0:  
    numero = numero - 1  
    print(numero)  
print('Boom!!!')
```

Exemplo 2 – Contador

- Programa que imprime a quantidade de números pares de 100 até 200, incluindo-os

Exemplo 2 – Contador

- Programa que imprime a quantidade de números pares de 100 até 200, incluindo-os

```
num = 100
contador_pares = 0
while num <= 200:
    if num % 2 == 0:
        contador_pares = contador_pares + 1
    num = num + 1
print(contador_pares)
```

Desafio

- Como fazer para contar a quantidade de números pares entre dois números quaisquer?

Desafio

- Como fazer para contar a quantidade de números pares entre dois números quaisquer?

```
num1 = int(input('Entre com o valor inicial: '))  
num2 = int(input('Entre com o valor final: '))  
contador_pares = 0  
while num1 <= num2:  
    if num1 % 2 == 0:  
        contador_pares = contador_pares + 1  
    num1 = num1 + 1  
print(contador_pares)
```

Exemplo 3 – Acumulador

- Programa que imprime a soma de todos os números pares entre dois números quaisquer, incluindo-os

```
num1 = int(input('Entre com o valor inicial: '))
num2 = int(input('Entre com o valor final: '))
soma = 0
while num1 <= num2:
    if num1 % 2 == 0:
        soma = soma + num1
    num1 = num1 + 1
print('A soma é', soma)
```

Exemplo 4 – Fatorial de um número

```
numero = int(input('Digite um número inteiro positivo: '))
fatorial = 1
while numero > 0:
    fatorial = fatorial * numero
    numero = numero - 1
print('O fatorial desse número é', fatorial)
```

Exemplo 5

- Qual a saída do programa abaixo?

```
i = 1
```

```
while True:
```

```
    i = i + 1
```

```
    print(i)
```

Evitem forçar loops infinitos!!!

Exercício

- Faça um programa que gere números inteiros aleatórios entre 1 e 10 e calcule a soma desses números, até que seja gerado um número **num** que foi informado pelo usuário anteriormente.
 - Dica 1: antes de mais nada, peça para o usuário digitar um número entre 1 e 10 e guarde o valor em **num**
 - Dica2: use a função `randint(inicio, fim)` do módulo `random` para gerar um número aleatório entre 1 e 10

Solução do exercício

```
import random

num = int(input('Digite um número inteiro entre 1 e 10: '))
soma = 0
numero_sorteado = random.randint(1,10)
print(numero_sorteado)
while num != numero_sorteado:
    soma = soma + numero_sorteado
    numero_sorteado = random.randint(1,10)
    print(numero_sorteado)
print('A soma é', soma)
```

Quantas vezes acontecerá essa repetição?
- Não é possível determinar de antemão

Repetição contável

- E se o enunciado fosse “Faça um programa que soma X números gerados aleatoriamente no intervalo de 1 a 10, onde X é informado pelo usuário”?

Repetição contável

```
import random

x = int(input('Digite um número: '))
soma = 0
contador = 0
while contador < x:
    numero_sorteado = random.randint(1,10)
    print(numero_sorteado)
    soma = soma + numero_sorteado
    contador = contador + 1
print('A soma é', soma)
```

Número de repetições é fixo.
- Python tem uma estrutura para isso!

Repetição contável

Pseudocódigo

```
...  
para VARIÁVEL variando de  
VALOR INICIAL a VALOR  
FINAL com passo  
INCREMENTO  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```

Python

```
...  
for VARIÁVEL in (faixa-  
de-valores):  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```

Faixa de valores

- Os valores podem ser listados explicitamente
- Exemplo:

```
for x in (0,1,2,3,4):  
    print(x)
```

print(x) será repetido 5 vezes,
uma para cada valor especificado
entre parênteses no comando for

Faixa de valores

- Os valores podem ser especificados como um intervalo com início, fim e incremento, usando **range**
 - Repete de *início* até *fim* - 1, com passo *incremento*

```
for i in range (0 , 5 , 1) :  
    print (i)
```

início (opcional)
quando omitido,
início = 0

fim (obrigatório)

incremento (opcional)
quando omitido,
incremento = 1

Exemplo de range

```
for i in range(5):  
    print(i)
```

0

1

2

3

4

Exemplo de range

```
for i in range(1, 5):  
    print(i)
```

1

2

3

4

Exemplo de range

```
for i in range(2, 10, 2):  
    print(i)
```

2

4

6

8

Exemplo de range

```
for i in range(10, 0, -2):  
    print(i)
```

10

8

6

4

2

Retomando o exemplo de soma de números aleatórios

- Faça um programa que soma X números gerados aleatoriamente no intervalo de 1 a 10, onde X é informado pelo usuário

Soma de números aleatórios com *while*

```
import random

x = int(input('Digite um número: '))
soma = 0
contador = 0
while contador < x:
    numero_sorteado = random.randint(1,10)
    print(numero_sorteado)
    soma = soma + numero_sorteado
    contador = contador + 1
print('A soma é', soma)
```

Vamos substituir por um
for e eliminar a necessidade de
controlar o contador

Soma de números aleatórios com *for*

```
import random

x = int(input('Digite um número: '))
soma = 0
for contador in range(x):
    numero_sorteado = random.randint(1,10)
    print(numero_sorteado)
    soma = soma + numero_sorteado
print('A soma é', soma)
```

Exemplo

- Programa que imprime a soma de todos os números pares entre dois números quaisquer, incluindo-os

```
num1 = int(input('Entre com o valor inicial: '))
num2 = int(input('Entre com o valor final: '))
soma = 0
for i in range(num1, num2 + 1):
    if i % 2 == 0:
        soma = soma + i
print('A soma é', soma)
```

Fatorial

- Programa para calcular fatorial de um número:

```
numero = int(input('Digite um inteiro positivo: '))
fatorial = 1
for i in range(1, numero + 1):
    fatorial = fatorial * i
print('O fatorial desse número é', fatorial)
```

Tabela de jogos

- Programa para gerar a tabela de jogos de um campeonato que tem 5 times (times jogam em casa e na casa do adversário)

```
for time1 in ('Fla', 'Flu', 'Bot', 'Vas', 'Ame'):  
    for time2 in ('Fla', 'Flu', 'Bot', 'Vas', 'Ame'):  
        if time1 != time2:  
            print(time1, 'x', time2)
```

Agenda

- Programa para imprimir uma agenda diária, com horários de 15 em 15 minutos

```
for hora in range(24):  
    for minuto in range(0,60,15):  
        print(str(hora) + ':' + str(minuto))
```

Exercícios

1. Faça um programa para montar a tabela de multiplicação de números de 1 a 10 (ex.: $1 \times 1 = 1$, $1 \times 2 = 2$, etc.)
2. Faça um programa para determinar o número de dígitos de um número inteiro positivo informado
3. Faça um programa para calcular a série de Fibonacci para um número informado pelo usuário, sendo $F(0) = 0$, $F(1) = 1$ e $F(n) = F(n-1) + F(n-2)$
 - Por exemplo, caso o usuário informe o número 9, o resultado seria:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Exercícios

4. Faça um programa para listar todos os divisores de um número ou dizer que o número é primo caso não existam divisores
 - Ao final, verifique se o usuário deseja analisar outro número

Exercícios

5. Faça um programa que calcule o retorno de um investimento financeiro fazendo as contas mês a mês, sem usar a fórmula de juros compostos
- O usuário deve informar quanto será investido por mês e qual será a taxa de juros mensal
 - O programa deve informar o saldo do investimento após um ano (soma das aplicações mês a mês considerando os juros compostos), e perguntar ao usuário se ele deseja que seja calculado o ano seguinte, sucessivamente
 - Por exemplo, caso o usuário deseje investir R\$ 100,00 por mês, e tenha uma taxa de juros de 1% ao mês, o programa forneceria a seguinte saída:

Saldo do investimento após 1 ano: R\$ 1268.25

Deseja processar mais um ano? (S/N)

Exercícios

6. Escreva um programa que imprime na tela os n primeiros números perfeitos. Um número perfeito é aquele que é igual à soma dos seus divisores. Por exemplo, $6 = 1 + 2 + 3$.
7. Um número inteiro pode ser igual ao produto de 3 números inteiros consecutivos, como, por exemplo, $120 = 4 \times 5 \times 6$. Elabore um programa que, após ler um número n do teclado, verifique se n tem essa propriedade.

Exercícios

8. Elabore um programa que leia n valores e mostre a soma de seus quadrados.
9. Faça um programa que leia dois valores x e y , e calcule o valor de x dividido por y , além do resto da divisão. Não é permitido usar as operações de divisão e resto de divisão do Python (use apenas soma e subtração).

Exercícios

10. Faça um programa em Python que calcule o valor de Pi, utilizando a fórmula de Leibniz

$$\pi/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + 1/13 - \dots$$

Adicione parcelas no cálculo até que a diferença de uma interação para a seguinte seja menor do que um valor de erro aceitável x informado pelo usuário.

Referências

- Slides feitos em conjunto com Aline Paes e Vanessa Braganholo

Estruturas de Repetição

