

Fundamentos de Programação

Prof. Márcio Miguel Gomes

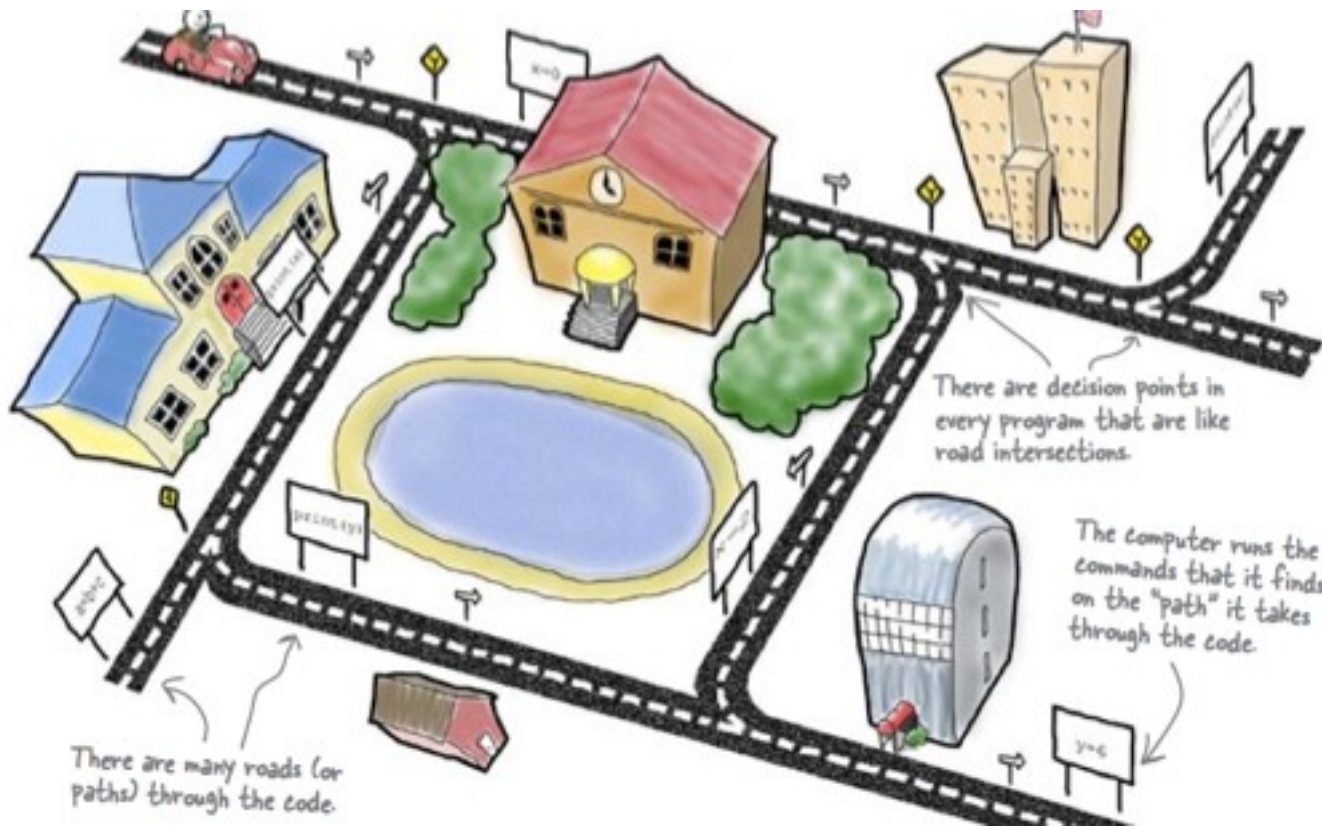


JESUÍTAS BRASIL



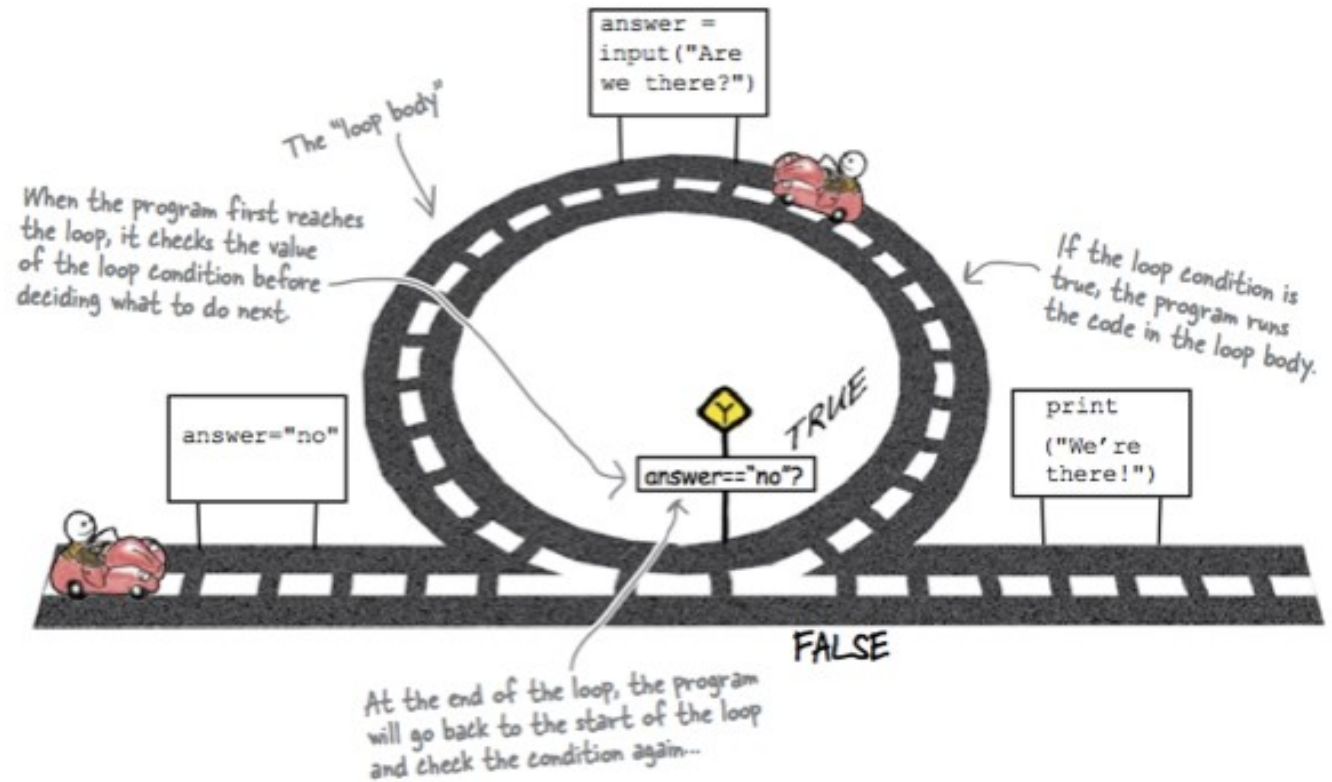
Estrutura de Repetição

A execução de um código é semelhante a um veículo transitando por uma cidade



Estrutura de Repetição

Em um determinado momento, chegamos em uma rotatória. Devemos entrar? Quantas vezes?



Estruturas de Repetição

- Alguns algoritmos podem ser muito repetitivos, por exemplo:
 - Geração de sequências numéricas
 - Números naturais, pares, ímpares
 - Validações por tentativa e erro
 - Números primos, MMC
 - Processamento em lote
 - Entrada de dados em sequência, máximo, mínimo

Estruturas de Repetição

- Permitem que uma ou mais instruções sejam executadas
 - Uma **quantidade** determinada de vezes

```
para nr de 1 até 10 passo 1:  
    mostre nr
```

- Enquanto uma determinada **condição** for satisfeita

```
resp = 'x'  
enquanto !(resp == 's' ou resp == 'n')  
    resp = leia('Deseja continuar? s/n')
```

Estruturas de Repetição

- **Contada**

- Repetem uma quantidade exata (pré-definida) de vezes
- São usadas quando se sabe previamente quantas vezes deve-se executar as instruções do bloco de repetição

- **Condicional**

- Repetem enquanto uma condição de repetição é atendida
- São usadas quando não se sabe previamente quantas vezes deve-se executar as instruções do bloco de repetição

Estruturas de Repetição

- **Contada**

- Realiza repetições em quantidades predeterminadas
- Usa uma variável de controle do tipo “contador” ou
- Incorpora automaticamente o funcionamento de um contador
- Também pode percorrer uma lista de valores fixos

Estruturas de Repetição

- **Contada - Comando “for” com “range”**

```
print("\nQuantidade fixa:")  
for x in range(5):  
    print(x)
```

```
print("\nInício e fim do intervalo:")  
for x in range(10, 15):  
    print(x)
```

```
print("\nInício, fim e passo do intervalo:")  
for x in range(20, 30, 3):  
    print(x)
```

```
print("\nInício, fim e passo do intervalo  
decrecente:")  
for x in range(30, 20, -3):  
    print(x)
```


Estruturas de Repetição

- **Contada - Comando “for” com “iterador”**

```
print("\nColeção de números fixos")
for x in (1, 2, 3, 5, 8):
    print(x)

print("\nColeção de textos fixos")
for x in ['banana', 'laranja', 'abacaxi', 'uva']:
    print(x)
```

Estruturas de Repetição

- **Condicional**

- Realiza repetições em quantidades indeterminadas
- Variável de controle do tipo “acumulador” ou “registrador”
- Utiliza uma expressão lógica para decidir se permanece no laço
- Avalia a expressão lógica a cada iteração do laço

Estruturas de Repetição

- **Condicional - Comando “while” com sequências**

```
print("\nQuantidade fixa:")
i = 0
while i < 5:
    print(i)
    i += 1

print("\nInício e fim do intervalo:")
i = 10
while i < 15:
    print(i)
    i += 1
```

```
print("\nInício, fim e passo do intervalo:")
i = 20
while i < 30:
    print(i)
    i += 3

print("\nInício, fim e passo do intervalo
decrecente:")
i = 30
while i > 20:
    print(i)
    i -= 3
```

Estruturas de Repetição

- **Condicional - Comando “while” com interação do usuário**

```
print('\nDigitar valores até atingir um somatório:')
soma = 0
while soma < 100:
    soma += int(input('Digite um número: '))
print('A soma dos valores digitados é', soma)

print('\nSolicitar um valor pertencente a um domínio:')
sexo = ''
while sexo not in ('m', 'M', 'f', 'F'):
    sexo = input('Informe um sexo (m/f): ')
print('Sexo informado:', sexo)
```

Estruturas Aninhadas

- É possível combinar estruturas de repetição em 2 ou mais níveis
- Também pode-se usar estruturas de seleção dentro de laços

```
print('\nLaços aninhados')
for letra in range(ord('A'), ord('D')):
    for numero in range(1, 4):
        print(chr(letra), numero)
    print()
```

```
print('\nDigite 5 letras ou números')
for x in range(1, 6):
    valor = input('{}: '.format(x))
    if valor.isnumeric():
        print('Numero')
    else:
        print('Letra')
```

Intervenções no Fluxo

- **continue:** suspende a execução corrente e pula para o próximo valor da sequência ou reavalia a condição de permanência

```
print('\nImprime valores de 1 a 10 exceto múltiplos de 3:')  
for x in range(1, 11):  
    if x % 3 == 0:  
        continue  
    print(x)
```

Intervenções no Fluxo

- **break:** encerra a execução das instruções e transfere o controle para fora do laço

```
print('\nMenor divisor de um número')
nr = int(input('Informe um número inteiro: '))
for x in range(2, nr + 1):
    if nr % x == 0:
        break

print(nr, 'é divisível por', x)
```

Exercícios

1. Desenvolva um programa que imprima na tela uma sequência com os “n” primeiros números naturais. Peça para o usuário informar o valor de “n” antes de gerar a sequência. Utilize tanto uma estrutura contada quanto condicional
2. Desenvolva um programa que imprima na tela uma sequência com os “n” primeiros números naturais pares. Peça para o usuário informar o valor de “n” antes de gerar a sequência. Utilize tanto uma estrutura contada quanto condicional
3. Desenvolva um programa que imprima na tela uma sequência com os “n” primeiros números naturais ímpares. Peça para o usuário informar o valor de “n” antes de gerar a sequência. Utilize tanto uma estrutura contada quanto condicional
4. Desenvolva um programa que solicite um número inteiro “n” e imprima na tela o seu fatorial. Fatorial é o produto dos números inteiros consecutivos de 1 até um dado inteiro “n”. Utilize tanto uma estrutura contada quanto condicional

Exercícios

5. Desenvolva um programa que imprima na tela uma sequência com os “n” primeiros números fracionários múltiplos de 0,5 iniciando em 1,5. Peça para o usuário informar o valor de “n” antes de gerar a sequência.

Ex: “n” = 5; sequência = 1,5 2,0 2,5 3,0 3,5

6. Desenvolva um programa que imprima na tela uma sequência com os “n” primeiros números fracionários com passo “p” iniciando em um número “i”. Peça para o usuário informar os valores de “n”, “p” e “i” antes de gerar a sequência.

Ex: “n” = 6, “p” = 0,3 e “i” = -1; sequência = -1,0 -0,7 -0,4 -0,1 0,2 0,5

Exercícios

7. Desenvolva um programa que solicite uma quantidade indeterminada de números positivos ao usuário, e ao digitar o número zero, calcule e informe:
- A quantidade de números digitados
 - A quantidade de pares
 - A quantidade de ímpares
 - A soma destes números
 - A média aritmética
 - O maior número
 - O menor número

Exercícios

8. Desenvolva um programa que imprima na tela a seguinte forma:

```
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

Obs.: Utilizando laços e estruturas de controle;

Exercícios

9. Desenvolva um programa que imprima na tela a seguinte forma:

```
*  
* *  
* * *  
* * * *  
* * * * *
```

Obs.: Utilizando laços e estruturas de controle;

Exercícios

10. Desenvolva um programa que imprima na tela a seguinte forma:

* * * * *

* * * *

* * *

* *

*

Obs.: Utilizando laços e estruturas de controle;

Referências

- https://www.w3schools.com/python/python_for_loops.asp
- https://www.w3schools.com/python/python_while_loops.asp