



Pensamento Computacional

2023

Estruturas de repetição

- Permite a execução, de forma repetida e controlada, de um bloco de comandos.
- Existem **várias estruturas** de repetição. Durante a programação, deve-se optar pela estrutura que mais convém com a necessidade (após o estudo de todas as estruturas, retornaremos a este ponto).

Função range

- A função range será utilizada para especificar o intervalo de repetição do for.
- Formato geral:
range(fim)
range(inicio, fim)
range(inicio, fim, passo)

Exemplos:

```
range(10)  
[0,1,2,3,4,5,6,7,8,9]  
range(10, 15, 1)  
[10,11,12,13,14]  
range(2, 10, 2)  
[2,4,6,8]  
range(10, 0, -1)  
[10,9,8,7,6,5,4,3,2,1]
```

Estruturas de repetição com variável de controle

- O laço de repetição **for** realiza repetições de acordo com uma variável de controle
- Exemplo:

```
for contador in range(10):  
    print(contador)
```

Importante: a função range cria uma sequência de valores. No caso acima, os valores vão de 0 a 9

Percorrer um vetor indexando

```
frutas = ['laranja', 'maça', 'pera', 'banana', 'kiwi', 'maça',  
'banana']
```

```
for indice in range(len(frutas)):  
    print(frutas[indice])
```

Percorrer um vetor atribuindo valor

```
frutas = ['laranja', 'maça', 'pera', 'banana', 'kiwi', 'maça',  
'banana']
```

```
for elemento in frutas:  
    print(elemento)
```

Comparativo

for

```
soma = 0
```

```
for cont in range(1, 11, 1):  
    soma = soma + cont
```

```
print("Soma = ", soma)
```

while

```
cont = 1  
soma = 0
```

```
while cont < 11:  
    soma = soma + cont  
    cont = cont + 1
```

```
print("Soma = ", soma)
```

?

```
nota = float(input("Digite a nota  
(entre 0 e 10) = "))
```

```
while nota < 0 or nota > 10:  
    print("Nota invalida.")  
    nota = float(input("Digite a nota  
(entre 0 e 10) = "))
```

Comparativo (cont.)

- Note que o **for** é usado quando sabemos, de forma exata, quantas vezes a repetição deve ser realizada
- Caso esse número não seja conhecido, devemos usar um laço de repetição do tipo **while**

Exercícios

1. Escreva um algoritmo que leia um vetor com 4 valores e apresente sua média
2. Escreva um algoritmo que crie uma lista de tamanho 5 e imprima seus valores e em seguida a soma dos valores pares e ímpares
3. Ordene um vetor de 100 números inteiros gerados aleatoriamente.

Exercícios

- Implemente um programa em Python para verificar quantos números uma aposta acertou na Megasena. O programa deve ler do teclado os 6 números apostados e comparar com 6 números sorteados. Ao final o programa deve exibir os números sorteados, números jogados e quantidade de acertos.

Desafio

Desenvolva um programa para o “**Jogo da Forca**”. Para tal, crie um vetor de strings inicializado com um conjunto de palavras-chave (por exemplo: nomes de capitais do Brasil, ou times de futebol ou Países da América do Sul etc). Sorteie uma das palavras para ser o segredo e forneça seis vidas para o usuário acertar o segredo. A cada rodada informe o número de vidas disponíveis e a disposição das letras acertadas e ausentes na palavra segredo (lembre de quando brincava com este jogo em caderno na infância), mostre também quais as letras que já foram usadas (e não compute acerto ou erro no caso do usuário repetir uma letra já fornecida).

Bibliografia

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. Algoritmos: lógica para desenvolvimento de programação de computadores. 24. ed., rev. São Paulo: Érica, 2010.

MENEZES, Nilo Ney Coutinho. Introdução à programação com Python: algoritmos e lógica de programação para iniciantes. 1. ed. São Paulo: Novatec, 2010.



PUCPR
GRUPO MARISTA

Contato:

vilmar.abreu@pucpr.br