



Programação de Aplicativos

Professor: Euclides Paim
euclidespaim@gmail.com



Laços de Repetição

For Loops

Professor: Euclides Paim
euclidespaim@gmail.com



Programação de Aplicativos

Laços de Repetição

Sumário

- Revisão
- *For Loops*
- Sintaxe do *For*
- Exercícios de fixação
- Resumo da aula





Programação de Aplicativos

Laços de Repetição

Laços (*Loops*) em Python

O ***Python*** possui dois comandos primitivos de repetição:

- laços **while**
- laços **for**



Programação de Aplicativos

Laços de Repetição

O laço *for*

- Com o laço **for**, podemos executar um conjunto de instruções enquanto uma condição for verdadeira.

Sintaxe:

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```

Observação: O laço **for** não exige que uma variável de índice seja definida previamente.



Programação de Aplicativos

Laços de Repetição

O laço *for*

Isso é menos parecido com a palavra-chave **for** em outras linguagens de programação, e funciona **mais como um método de iterador**, como é encontrado em outras linguagens de programação orientadas a objetos.

Com o laço **for**, podemos executar um conjunto de instruções **uma vez para cada item em uma lista, tupla, conjunto (set), etc.**



Programação de Aplicativos

Laços de Repetição

Percorrendo uma String

Mesmo as **strings** são **objetos iteráveis**, pois contêm uma **sequência de caracteres**.

Exemplo:

Percorrer as letras da palavra "banana":

```
for x in "banana":  
    print(x)
```



Programação de Aplicativos

Laços de Repetição

A Função `range()`

Para executar um conjunto de comandos um **número específico de vezes**, podemos usar a função `range()`.

A função `range()` retorna uma **sequência de números**, começando em **0 por padrão**, incrementando de **1 em 1 (também por padrão)**, e **terminando em um número especificado (o número final não é incluído)**.

Exemplo:

```
for x in range(6):  
    print(x)
```

Observe que `range(6)` não representa os valores de 0 a 6, mas sim os valores de 0 até 5.



Programação de Aplicativos

Laços de Repetição

Sintaxe do **range()**

```
range(início, fim, passo)
```

Onde:

- **início** → (opcional) número de onde começa. Se não for informado, começa do 0.
- **fim** → (obrigatório) até onde vai, **mas não inclui esse número**.
- **passo** → (opcional) de quanto em quanto ele vai contar. O padrão é 1.



Programação de Aplicativos

Laços de Repetição

Resumo: Laço **for** em Python

- Laço de **repetição controlada**.
- Percorre elementos de uma **sequência** (listas, strings, range...).
- Executa o bloco de código para **cada item** da sequência.
- Útil quando sabemos **quantas vezes** vamos repetir.



That's all Folks!



Programação de Aplicativos

Sintaxe, Variáveis e Tipos de Dados

Exercícios:

Exercício 1: Contador simples (nível fácil)

Enunciado:

Escreva um programa que use um laço `while` para exibir os números de 1 a 10 na tela.

Exemplo de saída esperada:

```
1
2
3
...
10
```



Programação de Aplicativos

Sintaxe, Variáveis e Tipos de Dados

Exercícios:

Exercício 2: Peça para o usuário digitar números inteiros. O programa deve continuar pedindo números até que o usuário digite 0. Ao final, mostre a soma de todos os números digitados (exceto o zero).

Exemplo:

```
Digite um número: 5
Digite um número: -2
Digite um número: 3
Digite um número: 0
Soma total: 6
```



Programação de Aplicativos

Sintaxe, Variáveis e Tipos de Dados

Exercícios:

Exercício 3: Implemente um programa que solicite a senha de acesso ao usuário. O laço `while` deve continuar pedindo a senha até que o usuário acerte. A senha correta é `"python123"`.

Extra: Mostre uma mensagem de sucesso ao final.

Exemplo:

```
Digite a senha: admin
Senha incorreta.
Digite a senha: 1234
Senha incorreta.
Digite a senha: python123
Acesso concedido!
```



Referências

Referências Básicas

CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford. ***Algoritmos: teoria e prática***. 3. ed. Rio de Janeiro: Elsevier, 2013.

SEBESTA, Robert W. ***Conceitos de linguagens de programação***. 10. ed. São Paulo: Pearson, 2018.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. ***Algoritmos: lógica para desenvolvimento de programação de computadores***. 27. ed. São Paulo: Érica, 2016.

DOWNEY, Allen B. ***Pense em Python: como pensar como um cientista da computação***. Tradução de Cássio de Souza Costa. 2. ed. São Paulo: Novatec, 2016.

Referências Complementares

IEPSEN, Edécio Fernando. ***Lógica de programação e algoritmos com JavaScript***. 2. ed. São Paulo: Novatec, 2022.

Referências na Internet

<https://www.w3schools.com/python/>