



Programação de Aplicativos

Professor: Euclides Paim
euclidespaim@gmail.com



Laços de Repetição

While Loops

Professor: Euclides Paim

euclidespaim@gmail.com



Programação de Aplicativos

Estruturas de Controle

Sumário

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





Programação de Aplicativos

Estruturas de Controle

Laços (*Loops*) em Python

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

- laços **while**
- laços **for**



Programação de Aplicativos

Estruturas de Controle

O laço *while*

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

Sintaxe:

```
i = 1
while i < 6:
    print(i)
    i += 1
```

Observação: lembre-se de incrementar *i*, caso contrário, o laço continuará para sempre.

O laço *while* requer que as variáveis relevantes estejam preparadas. Neste exemplo, precisamos definir uma variável de indexação, *i*, que definimos como 1.



Programação de Aplicativos

Estruturas de Controle

A instrução *break*

Com a instrução `break`, podemos interromper o laço mesmo que a condição do `while` ainda seja verdadeira:

```
i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i += 1
```



Programação de Aplicativos

Estruturas de Controle

A instrução *continue*

Com a instrução `continue`, podemos interromper a iteração atual e continuar com a próxima:

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```



Programação de Aplicativos

Estruturas de Controle

A instrução `else`

Com a instrução `else`, podemos executar um bloco de código uma vez, quando a condição não for mais verdadeira:

```
i = 1
while i < 6:
    print(i)
    i += 1
else:
    print("i já não é menor que 6")
```





Programação de Aplicativos

Estruturas de Controle

Resumo: Laço **while** em Python

- O laço **while** é usado para repetir um bloco de código **enquanto uma condição for verdadeira**.
- Componentes importantes:
 - a. **Condição:** O **while** verifica a condição antes de cada repetição.
 - b. **Variáveis de controle:** Precisamos inicializar e atualizar o **i**, para não gerar um em *loop* infinito.
- **break:**
 - a. Interrompe o laço imediatamente, mesmo se a condição ainda for verdadeira.
- **continue:**
 - a. Pula para a **próxima iteração** do laço, ignorando o restante do código no bloco atual.
- **else com while:**
 - a. Executa um bloco de código **uma vez, quando** a condição do **while** deixar de ser verdadeira (e o laço não for interrompido por **break**).



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/>