

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**

Estruturas de repetição

Laço de repetição: *FOR*

Aula 1

Código da aula: [SIS]ANO1C1B2S14A1

Exposição



Objetivo da Aula

- Compreender aspectos básicos do laço *FOR*, destacando sua importância como ferramenta fundamental em programação para controlar o fluxo de repetição.



Competências da Unidade (Técnicas e Socioemocionais)

- Desenvolver sistemas computacionais, utilizando ambiente de desenvolvimento;
- Migrar sistemas, implementando rotinas e estruturas de dados mais eficazes;
- Trabalhar a criatividade e o comprometimento na resolução de problemas computacionais.



Recursos Didáticos

- Recurso audiovisual para a exibição de vídeos e imagens;
- Folhas sulfite, canetas coloridas, lápis.



Duração da Aula

50 minutos

Fundamentos do laço *FOR*

Vamos aprender os aspectos básicos do laço *FOR*, destacando sua importância como ferramenta fundamental em programação para controlar o fluxo de repetição.

- ✓ Sintaxe do laço *FOR*.
- ✓ Fluxo de execução (início, condição, incremento).
- ✓ Aplicações básicas do laço *FOR*.

Exposição

Vamos lembrar o que são estruturas de repetição?



Estruturas de repetição, também conhecidas como **laços ou loops**, são construções fundamentais para a programação que permitem executar um bloco de código repetidamente sob certas condições.

Elas são essenciais para realizar tarefas que envolvem processamento repetitivo e automatizado de dados ou outras operações.

© Getty Images

Por que existem estruturas de repetição?

- **Automatização de tarefas repetitivas:** sem estruturas de repetição, seria necessário escrever o mesmo código várias vezes para executar tarefas repetitivas. Os loops permitem simplificar isso, reduzindo a redundância e melhorando a eficiência do código.
- **Processamento das coleções de dados:** em muitas situações, é necessário processar itens em uma coleção de dados, como listas ou arquivos. Estruturas de repetição garantem percorrer esses itens de maneira sistemática.
- **Condições dinâmicas:** loops podem ser usados para executar um bloco de código até que uma determinada condição seja atendida, o que é útil em cenários nos quais o número de iterações não é conhecido antecipadamente.

Exemplos de aplicação das estruturas de repetição no mercado

- **Processamento de dados em lote:** em big data e análise de dados, loops são usados para processar grandes volumes de dados, como em operações de limpeza de dados, análise estatística ou transformações de dados.
- **Desenvolvimento de jogos:** em jogos, estruturas de repetição são essenciais para implementar ciclos de jogo (game loops), atualizando constantemente o estado do jogo e a renderização gráfica.

Exemplos de aplicação das estruturas de repetição no mercado

- **Automação de tarefas de TI:** em automação de TI, scripts que utilizam loops são frequentemente empregados para executar tarefas rotineiras, como backups de dados, monitoramento de sistemas ou processamento de logs.
- **Aplicações web e servidores:** loops são usados para manipular requisições de usuários, processar dados de formulários ou gerenciar sessões de usuários em aplicações web.
- **Análise financeira:** no setor financeiro, algoritmos que incluem laços são usados para analisar tendências de mercado, realizar cálculos de risco e processar transações em grande escala.

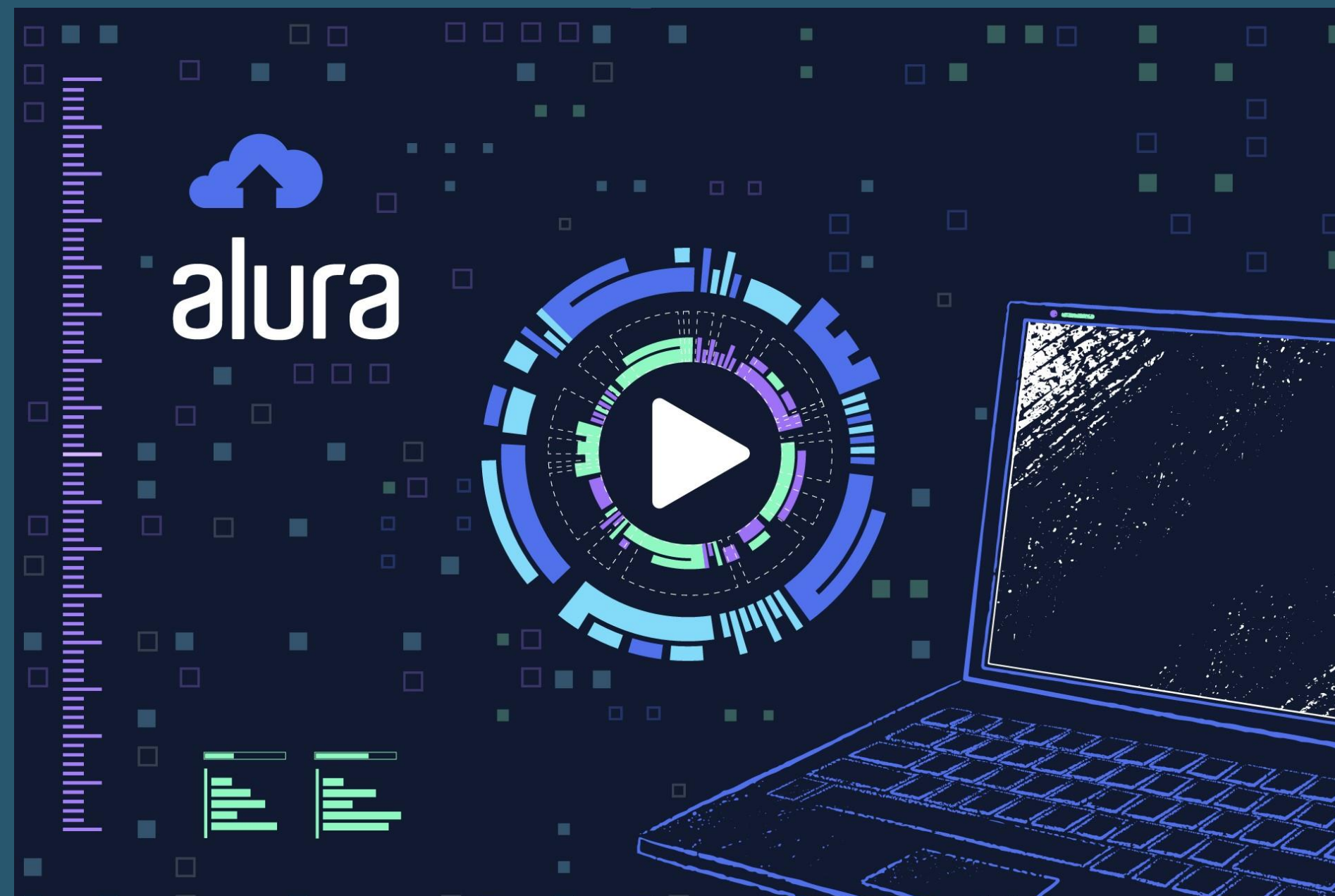


Conheça um projeto em Python que utiliza os conceitos que vimos:



© Getty Images

Exposição



ALURA. *Python: começando com a linguagem*. 03 Comparando variáveis. Disponível em: <https://cursos.alura.com.br/course/python-introducao-a-linguagem/task/22779>. Acesso em: 28 mar. 2024.

Sintaxe e funcionamento detalhado do laço *FOR*

O laço *FOR* em Python é uma estrutura de controle que viabiliza iterar sobre sequências, como listas, strings, tuplas ou objetos iteráveis.

Sintaxe básica:

```
for variavel in sequencia:  
    # executa alguma ação
```

Aqui, a variável é **temporária** e assume o valor de cada elemento da sequência a cada iteração do laço. No código abaixo, a variável *letra* assume sucessivamente os valores **'P', 'y', 't', 'h', 'o', 'n'** e, a cada iteração, imprime a letra atual.



Exemplo

```
for letra in "Python":  
    print(letra)
```


Exposição

Fluxo de execução em profundidade

O laço *FOR* segue um padrão específico de execução:

1

Inicialização

Ao encontrar um laço *FOR*, o Python inicia a iteração no primeiro elemento da sequência.

2

Atribuição

A variável de loop (ex.: letra) recebe o valor do elemento atual da sequência.

3

Execução do bloco de código

O Python executa o bloco de código dentro do laço com a variável atual.

4

Incremento e continuação

A variável do laço é automaticamente atualizada para o próximo elemento da sequência. Se ainda houver elementos, o passo 3 é repetido.

5

Término do laço

Quando a sequência é totalmente percorrida, o laço *FOR* é concluído.

Fluxo de execução em profundidade

Exemplo:

```
for i in range(3): # range(3) gera 0, 1, 2  
    print(i)
```

Aqui, *i* começa em 0 e incrementa em 1 a cada iteração até que a sequência `range(3)` seja totalmente percorrida.



Tome nota

O comando `range` é usado para gerar uma sequência de números. É frequentemente utilizado em loops (como *for* loops) para repetir um bloco de código em um número específico de vezes.

Aplicações avançadas e contextualização no mercado

O laço *FOR* tem ampla aplicação no desenvolvimento de software e análise de dados. Vamos explorar alguns usos avançados:

- **Iteração em dicionários**

Útil para acessar chaves e valores em dicionários, estruturas de dados fundamentais para armazenar e processar informações de modo eficiente.

```
dados = {"nome": "Alice", "idade": 30}
for chave, valor in dados.items():
    print(f"{chave}: {valor}")
```

© Getty Images

Aplicações avançadas e contextualização no mercado

- **List comprehensions**

Compreende uma maneira poderosa e concisa de criar novas listas, muito utilizada em manipulação de dados.

```
quadrados = [x**2 for x in range(10)]
```

- **Laços aninhados**

Geralmente usados em aplicações que envolvem matrizes ou tabelas de dados.

```
matriz = [[1, 2], [3, 4]]  
for linha in matriz:  
    for elemento in linha:  
        print(elemento)
```




Vamos
fazer um
quiz

Assinale a alternativa correta.
Qual é a principal função de um laço
***FOR* em Python?**

Executar um bloco de código uma única vez.

Executar um bloco de código um número indefinido de vezes até que uma condição seja atendida.

Executar um bloco de código para cada elemento de uma sequência.

Dividir um programa em funções menores para melhor organização do código.



Vamos
fazer um
quiz

Assinale a alternativa correta. Qual é a principal função de um laço *FOR* em Python?

Executar um bloco de código uma única vez.

Executar um bloco de código um número indefinido de vezes até que uma condição seja atendida.



Executar um bloco de código para cada elemento de uma sequência.

Dividir um programa em funções menores para melhor organização do código.

RESPOSTA CORRETA!

Laços *FOR* em Python são usados para iterar sobre elementos de uma sequência (como listas, strings ou tuples), executando um bloco de código para cada elemento.



Vamos
fazer um
quiz

Assinale a alternativa correta.
Em um laço *FOR*, o que acontece quando a sequência sobre a qual estamos iterando chega ao fim?

O laço reinicia automaticamente do primeiro elemento da sequência.

O programa gera um erro e para de executar.

O laço termina e o programa continua a executar o código após o laço.

O último elemento da sequência é processado indefinidamente.



Vamos
fazer um
quiz

Assinale a alternativa correta. Em um laço *FOR*, o que acontece quando a sequência sobre a qual estamos iterando chega ao fim?

O laço reinicia automaticamente do primeiro elemento da sequência.

O programa gera um erro e para de executar.

✓ O laço termina e o programa continua a executar o código após o laço.

O último elemento da sequência é processado indefinidamente.

RESPOSTA CORRETA!

Quando um laço *FOR* termina de iterar sobre todos os elementos da sequência, ele é concluído e o programa prossegue para a próxima linha de código.



Vamos
fazer um
quiz

Assinale a alternativa correta.
Qual das seguintes opções é um uso comum de laços *FOR* em Python?

Conectar-se com um banco de dados.

Iterar sobre os elementos de uma lista para processá-los individualmente.

Criar uma nova instância de uma classe.

Capturar exceções e erros em um bloco de código.



Vamos
fazer um
quiz

Assinale a alternativa correta.
Qual das seguintes opções é um uso comum de laços *FOR* em Python?

Conectar-se com um banco de dados.

Iterar sobre os elementos de uma lista para processá-los individualmente.

Criar uma nova instância de uma classe.

Capturar exceções e erros em um bloco de código.

RESPOSTA CORRETA!

Uma aplicação comum de laços *FOR* é iterar sobre cada elemento de uma lista (ou qualquer outra sequência) para realizar algum tipo de processamento em cada elemento.



O que nós
**aprendemos
hoje?**

Hoje desenvolvemos:

- 1 Revisão do que são as estruturas de repetição e seu funcionamento dentro dos sistemas de software.
- 2 O conceito de fluxo de execução (início, condição, incremento).
- 3 Análise de exemplos simples de aplicações que utilizam o laço *FOR*.

© Getty Images



Saiba mais

Durante esta aula, vimos que a linguagem Python pode ser utilizada no contexto de diferentes linguagens de programação.

Que tal conhecer mais sobre sua aplicação na linguagem C#?

SHARPAX. *Aula 10 – Estrutura de repetição para (FOR) | Lógica de Programação*. Disponível em: <https://www.youtube.com/watch?v=rRWDVXFj0gk>. Acesso em: 28 mar. 2024.

```
void _decode_(char cbuff **buff)
{
    if (step == AES_LOC_PASS) {
        src = cbuff->load();
        dest = getattr(&ptr, &mod,
            if (mod != NULL) as dest)
        dest += buffer->TABLE[mod]
        mask |= (0xFFFF & 0x00000000)
        if (mask & SIG_KERNEL) {
            return _ERROR_
        }
        return mask
    }
}
```

Referências da aula

ALURA. *Python: começando com a linguagem*. 03 Comparando variáveis. Disponível em: <https://cursos.alura.com.br/course/python-introducao-a-linguagem/task/22779>. Acesso em: 28 mar. 2024.

SHARPAX. *Aula 10 – Estrutura de repetição para (FOR) | Lógica de Programação*. Disponível em: <https://www.youtube.com/watch?v=rRWDVXFj0gk>. Acesso em: 28 mar. 2024.

ZANELALO, J. *Lógica de programação – Estruturas de repetição*. PodProgramar, 25 fev. 2018. Disponível em: <https://podprogramar.com.br/logica-de-programacao-estruturas-de-repeticao/>. Acesso em: 28 mar. 2024.

Identidade visual: imagens © Getty Images.

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**