**Nome:** Aline Neves de Melo **RA:** 0220482412005

**Disciplina:** Algoritmos e Lógica de Programação

**Tarefa Para Casa 01 - Para Aula 12 – Prazo final: 02/05/24 às 12h50**

**Descrição:** Responda as perguntas a seguir:

1) Explique a sintaxe e o funcionamento do FOR no python.

2) Explique a sintaxe do FOR no Java.

3) Dê exemplo do FOR no java compare com o python.

**1) Explique a sintaxe e o funcionamento do FOR no python.**

O loop "for" em Python é uma estrutura de repetição usada para iterar sobre elementos de uma sequência. Ele é útil para automatizar tarefas que precisam ser executadas várias vezes com cada item da sequência.

Veja a sintaxe básica:

*for <variavel> in <iterável>:*

*# Bloco de código indentado*

<variavel>: É o nome da variável que vai receber cada elemento da sequência durante a iteração.

<iterável>: É a sequência de elementos que o loop irá percorrer. Pode ser uma lista, tupla, string, dicionário ou qualquer objeto que suporte iteração.

Durante a execução do loop, a variável <variavel> assume o valor de cada elemento presente no <iterável> a cada iteração. O bloco de código indentado após o : será executado para cada elemento.

Por exemplo, para percorrer uma lista de frutas e imprimir cada item:

frutas = [" maçã", " banana", " laranja"]

for fruta in frutas:

print(fruta)

Neste exemplo, a variável fruta assume o valor de cada fruta da lista frutas a cada vez que o loop roda. O código indentado (print(fruta)) imprime o nome de cada fruta.

Aqui estão alguns outros pontos importantes sobre o loop "for" em Python:

* Função range(): Você pode usar a função range() para criar uma sequência de números para iteração. Por exemplo:

*for i in range(5): # Irá iterar 5 vezes (0, 1, 2, 3, 4)*

*print(i)*

* Loop aninhado: Você pode ter loops "for" aninhados para iterar sobre sequências multidimensionais.
* Instruções *break* e *continue*: Você pode usar break para sair do loop antecipadamente e continue para pular para a próxima iteração.
* Loop "for/else": Você pode adicionar uma cláusula else após o loop para executar um bloco de código somente após a iteração completa.

**2) Explique a sintaxe do FOR no Java.**

O loop for em Java é uma estrutura de controle usada para executar um bloco de código repetidamente um número determinado de vezes ou até que uma condição seja satisfeita. É uma das estruturas de repetição mais comuns em Java e é utilizada para automatizar tarefas que precisam ser executadas várias vezes com base em um conjunto de dados.

A sintaxe básica do loop for em Java é a seguinte:

*for (inicialização; condição; incremento) {*

*// Bloco de código a ser executado*

*}*

Componentes da sintaxe:

* **inicialização:** Esta parte declara e inicializa a variável de controle do loop. Essa variável é geralmente um número inteiro que é usado para acompanhar a iteração do loop.
* **condição:** Esta parte define a condição que será avaliada antes de cada iteração do loop. Se a condição for verdadeira, o bloco de código será executado. Se a condição for falsa, o loop será encerrado.
* **incremento:** Esta parte é opcional e especifica como a variável de controle será atualizada após cada iteração do loop. Isso geralmente envolve incrementar o valor da variável por 1.

Exemplo de uso:

*for (int i = 0; i < 5; i++) {*

*System.out.println(i);*

*}*

Neste exemplo, o loop será executado 5 vezes. A variável de controle i será inicializada com 0, depois incrementada em 1 a cada iteração. O valor de i será impresso em cada iteração.

Variantes do loop for:

**Loop for com incremento:**

*for (int i = 10; i >= 0; i--) {*

*System.out.println(i);*

*}*

Neste exemplo, o loop será executado 11 vezes, decrementando o valor de i de 10 a 0.

**Loop for com array:**

*int[] numeros = {1, 2, 3, 4, 5};*

*for (int numero : numeros) {*

*System.out.println(numero);*

*}*

Neste exemplo, o loop irá iterar sobre os elementos do array numeros, imprimindo cada elemento na tela.

**Loop for com iterador personalizado:**

*List<String> nomes = Arrays.asList("João", "Maria", "Pedro");*

*for (String nome : nomes) {*

*System.out.println(nome);*

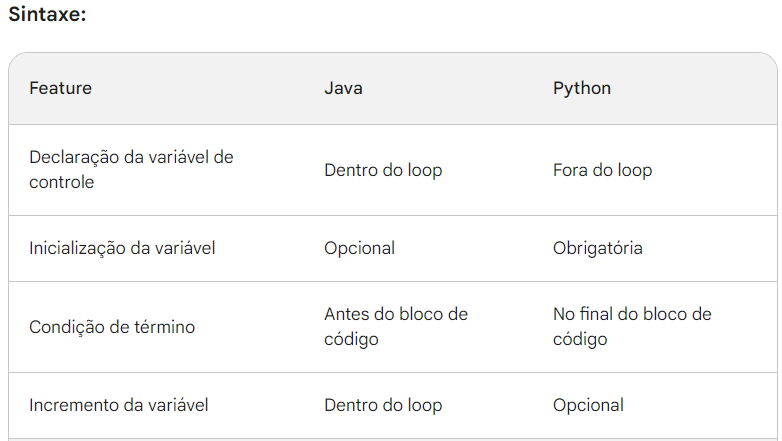
*}*

Neste exemplo, o loop irá iterar sobre os elementos da lista nomes, imprimindo cada nome na tela.

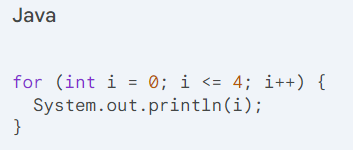
O loop for em Java é uma ferramenta poderosa e versátil que pode ser usada para automatizar diversas tarefas. Ao entender a sintaxe e as variantes do loop, você poderá escrever código mais eficiente e legível.

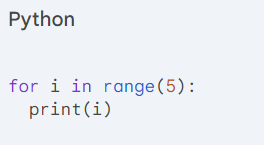
**3) Dê exemplo do FOR no java compare com o python.**

Tanto o Java quanto o Python possuem o loop for como uma estrutura de controle fundamental para iterar sobre sequências de elementos. Apesar de compartilharem a mesma funcionalidade básica, existem algumas diferenças importantes na sintaxe e na forma como são implementados em cada linguagem.

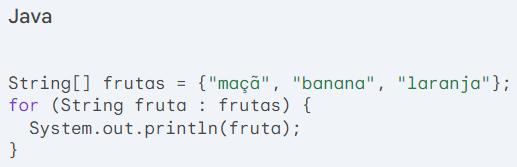


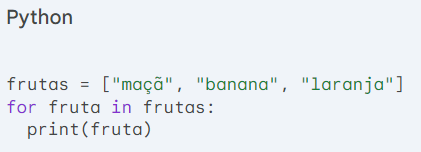
Exemplo 1: Imprimindo números de 0 a 4:





Exemplo 2: Iterando sobre elementos de uma lista:





Diferenças notáveis:

* Declaração da variável: Em Java, a variável de controle é declarada dentro do loop, enquanto em Python ela é declarada fora do loop. Isso pode afetar o escopo da variável após o término do loop.
* Inicialização: Em Java, a inicialização da variável é opcional, enquanto em Python é obrigatória. Se não for inicializada em Java, a variável pode conter um valor não definido.
* Condição de término: Em Java, a condição de término é verificada antes do bloco de código ser executado, enquanto em Python ela é verificada no final do bloco de código. Isso significa que em Java o bloco de código pode ser executado pelo menos uma vez, mesmo que a condição seja falsa no início.
* Incremento: Em Java, o incremento da variável é feito dentro do loop, enquanto em Python é opcional e pode ser especificado fora do loop.