Material: Python Básico de laço de repetição e Listas.

Professor Rafael Tasinato

O que é uma lista?
Uma lista é uma estrutura de dados que armazena uma coleção ordenada e mutável de valores. Em Python, uma lista é representada por colchetes [ ] que contêm os valores separados por vírgulas.
Por exemplo:

numeros = [1, 2, 3, 4, 5]

Essa é uma lista que contém cinco números inteiros. Uma lista pode conter valores de diferentes tipos, como números, strings, booleanos ou até outras listas.

Por exemplo:

mistura = [10, “Python”, True, [6, 7, 8]]

Essa é uma lista que contém quatro valores de tipos diferentes: um inteiro, uma string, um booleano e outra lista.

Uma lista é ordenada porque os valores têm uma posição fixa na sequência. Essa posição é chamada de índice e começa em zero. Podemos acessar um valor específico de uma lista usando o seu índice entre colchetes. Por exemplo:

numeros = [1, 2, 3, 4, 5]

print(numeros[0]) # imprime o primeiro valor da lista print(numeros[3])

imprime o quarto valor da lista print(numeros[-1])

imprime o último valor da lista

A saída será:

1 4 5

Uma lista é mutável porque podemos alterar os valores da lista usando atribuição ou métodos.

Por exemplo:

numeros = [1, 2, 3, 4, 5]

numeros[0] = 10 # altera o primeiro valor da lista para 10 numeros.append(6)

adiciona o valor 6 ao final da lista numeros.remove(4)

remove o valor 4 da lista print(numeros)

A saída será:

[10, 2, 3, 5, 6]

Essas são algumas características básicas de uma lista em Python. Uma lista é uma estrutura de dados muito útil e versátil que permite armazenar e manipular vários valores em um único objeto. Espero que essa explicação tenha sido útil para você.

Como Percorrer os valores em uma lista?
Para percorrer os valores de uma lista, você pode usar um laço de repetição for, que permite iterar sobre uma sequência de dados. A cada iteração, uma variável recebe o valor do elemento da lista e você pode executar um bloco de código com esse valor. Por exemplo:

```
numeros = [1, 2, 3, 4, 5]

for n in numeros:
    print(n * 2) # imprime o dobro de cada valor da lista
```

A saída será:

2 4 6 8 10

Esse é um exemplo simples de como percorrer os valores de uma lista usando um laço for em Python. Você pode usar esse laço para realizar diversas operações com os valores da lista, como somar, multiplicar, comparar.

Inserir os valores em uma lista.
Para inserir valores em uma lista, você pode usar o método append(), que adiciona um valor ao final da lista.
Por exemplo:

```
numeros = [1, 2, 3]

numeros.append(4) # adiciona o valor 4 ao final da lista print(numeros)
```

A saída será:

[1, 2, 3, 4]

Você também pode usar o método insert(), que adiciona um valor em uma posição específica da lista.

Por exemplo:

```
numeros = [1, 2, 3]

numeros.insert(1, 10) # adiciona o valor 10 na posição 1 da lista print(numeros)
```

A saída será:

[1, 10, 2, 3]

Esses são dois métodos que permitem inserir valores em uma lista em Python. Você pode usar esses métodos para adicionar novos valores à sua lista conforme necessário.

Remover valores em uma lista.
Para remover valores de uma lista, você pode usar o método remove(), que elimina o primeiro valor que corresponde ao argumento passado.

Por exemplo:

numeros = [1, 2, 3, 4, 5]

numeros.remove(3) # remove o valor 3 da lista print(numeros)

A saída será:

[1, 2, 4, 5]

Você também pode usar o método pop(), que elimina o valor na posição indicada pelo argumento passado e retorna esse valor. Se nenhum argumento for passado, o método pop() elimina e retorna o último valor da lista. Por exemplo:

numeros = [1, 2, 3, 4, 5]

n = numeros.pop(2) # remove e retorna o valor na posição 2 da lista print(n) print(numeros)

A saída será:

3 [1, 2, 4, 5]

Como limpar todos os valores de uma lista?

Esses são dois métodos que permitem remover valores de uma lista em Python. Você pode usar esses métodos para eliminar valores indesejados ou desnecessários da sua lista.
Mensagem recebida. Para remover valores

Para limpar todos os valores de uma lista, você pode usar o método clear(), que elimina todos os elementos da lista e deixa ela vazia. Por exemplo:

numeros = [1, 2, 3, 4, 5]

numeros.clear() # elimina todos os valores da lista print(numeros)

A saída será:

[]

Esse é um método que permite limpar todos os valores de uma lista em Python. Você pode usar esse método para esvaziar a sua lista quando não precisar mais dos valores armazenados nela.

Para Iterar uma string
Para iterar uma string, você pode usar um laço de repetição for, que permite percorrer cada caractere da string. A cada iteração, uma variável recebe o valor do caractere da string e você pode executar um bloco de código com esse valor. Por exemplo:

texto = “Python” for letra in texto: print(letra) # imprime cada letra da string

A saída será:

P

y

t

h

o

n

Esse é um exemplo simples de como iterar uma string usando um laço for em Python. Você pode usar esse laço para realizar diversas operações com os caracteres da string, como contar, substituir, comparar,

Substituir uma letra de uma string.

Para substituir uma letra ou palavra de um texto usando Python, você pode usar o método replace(), que recebe dois argumentos: o valor a ser substituído e o novo valor. O método replace() retorna uma nova string com as substituições feitas. Por exemplo:

texto = “Python é uma linguagem de programação”

novo_texto = texto.replace(“Python”, “Java”) # substitui “Python” por “Java” print(novo_texto)

A saída será:

Java é uma linguagem de programação

Você também pode usar o método replace() para substituir letras ou partes de palavras. Por exemplo:

texto = “Python é uma linguagem de programação”

novo_texto = texto.replace(“o”, “a”) # substitui todas as ocorrências de “o” por “a” print(novo_texto)

A saída será:

Pythan é uma linguagem de pragramaçãa

Esse é um método que permite substituir uma letra ou palavra de um texto usando Python. Você pode usar esse método para modificar o conteúdo de uma string conforme necessário.

Laço de Repetição

Um laço de repetição é uma estrutura de controle que permite executar um bloco de código várias vezes, enquanto uma condição for verdadeira. Em Python, existem dois tipos de laços de repetição: o for e o while. Vou explicar como eles funcionam e quando usá-los.

O laço for é usado para percorrer ou iterar sobre uma sequência de dados, como uma lista, uma tupla, uma string ou um dicionário. A cada iteração, uma variável recebe o valor do elemento da sequência. A sintaxe básica do laço for é:

for <variável> in <sequência>: # bloco de código que se repete

Por exemplo, se quisermos imprimir os números de 1 a 5, podemos usar um laço for com a função range(), que gera uma sequência de números:

for i in range(1, 6): print(i)

A saída será:

1 2 3 4 5

O laço while é usado quando não sabemos quantas vezes o bloco de código deve se repetir, mas sim quando ele deve parar. O laço while verifica uma condição lógica antes de cada iteração e só continua se ela for verdadeira. A sintaxe básica do laço while é:

while <condição>: # bloco de código que se repete

Por exemplo, se quisermos imprimir os números pares menores que 10, podemos usar um laço while com uma variável que vai sendo incrementada:

i = 0

while i < 10:

print(i) i += 2

A saída será:

0 2 4 6 8

Tanto o laço for quanto o laço while podem ter um bloco else associado, que é executado quando o laço termina normalmente (sem ser interrompido por um break). Por exemplo:

for i in range(1, 6):

print(i)

else:

print(“Fim do laço”)

A saída será:

1 2 3 4 5 Fim do laço

Exemplo 1: Imprimir os elementos de uma lista

Suponha que temos uma lista de frutas:

frutas = [“maçã”, “banana”, “laranja”, “manga”]

Se quisermos imprimir cada fruta da lista, podemos usar um laço for que percorre a lista e atribui cada elemento a uma variável chamada fruta. A cada iteração, o comando print(fruta) mostra o valor da variável fruta na tela. O código fica assim:

frutas = [“maçã”, “banana”, “laranja”, “manga”] for fruta in frutas: print(fruta)

A saída será:

maçã banana laranja manga

Exemplo 2: Calcular a soma dos números de 1 a 10

Se quisermos calcular a soma dos números de 1 a 10, podemos usar um laço for com a função range(), que gera uma sequência de números. A cada iteração, uma variável chamada i recebe o valor do número da sequência e é adicionada a uma variável chamada soma, que armazena o resultado parcial. No final do laço, o comando print(soma) mostra o valor final da variável soma na tela. O código fica assim:

soma = 0 for i in range(1, 11): soma = soma + i print(soma)

A saída será:

55

Exemplo 3: Imprimir uma tabuada

Se quisermos imprimir a tabuada de um número n, podemos usar um laço for com a função range(), que gera uma sequência de números de 1 a 10. A cada iteração, uma variável chamada i recebe o valor do número da sequência e é multiplicada pelo valor de n. O comando print(f"{n} x {i} = {n*i}") mostra o resultado da multiplicação na tela, usando uma f-string para formatar a saída. O código fica assim:

n = int(input(“Digite um número: “)) for i in range(1, 11): print(f”{n} x {i} = {n*i}”)

A saída será (supondo que o usuário digitou 5):

5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25 5 x 6 = 30 5 x 7 = 35 5 x 8 = 40 5 x 9 = 45 5 x 10 = 50

Exercícios:

1. Escreva um programa que leia uma string do usuário e imprima cada caractere da string em uma linha separada.
2. Escreva um programa que leia um número inteiro positivo n do usuário e imprima os n primeiros números naturais (de 1 a n) usando um laço for.
3. Escreva um programa que leia uma lista de números do usuário e imprima a soma de todos os números da lista usando um laço for.
4. Escreva um programa que leia uma lista de palavras do usuário e imprima a palavra mais longa da lista usando um laço for.
5. Escreva um programa que leia uma string do usuário e imprima o número de vogais (a, e, i, o, u) que aparecem na string usando um laço for.
6. Escreva um programa que leia um número inteiro positivo n do usuário e imprima a tabuada de n (de 1 a 10) usando um laço for.
7. Escreva um programa que leia uma string do usuário e imprima a string ao contrário usando um laço while.
8. Escreva um programa que leia um número inteiro positivo n do usuário e imprima se n é primo ou não usando um laço while.
9. Escreva um programa que leia uma lista de números do usuário e imprima o maior e o menor número da lista usando um laço while.
10. Escreva um programa que leia duas listas de números do usuário e imprima a lista resultante da concatenação das duas listas usando um laço for.

Referências:

http://devfuria.com.br/python/lacos-de-repeticao/

https://www.freecodecamp.org/portuguese/news/laco-for-em-python-exemplo-de-for-i-in-range/

https://vaiprogramar.com/como-usar-os-lacos-de-repeticao-for-e-while-em-python/

https://www.respondeai.com.br/conteudo/programacao/python/lacos-de-repeticao/2042

https://bing.com/search?q=Python+listas+strings+la%c3%a7os+de+repeti%c3%a7%c3%a3o