

CAPÍTULO 8

Laços

Em linguagens de programação, um *laço* é uma estrutura que permite a repetição de um bloco de código várias vezes. *Laços* são ferramentas poderosas na programação, pois permitem automatizar tarefas repetitivas e processar grandes quantidades de dados de forma eficiente.

As duas formas principais de criar *laços* são o *for* e o *while*.

8.1 Laço for

O **for** é usado quando se sabe previamente o número de iterações que devem ser executadas. Por exemplo, o número de elementos numa estrutura de dados é o número de . O laço percorre uma sequência de elementos (string, lista, uma tupla).

Um exemplo:

```
1 >>> genes = ["APOE", "TOMM40", "FTO"]
2 >>> for gene in genes:
3 >>>     print(gene)
```

Nesse exemplo, o laço **for** percorre a lista de genes e imprime cada uma delas. A variável *gene* assume o valor de cada elemento da lista a cada iteração.

Quando usar o laço **for**:

- Quando você tem uma sequência fixa de elementos para percorrer.
- Quando você sabe antecipadamente quantas iterações são necessárias.
- Quando você precisa realizar uma operação para cada elemento em uma sequência.

8.2 Laço while

O **while** é usado quando não se sabe quantas vezes o bloco de código precisa ser executado. A repetição é continuada enquanto uma determinada condição for verdadeira.

Veja o exemplo seguinte. O laço **while** imprime o valor do contador enquanto o contador for menor que 3. A cada iteração, a variável índice é incrementado em 1.

```
1 >>> indice = 0
2 >>> genes = ["APOE", "TOMM40", "FTO"]
3 >>> while indice < len(genes):
4 >>>     print(genes[indice])
5 >>>     indice += 1
```

Evite laços infinitos, que nunca terminam porque a condição nunca se torna falsa.

Na estrutura dos laços, você pode incluir outras estruturas condicionais, como **if**, para controlar o fluxo do programa com base em certas condições. Além disso, é possível utilizar as palavras-chave **break** e **continue** para interromper o laço ou avançar para a próxima iteração, respectivamente.

O **break** e o **continue** são palavras-chave usadas em laços **for** para controlar o fluxo de execução do laço.

O **break** **interrompe** a execução do laço *for* de forma **abrupta**. Quando o **break** é encontrado dentro de um laço **for**, o laço é encerrado imediatamente, independentemente de quantas iterações ainda restem.

No exemplo abaixo, o laço **for** percorre os números de 1 a 9. Quando **i** é igual a 5, o **break** é acionado e o laço é interrompido

```
1 >>> for i in range(1, 10):
2 >>>     if i == 5:
3 >>>         break
4 >>>     print(i)
```

O **continue** é usado para saltar o restante do código dentro de uma iteração específica do laço **for** ou **while** e avançar para a próxima iteração. Quando o **continue** é encontrado dentro de um laço, o código seguinte do **continue** não é executado e o laço continua com a próxima iteração.

```
1 >>> for i in range(1, 6):  
2 >>>     if i == 3:  
3 >>>         continue  
4 >>>     print(i)
```

Quando usar o laço while:

Quando você precisa repetir um bloco de código até que uma determinada condição seja atendida. Quando você não sabe quantas iterações serão necessárias para atingir um objetivo. Quando a condição de parada pode ser baseada em entradas do usuário ou em variáveis que podem mudar durante a execução do programa.

Em resumo, use o laço **for** quando você souber a quantidade fixa de iterações e quiser percorrer uma sequência, e use o laço **while** quando você precisar repetir um bloco de código até que uma condição seja satisfeita, e essa condição possa variar durante a execução do programa.

Em resumo, o **break** é usado para sair completamente de um laço **for** ou **while**, enquanto o **continue** é usado para pular o restante do código dentro de uma iteração específica e avançar para a próxima iteração.