



Iteráveis

Iterável

▼ Definição

? Um **iterável** é qualquer objeto que pode ser percorrido em um loop, ou seja, um objeto que contém uma sequência de elementos acessíveis um de cada vez. Para ser considerado iterável, o objeto deve implementar o método especial `__iter__()`, que retorna um iterador. Exemplos comuns de iteráveis incluem: strings, o tipo `range`, listas, tuplas e dicionários. Isso permite a utilização desses objetos em loops, como o `for`, para acessar seus elementos sequencialmente.

▼ Strings (`str`)

- Strings são iteráveis, pois você pode percorrer cada caractere.

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

▼ Range (`range(i, f, p)`)



É um tipo de dado utilizado para gerar sequências de números de forma eficiente.

Ele é comumente usado em loops, como o

`for`, para iterar sobre uma sequência de números. O `range` é um **objeto iterável**, mas, ao contrário de uma lista, ele não armazena os números na memória de forma explícita. Em vez disso, ele gera os números conforme necessário, o que torna o `range` muito eficiente para grandes sequências.

- Possui um parâmetro para início, um para fim (não incluindo esse valor) e um para passo (intervalo entre os números).
- Sintaxe: `range (i , f , p)`
- Exemplo simples:

```
for i in range(3):  
    print(i) # Exibe os números 0, 1, 2
```

- Exemplo com início, fim e passo:

```
for i in range(2, 11, 2):  
    print(i) # Exibe os números 2, 4, 6, 8, 10
```

▼ Pegar índices de iteráveis (`enumerate()`)

- `enumerate (variavel)` retorna um iterador, podendo ser utilizado em loops de repetição. Exemplo:

```
frutas = ["maçã", "banana", "laranja"]  
for indice, fruta in enumerate(frutas):  
    print(f"{indice}: {fruta}") # Exibe 0: maçã 1: banana 2: laranja
```

- Pode ser usado para enumerar valores de um iterável. Exemplo:

```
frutas = ["maçã", "banana", "laranja"]  
frutas_enumeradas = list(enumerate(frutas))
```

```
print(frutas_enumeradas) # [(0, 'Maria'), (1, 'Paulo'), (2, 'Heitor')]
```

- O argumento `start` no `enumerate ()` permite especificar o índice inicial ao iterar sobre os elementos de um iterável. No caso a seguir, a lista foi enumerada de 1 a 3, ao invés de 0 a 2.

```
frutas = ["maçã", "banana", "laranja"]  
for indice, fruta in enumerate(frutas, start = 1):  
    print(f"{indice}: {fruta}") # Exibe 1: maçã 2: banana 3: laranja
```