

Usar loops 'for' com listas

5 minutos

No Python, as listas podem armazenar qualquer tipo de valor, como cadeias de caracteres ou números:

Python

```
planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus",  
"Neptune"]
```

Você pode acessar qualquer item em uma lista ao colocar o *índice* entre colchetes (`[]`) após o nome da variável. Os índices começam com 0:

Python

```
planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus",  
"Neptune"]  
  
print("The first planet is ", planets[0])  
print("The second planet is ", planets[1])  
print("The third planet is ", planets[2])
```

Você também pode determinar o número de itens em uma lista usando `len`. Portanto, você pode usar um loop `while` e um contador para executar um loop ou iterar em cada item da lista. Como essa é uma operação comum, o Python fornece loops `for`, que podem ser usados para a iteração em listas.

ⓘ Observação

O Python tem muitos tipos que podem ser percorridos com os loops. Esses tipos são conhecidos como *objetos iteráveis*.

As listas do Python são iteráveis e podem ser usadas com o loop `for`. Você usa um loop `for` com objetos iteráveis, em que você executa um loop por um número conhecido de vezes, uma vez para cada item do objeto iterável.

Sobre os loops `for`

Veja o seguinte exemplo de um loop `for` que conta regressivamente de 4 a 0:

Python

```
countdown = [4, 3, 2, 1, 0]
for number in countdown:
    print(number)
print("Blast off!! 🚀")
```

O loop `for` é uma instrução com cinco partes importantes:

- A palavra `for`, seguida por um espaço.
- O nome da variável que você deseja criar para cada valor da sequência (`number`).
- A palavra `in`, cercada por espaços.
- O nome da lista (`countdown`, no exemplo anterior) ou do objeto iterável que você deseja percorrer com o loop, seguido de dois-pontos (`:`).
- O código que você deseja executar para cada item do objeto iterável, separado por espaço em branco aninhado.

Vamos alterar esse código para aguardar um segundo entre cada número usando a função `sleep()`:

Python

```
from time import sleep

countdown = [4, 3, 2, 1, 0]

for number in countdown:
    print(number)
    sleep(1) # Wait 1 second
print("Blast off!! 🚀")
```

📌 Observação

A maioria dos códigos Python usa quatro espaços como a unidade de espaço em branco. Para evitar ter que pressionar a barra de espaço quatro vezes, a maioria dos editores tem um atalho de tecla `Tab` que insere quatro espaços.

Unidade seguinte: Exercício – Criar um loop 'for'

Continuar >
