

Apresentação das listas

5 minutos

O Python tem muitos tipos internos, como cadeias de caracteres e inteiros. O Python tem um tipo para armazenar coleções de valores: a lista.

Cria uma lista

Você cria uma lista atribuindo uma sequência de valores a uma variável. Cada valor é separado por vírgula e cercado por colchetes (`[]`). O seguinte exemplo armazena a lista de todos os planetas na variável `planets`:

Python

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

Acessar itens de lista por índice

Você pode acessar qualquer item em uma lista colocando o *índice* entre colchetes `[]` após o nome da variável de lista. Os índices começam com 0, portanto, no seguinte código, `planets[0]` é o primeiro item na lista `planets`:

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])
```

Output

```
The first planet is Mercury
The second planet is Venus
The third planet is Earth
```

ⓘ Observação

Como todos os índices começam de 0, [1] é o segundo item, [2] é o terceiro e assim por diante.

Você também pode modificar os valores de uma lista usando um índice. Você faz isso atribuindo um novo valor, praticamente da mesma forma que faria com uma variável. Por exemplo, você pode alterar o nome de Marte na lista para usar o apelido desse planeta:

Python

```
planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune"]
planets[3] = "Red Planet"
print("Mars is also known as", planets[3])
```

Saída: Mars is also known as Red Planet

Determinar o comprimento de uma lista

Para obter o comprimento de uma lista, use a função interna `len()`. O código a seguir cria uma variável, `number_of_planets`. O código atribui essa variável ao número de itens da lista `planets` (8).

Python

```
planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune"]
number_of_planets = len(planets)
print("There are", number_of_planets, "planets in the solar system.")
```

Saída: There are 8 planets in the solar system

Adicionar valores às listas

As listas em Python são dinâmicas: você pode adicionar e remover itens depois que eles são criadas. Para adicionar um item a uma lista, use o método `.append(value)`.

Por exemplo, o seguinte código adiciona a cadeia de caracteres `"Pluto"` ao final da lista `planets`:

Python

```
planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus",  
"Neptune"]  
planets.append("Pluto")  
number_of_planets = len(planets)  
print("There are actually", number_of_planets, "planets in the solar system.")
```

Saída: There are actually 9 planets in the solar system.

Remover valores das listas

Você pode remover o último item de uma lista chamando o método `.pop()` na variável de lista:

Python

```
planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus",  
"Neptune", "Pluto"]  
planets.pop() # Goodbye, Pluto  
number_of_planets = len(planets)  
print("No, there are definitely", number_of_planets, "planets in the solar  
system.")
```

Usar índices negativos

Você viu como usar índices para buscar um item individual em uma lista:

Python

```
print("The first planet is", planets[0])
```

Saída: The first planet is Mercury

Os índices começam do zero em diante. Os índices negativos começam ao final da lista e funcionam de maneira retroativa.

No exemplo a seguir, o índice `-1` retorna o último item da lista. O índice `-2` retorna o penúltimo item da lista.

Python

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

```
print("The last planet is", planets[-1])  
print("The penultimate planet is", planets[-2])
```

Output

```
The last planet is Neptune  
The penultimate planet is Uranus
```

Para retornar o antepenúltimo item da lista, basta usar o índice `-3` (e assim por diante).

Localizar um valor em uma lista

Para determinar em que local da lista um valor está armazenado, use o método `index` sobre a lista. Esse método pesquisa o valor e retorna o índice desse item na lista. Se não encontrar uma correspondência, ele retornará `-1`.

O seguinte exemplo mostra o uso de "Jupiter" como o valor de índice:

Python

```
planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus",  
"Neptune"]  
jupiter_index = planets.index("Jupiter")  
print("Jupiter is the", jupiter_index + 1, "planet from the sun")
```

Saída: Jupiter is the 5 planet from the sun

⚠ Observação

Como a indexação começa no 0, você precisa adicionar 1 para exibir o número correto.

Veja outro exemplo:

Python

```
planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus",  
"Neptune"]  
mercury_index = planets.index("Mercury")  
print("Mercury is the", mercury_index + 1, "planet from the sun")
```

Saída: Mercury is the 1 planet from the sun

Unidade seguinte: Exercício – Criar e usar listas no Python

Continuar >
