

Manipular dados de lista

5 minutos

Você pode precisar trabalhar com diferentes partes de uma lista. Por exemplo, suponha que você tem uma lista com os volumes de chuva de vários meses. Para analisar corretamente esse tipo de dado, talvez seja necessário examinar a precipitação pluvial em um período de três meses. Ou talvez você precise classificar a lista em ordem decrescente de volume de chuva.

O Python fornece suporte robusto para trabalhar com os dados em listas. Esse suporte inclui fatiar (examinar apenas parte) e classificar os dados da lista.

Fatiar listas

Você pode extrair parte de uma lista usando uma *fatia*. A fatia usa colchetes, mas em vez de conter apenas um item ela tem os índices inicial e final da fatia. Ao usar uma fatia, você cria uma lista que começa no índice inicial e termina imediatamente antes do índice final (*não* inclui ele).

A lista de planetas tem oito itens. A terra é a terceira da lista. Para obter os planetas antes da terra, use uma fatia para obter os itens começando em 0 e terminando em 2:

Python

```
planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus",  
"Neptune"]  
planets_before_earth = planets[0:2]  
print(planets_before_earth)
```

Saída: ['Mercury', 'Venus']

Observe que a terra não está incluída na lista. O motivo é que o índice termina antes do índice final.

Para obter todos os planetas após a terra, comece no terceiro e vá até o oitavo:

Python

```
planets_after_earth = planets[3:8]  
print(planets_after_earth)
```

Saída: ['Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune']

Neste exemplo, Netuno é exibido. O motivo é que o índice de Netuno é 7, já que a indexação começa em 0. Como o índice final era 8, ele inclui o último valor. Se você não informar o índice de término de uma fatia, o Python pressupõe que você deseja que ela abranja até o final da lista:

Python

```
planets_after_earth = planets[3:]  
print(planets_after_earth)
```

Saída: ['Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune']

Importante

Cada fatia representa uma *nova* lista. Ela não modifica a lista atual.

Unir listas

Você viu que é possível usar fatias para dividir as listas, mas será que podemos uni-las de volta?

Para unir duas listas, use o operador (+) com duas listas para retornar uma terceira.

Há 79 luas conhecidas em Júpiter. As quatro maiores são Io, Europa, Ganymede e Callisto. Essas luas são chamadas de galileanas porque Galileu Galilei as descobriu usando um telescópio em 1610. Mais próximas de Júpiter do que o grupo de Galileo está o grupo de Amalteia. Ele consiste nas luas Metis, Adrastea, Amalteia e Tebe.

Crie duas listas. Preencha a primeira com as quatro luas de Amalteia e a segunda lista com as quatro luas de Galileo. Junte-as usando + para criar uma terceira lista:

Python

```
amalthea_group = ["Metis", "Adrastea", "Amalthea", "Thebe"]  
galilean_moons = ["Io", "Europa", "Ganymede", "Callisto"]  
  
regular_satellite_moons = amalthea_group + galilean_moons  
print("The regular satellite moons of Jupiter are", regular_satellite_moons)
```

Saída: The regular satellite moons of Jupiter are ['Metis', 'Adrastea', 'Amalthea', 'Thebe', 'Io', 'Europa', 'Ganymede', 'Callisto']

📘 Importante

A junção de duas listas gera uma *nova* lista. Ela não modifica a lista atual.

Classificar listas

Para classificar uma lista, use o método `.sort()` sobre a lista. O Python classifica uma lista de cadeias de caracteres em ordem alfabética e uma lista de números em ordem numérica:

Python

```
amalthea_group = ["Metis", "Adrastea", "Amalthea", "Thebe"]
galilean_moons = ["Io", "Europa", "Ganymede", "Callisto"]

regular_satellite_moons = amalthea_group + galilean_moons
regular_satellite_moons.sort()
print("The regular satellite moons of Jupiter are", regular_satellite_moons)
```

Saída: The regular satellite moons of Jupiter are ['Adrastea', 'Amalthea', 'Callisto', 'Europa', 'Ganymede', 'Io', 'Metis', 'Thebe']

Para classificar uma lista em ordem inversa, chame `.sort(reverse=True)` sobre a lista:

Python

```
amalthea_group = ["Metis", "Adrastea", "Amalthea", "Thebe"]
galilean_moons = ["Io", "Europa", "Ganymede", "Callisto"]

regular_satellite_moons = amalthea_group + galilean_moons
regular_satellite_moons.sort(reverse=True)
print("The regular satellite moons of Jupiter are", regular_satellite_moons)
```

Saída: The regular satellite moons of Jupiter are ['Thebe', 'Metis', 'Io', 'Ganymede', 'Europa', 'Callisto', 'Amalthea', 'Adrastea']

📘 Importante

O uso de `sort` modifica a lista atual.

Unidade seguinte: Exercício – Trabalhar com os dados da lista

Continuar >
