

Programação dinâmica com dicionários

3 minutos

Em seu programa, você deseja executar vários cálculos, como totalizar o número de luas. Além disso, à medida que você entrar em uma programação mais avançada, poderá achar que está carregando esse tipo de informação de arquivos ou de um banco de dados, em vez de codificar diretamente no Python.

Para ajudar a dar suporte a esses cenários, o Python permite que você trate as chaves e os valores dentro de um dicionário como uma lista. Você pode determinar as chaves e os valores dinamicamente e executar vários cálculos.

Imagine um dicionário que armazena os volumes de chuva mensais. Provavelmente, você teria chaves para cada mês e o volume de chuva associado. Você deseja somar o volume total de chuva e escrever código para executar a operação usando cada chave individual seria um tanto quanto trabalhoso.

Recuperar todas as chaves e valores

O método `keys()` retorna um objeto de lista que contém todas as chaves. Você pode usar esse método para percorrer todos os itens no dicionário.

Imagine que você tem o dicionário a seguir, armazenando os últimos três meses de chuva.

Python

```
rainfall = {  
    'october': 3.5,  
    'november': 4.2,  
    'december': 2.1  
}
```

Digamos que você deseja exibir a lista com todos os volumes de chuva registrados. Você poderia digitar o nome de cada mês, mas isso seria entediante.

Python

```
for key in rainfall.keys():  
    print(f'{key}: {rainfall[key]}cm')
```

Output

```
october: 3.5cm  
november: 4.2cm  
december: 2.1cm
```

❗ Observação

Você ainda pode usar colchetes (`[]`) com um nome de variável, em vez do literal de cadeia de caracteres embutido em código.

Determinar se uma chave existe em um dicionário

Quando você atualiza um valor em um dicionário, o Python substitui o valor existente ou cria outro valor, se a chave não existir. Se você quiser adicionar um valor em vez de substituí-lo, poderá verificar se a chave existe usando `in`. Por exemplo, se quiser adicionar um valor a Dezembro ou criar um do zero, caso não exista, você poderá usar o seguinte:

Python

```
if 'december' in rainfall:  
    rainfall['december'] = rainfall['december'] + 1  
else:  
    rainfall['december'] = 1  
  
# Because december exists, the value will be 3.1
```

Recuperar todos os valores

Semelhante a `keys()`, `values()` retorna a lista com todos os valores em um dicionário *sem* as respectivas chaves. `values()` pode ser útil quando você usa a chave para fins de rotulagem, como no exemplo anterior, em que as chaves são o nome do mês. Você pode usar `values()` para determinar o volume total de chuva:

Python

```
total_rainfall = 0  
for value in rainfall.values():  
    total_rainfall = total_rainfall + value  
  
print(f'There was {total_rainfall}cm in the last quarter.')
```

Output

There was 10.8cm in the last quarter.

Unidade seguinte: Exercício – Programação dinâmica com dicionários

[Continuar >](#)