

Python - Remover itens do dicionário

Usando del palavra-chave

A **palavra-chave del** Python exclui qualquer objeto da memória. Aqui o usamos para excluir um par de valores-chave em um dicionário.

Sintaxe

```
del dict['key']
```

Exemplo

```
numbers = {10:"Ten", 20:"Twenty", 30:"Thirty",40:"Forty"}  
print ("numbers dictionary before delete operation: \n", numbers)  
del numbers[20]  
print ("numbers dictionary before delete operation: \n", numbers)
```

Ele produzirá a seguinte **saída** -

```
numbers dictionary before delete operation:  
{10: 'Ten', 20: 'Twenty', 30: 'Thirty', 40: 'Forty'}  
numbers dictionary before delete operation:  
{10: 'Ten', 30: 'Thirty', 40: 'Forty'}
```

Exemplo

A palavra-chave del com o próprio objeto dict o remove da memória.

```
numbers = {10:"Ten", 20:"Twenty", 30:"Thirty",40:"Forty"}  
print ("numbers dictionary before delete operation: \n", numbers)  
del numbers  
print ("numbers dictionary before delete operation: \n", numbers)
```

Ele produzirá a seguinte **saída** -

```
numbers dictionary before delete operation:  
{10: 'Ten', 20: 'Twenty', 30: 'Thirty', 40: 'Forty'}  
Traceback (most recent call last):
```



```
File "C:\Users\mlath\examples\main.py", line 5, in <module>
    print ("numbers dictionary before delete operation: \n", numbers)
                                         ^^^^^^^^
```

NameError: name 'numbers' is not defined

Usando o método pop()

O método pop() da classe dict faz com que um elemento com a chave especificada seja removido do dicionário.

Sintaxe

```
val = dict.pop(key)
```

Valor de retorno

O método pop() retorna o valor da chave especificada após remover o par chave-valor.

Exemplo

```
numbers = {10:"Ten", 20:"Twenty", 30:"Thirty",40:"Forty"}
print ("numbers dictionary before pop operation: \n", numbers)
val = numbers.pop(20)
print ("nubvers dictionary after pop operation: \n", numbers)
print ("Value popped: ", val)
```

Ele produzirá a seguinte **saída** -

```
numbers dictionary before pop operation:
{10: 'Ten', 20: 'Twenty', 30: 'Thirty', 40: 'Forty'}
nubvers dictionary after pop operation:
{10: 'Ten', 30: 'Thirty', 40: 'Forty'}
Value popped: Twenty
```

Usando o método popitem()

O método popitem() na classe dict() não aceita nenhum argumento. Ele exibe o último par de valores-chave inserido e retorna o mesmo que uma tupla

Sintaxe

```
val = dict.popitem()
```

Valor de retorno

O método `popitem()` retorna uma tupla contendo a chave e o valor do item removido do dicionário

Exemplo

```
numbers = {10:"Ten", 20:"Twenty", 30:"Thirty",40:"Forty"}
print ("numbers dictionary before pop operation: \n", numbers)
val = numbers.popitem()
print ("numbers dictionary after pop operation: \n", numbers)
print ("Value popped: ", val)
```

Ele produzirá a seguinte **saída** -

```
numbers dictionary before pop operation:
{10: 'Ten', 20: 'Twenty', 30: 'Thirty', 40: 'Forty'}
numbers dictionary after pop operation:
{10: 'Ten', 20: 'Twenty', 30: 'Thirty'}
Value popped: (40, 'Forty')
```

Usando o método `clear()`

O método `clear()` na classe `dict` remove todos os elementos do objeto de dicionário e retorna um objeto vazio.

Sintaxe

```
dict.clear()
```

Exemplo

```
numbers = {10:"Ten", 20:"Twenty", 30:"Thirty",40:"Forty"}
print ("numbers dictionary before clear method: \n", numbers)
numbers.clear()
print ("numbers dictionary after clear method: \n", numbers)
```

Ele produzirá a seguinte **saída** -

numbers dictionary before clear method:

```
{10: 'Ten', 20: 'Twenty', 30: 'Thirty', 40: 'Forty'}
```

numbers dictionary after clear method:

```
{}
```