

Noções básicas de cadeia de caracteres no Python

3 minutos

Embora as cadeias de caracteres no Python pareçam simples e diretas, há certa complexidade nas regras das cadeias de caracteres que é importante entender. Conhecer as regras ajuda você a evitar surpresas com o comportamento das cadeias de caracteres ao modificar valores ou formatar textos.

Uma cadeia de caracteres simples

No exemplo desta unidade, há um só fato sobre a lua atribuído a uma variável, desta forma:

Python

```
fact = "The Moon has no atmosphere."  
print(fact)
```

A saída mostra que o texto foi atribuído à variável: `The Moon has no atmosphere.`

Imutabilidade das cadeias de caracteres

No Python, as cadeias de caracteres são imutáveis. Ou seja, eles não podem mudar. Essa propriedade do tipo cadeia de caracteres pode ser bastante surpreendente, pois o Python não acusa erro quando você altera uma cadeia de caracteres.

Você precisa adicionar outro fato (frase) ao fato único atribuído a uma variável. *Parece que* adicionar o segundo fato alteraria a variável, como no seguinte exemplo:

Python

```
fact = "The Moon has no atmosphere."  
fact + "No sound can be heard on the Moon."
```

A saída deverá ser: `The Moon has no atmosphere.No sound can be heard on the Moon.`

Embora possa parecer que modificamos a variável `fact`, uma verificação rápida do valor revela que o valor original está intacto:

Python

```
fact = "The Moon has no atmosphere."  
fact + "No sound can be heard on the Moon."  
print(fact)
```

Saída: `The Moon has no atmosphere.`

O truque aqui é que você precisa usar um valor retornado. Quando você adiciona cadeias de caracteres, o Python não modifica nenhuma cadeia de caracteres, mas retorna uma *nova cadeia* de caracteres como resultado. Para manter esse novo resultado, atribua-o a uma nova variável:

Python

```
fact = "The Moon has no atmosphere."  
two_facts = fact + "No sound can be heard on the Moon."  
print(two_facts)
```

Saída: `The Moon has no atmosphere.No sound can be heard on the Moon.`

As operações em cadeias de caracteres sempre produzem novas cadeias de caracteres como resultado.

Sobre como usar aspas

Você pode colocar as cadeias de caracteres do Python entre aspas simples, duplas ou triplas. Embora você possa usá-las de maneira intercambiável, é melhor adotar o mesmo tipo de modo consistente em todo o projeto. Por exemplo, a seguinte cadeia de caracteres usa aspas duplas:

Python

```
moon_radius = "The Moon has a radius of 1,080 miles."
```

No entanto, quando uma cadeia de caracteres contém palavras, números ou caracteres especiais (uma *substring*) que também estão entre aspas, você deve usar um estilo diferente. Por exemplo, se uma substring usar aspas duplas, coloque toda a cadeia de caracteres entre aspas simples, conforme mostrado aqui:

Python

```
'The "near side" is the part of the Moon that faces the Earth.'
```

Da mesma forma, se houver aspas simples (ou um apóstrofo, como na palavra *Lua's* no exemplo a seguir) em qualquer lugar dentro da cadeia de caracteres, coloque toda a cadeia de caracteres entre aspas duplas:

Python

```
"We only see about 60% of the Moon's surface."
```

Se você não alternar entre aspas simples e duplas o interpretador do Python pode gerar um erro de sintaxe, conforme mostrado aqui:

Python

```
'We only see about 60% of the Moon's surface.'
File "<stdin>", line 1
    'We only see about 60% of the Moon's surface.'
                                   ^
SyntaxError: invalid syntax
```

Quando o texto tiver uma combinação de aspas simples e duplas, você poderá usar aspas triplas para evitar problemas com o interpretador:

Python

```
"""We only see about 60% of the Moon's surface, this is known as the "near side."""
```

Texto com várias linhas

Há algumas maneiras diferentes de definir várias linhas de texto como uma variável. As maneiras mais comuns são:

- Usar um caractere de nova linha (`\n`).
- Usar aspas triplas (`"""`).

Os caracteres de nova linha separam o texto em várias linhas quando você imprime a saída:

Python

```
multiline = "Facts about the Moon:\n There is no atmosphere.\n There is no sound."
print(multiline)
```

Output

```
Facts about the Moon:
There is no atmosphere.
There is no sound.
```

Você pode obter o mesmo resultado usando aspas triplas:

Python

```
multiline = """Facts about the Moon:  
There is no atmosphere.  
There is no sound."""  
print(multiline)
```

Unidade seguinte: Métodos de cadeia de caracteres no Python

[Continuar >](#)