

Formato de cadeia de caracteres no Python

3 minutos

Além de transformar texto e executar operações básicas, como correspondência e pesquisa, é essencial formatar o texto quando você estiver apresentando informações. A maneira mais simples de apresentar informações de texto com o Python é usar a função `print()`. Você verá que é essencial colocar informações em variáveis e outras estruturas de dados em cadeias de caracteres que podem ser usadas por `print()`.

Nesta unidade, você aprenderá várias maneiras válidas de incluir valores de variáveis em texto usando o Python.

Formatação do sinal de porcentagem (%)

O espaço reservado para a variável na cadeia de caracteres é `%s`. Após a cadeia de caracteres, use outro caractere `%` seguido pelo nome da variável. O exemplo a seguir mostra como formatar usando o caractere `%`:

Python

```
mass_percentage = "1/6"
print("On the Moon, you would weigh about %s of your weight on Earth." % mass_percentage)
```

Saída: On the Moon, you would weigh about 1/6 of your weight on Earth.

O uso de vários valores altera a sintaxe, pois requer parênteses ao redor das variáveis que são passadas:

Python

```
print("""Both sides of the %s get the same amount of sunlight, but only one side is seen from %s because the %s rotates around its own axis when it orbits %s.""" % ("Moon", "Earth", "Moon", "Earth"))
```

Saída: Both sides of the Moon get the same amount of sunlight, but only one side is seen from Earth because the Moon rotates around its own axis when it orbits Earth.

💡 Dica

Embora esse método ainda seja uma forma válida de formatar cadeias de caracteres, ele pode levar a erros e reduzir a clareza do código quando você estiver lidando com diversas variáveis. Qualquer uma das outras duas opções de formatação descritas nesta unidade seria mais adequada para essa finalidade.

O método `format()`

O método `.format()` usa chaves (`{}`) como espaços reservados em uma cadeia de caracteres e usa a atribuição de variável para substituir o texto.

Python

```
mass_percentage = "1/6"
print("On the Moon, you would weigh about {} of your weight on Earth.".format(mass_percentage))
```

Saída: On the Moon, you would weigh about 1/6 of your weight on Earth.

Você não precisa atribuir variáveis repetidas diversas vezes, tornando-as menos detalhadas porque menos variáveis precisam ser atribuídas:

Python

```
mass_percentage = "1/6"
print("""You are lighter on the {0}, because on the {0} you would weigh about {1} of your weight on
Earth.""".format("Moon", mass_percentage))
```

Saída: You are lighter on the Moon, because on the Moon you would weigh about 1/6 of your weight on Earth.

Em vez de chaves vazias, a substituição deve usar números. O `{0}` significa usar o primeiro argumento (índice zero) para `.format()`, que, nesse caso, é `Moon`. Para repetição simples `{0}` funciona bem, mas reduz a legibilidade. Para melhorar a legibilidade, use argumentos de palavra-chave em `.format()` e faça referência aos mesmos argumentos entre chaves:

Python

```
mass_percentage = "1/6"
print("""You are lighter on the {moon}, because on the {moon} you would weigh about {mass} of your weight on
Earth.""".format(moon="Moon", mass=mass_percentage))
```

Saída: You are lighter on the Moon, because on the Moon you would weigh about 1/6 of your weight on Earth.

Sobre cadeias de caracteres f

Desde a versão 3.6 do Python, é possível usar as *cadeias de caracteres f*. Essas cadeias de caracteres se parecem com modelos e usam os nomes de variáveis do seu código. O uso de cadeias de caracteres f no exemplo anterior ficaria assim:

Python

```
print(f"On the Moon, you would weigh about {mass_percentage} of your weight on Earth.")
```

Saída: On the Moon, you would weigh about 1/6 of your weight on Earth.

As variáveis são passadas entre chaves e a cadeia de caracteres *deve* usar o prefixo `f`.

Além das cadeias de caracteres f serem menos detalhadas do que qualquer outra opção de formatação, é possível usar expressões dentro das chaves. Essas expressões podem ser funções ou operações diretas. Por exemplo, se você quiser representar o valor `1/6` como um percentual com uma casa decimal, poderá usar a função `round()` diretamente:

Python

```
print(round(100/6, 1))
```

Saída: 16.7

Com cadeias de caracteres f, você não precisa atribuir um valor a uma variável com antecedência:

Python

```
print(f"On the Moon, you would weigh about {round(100/6, 1)}% of your weight on Earth.")
```

Saída: On the Moon, you would weigh about 16.7% of your weight on Earth.

Usar uma expressão não requer uma chamada de função. Qualquer um dos métodos de cadeia de caracteres também é válido. Por exemplo, a cadeia de caracteres pode impor um uso específico de maiúsculas e minúsculas à criação de um título:

Python

```
subject = "interesting facts about the moon"
heading = f"{subject.title()}"
print(heading)
```

Saída: Interesting Facts About The Moon

Unidade seguinte: Exercício – Formatar cadeias de caracteres

[Continuar >](#)
