

Cursos

Productos

Rutas profesionales

Rutas de aprendizaje

Cursos

Centro para formadores

Centro de estudiantes

Preguntas más frecuentes y ayuda

NIVEL 3

2400 /3699 XP

Learn / Cursos / Examinar / Python para principiantes / Uso de cadenas en Python /

< Anterior

Unidad 5 de 8

Siguientes >

Formato de cadenas en Python

100 XP

3 minutos

Además de transformar texto y realizar operaciones básicas, como buscar y buscar coincidencias, es esencial dar formato al texto al presentar información. La manera más sencilla de presentar información de texto con Python consiste en usar la función `print()`. Comprobará que es fundamental incluir información en variables y otras estructuras de datos en cadenas que `print()` pueda usar.

En esta unidad, aprenderá varias maneras válidas de incluir valores de variable en el texto mediante Python.

## Formato de signo de porcentaje (%)

El marcador de posición de la variable de la cadena es `%s`. Después de la cadena, use otro carácter `%` seguido del nombre de la variable. En el ejemplo siguiente, se muestra cómo dar formato mediante el carácter `%`:

Python

Copiar

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

El uso de varios valores cambia la sintaxis, ya que se necesitan paréntesis para rodear las variables que se pasan:

Python

Copiar

```
>>> 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"))
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.
```

Sugerencia

Aunque este método sigue siendo una manera válida de dar formato a las cadenas, puede provocar errores y reducir la claridad del código cuando se trabaja con varias variables. Cualquiera de las otras dos opciones de formato descritas en esta unidad sería más adecuada para este propósito.

## El método format()

El método `.format()` usa llaves `{}` como marcadores de posición dentro de una cadena y utiliza la asignación de variables para reemplazar texto.

Python

Copiar

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

No es necesario asignar variables repetidas varias veces, lo que hace que sea menos detallado porque es necesario asignar menos variables:

Python

Copiar

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

En lugar de llaves vacías, la sustitución consiste en usar números. `{0}` significa usar el primer argumento (índice cero) de `.format()`, que en este caso es `Moon`. `{0}` funciona bien para una repetición simple, pero reduce la legibilidad. Para mejorar la legibilidad, use argumentos de palabra clave en `.format()` y, después, haga referencia a los mismos argumentos entre llaves:

Python

Copiar

```
>>> 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))
You are lighter on the Moon, because on the Moon you would weigh about 1/6 of your weight on Earth
```

## Acerca de las cadenas f-strings

A partir de la versión 3.6 de Python, es posible usar *f-strings*. Estas cadenas parecen plantillas y usan los nombres de variable del código. El uso de f-strings en el ejemplo anterior tendría el siguiente aspecto:

Python

Copiar

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

Las variables se incluyen entre llaves y la cadena *debe* usar el prefijo `f`.

Además de que las f-strings son menos detalladas que cualquier otra opción de formato, es posible usar expresiones entre llaves. Estas expresiones pueden ser funciones u operaciones directas. Por ejemplo, si quiere representar el valor `1/6` como un porcentaje con una posición decimal, puede usar directamente la función `round()`:

Python

Copiar

```
>>> round(100/6, 1)
16.7
```

Con f-strings, no es necesario asignar un valor a una variable de antemano:

Python

Copiar

```
>>> print(f"On the Moon, you would weigh about {round(100/6, 1)}% of your weight on Earth")
On the Moon, you would weigh about 16.7% of your weight on Earth
```

Para usar una expresión no es necesaria una llamada de función. Cualquiera de los métodos de cadena también son válidos. Por ejemplo, la cadena podría aplicar un uso específico de mayúsculas y minúsculas para crear un título:

Python

Copiar

```
>>> subject = "interesting facts about the moon"
>>> f"{subject.title()}"
'Interesting Facts About The Moon'
```

### Siguiente unidad: Ejercicio: Formato de cadenas

Continuar >

¿Necesita ayuda? Consulte nuestra [guía de solución de problemas](#) o [notifique un problema](#) para enviar comentarios específicos.

¿Qué tal lo estamos haciendo? ☆ ☆ ☆ ☆ ☆

Español (España)

Tema

Versiones anteriores

Blog

Contribuir

Privacidad

Términos de uso

Marcas comerciales

© Microsoft 2023