

Python Programming

Módulo 2

Formar cadenas

Formar cadenas

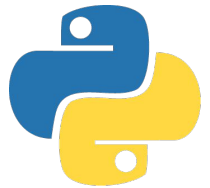
Es bastante usual tener que incluir otros tipos de datos dentro de una cadena, podemos usar la función `str()` para poder concatenar otros tipos de datos. Pero cuando debemos hacerlo con una gran cantidad de datos, sobre otro string, nos resultará conveniente utilizar el método `format()`.

```
>>> nombre = "Pablo"  
>>> edad = 30  
>>> mensaje = "Hola, mi nombre es {0}. Tengo {1} años."  
>>> mensaje.format(nombre, edad)  
'Hola, mi nombre es Pablo. Tengo 30 años.'
```



En el código creamos una cadena, en donde queremos incluir los objetos nombre y edad (una cadena y un entero, respectivamente). En su lugar, colocamos `{0}` y `{1}` (y así sucesivamente en relación con la cantidad de objetos que queramos incluir) y por último llamamos a `.format()` pasándole como argumentos los elementos que queremos insertar dentro de mensaje.

Las cadenas en Python son objetos inmutables, por lo que `.format()` retorna una nueva cadena con los valores reemplazados.



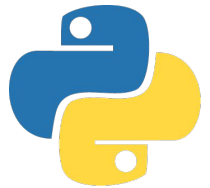
A partir de la versión 3.6 Python introduce una manera más legible, que consiste en poner nombres de variables entre llaves dentro de una cadena y anteponiendo la letra f.

```
>>> nombre = "Pablo"
>>> edad = 30
>>> mensaje = f"Hola, mi nombre es {nombre}. Tengo {edad} años."
>>> mensaje
'Hola, mi nombre es Pablo. Tengo 30 años.'
```

El lenguaje también soporta un sistema más antiguo proveniente del lenguaje C a través del operador %.

```
>>> "Hola, mi nombre es %s y tengo %d años." % (nombre, edad)
'Hola, mi nombre es Pablo y tengo 30 años.'
```

De todas formas, los métodos recomendados son los dos primeros.



¡Sigamos trabajando!