



Guía Rápida de *Strings*

IWI 131: Programación



Definición de *strings*

Los *strings* literales se pueden delimitar con comillas simples o dobles:

```
mensaje = 'Hola'
saludo = "Hola"
```

Lectura de *strings* como entrada

Es posible leer un *string* como entrada del programa utilizando `input()`:

```
nombre = input("Como te llamas? ")
```

Operadores sobre *strings*

+	Concatena dos <i>strings</i> , creando uno nuevo Ej: <code>mensaje = "Hola " + "mundo"</code>
*	Concatena un <i>string</i> , consigo mismo, una cantidad de veces, creando uno nuevo Ej: <code>mensaje = 3 * "Hola "</code>
[]	Recupera el caracter que se encuentra en una posición particular de un <i>string</i> ; el índice debe ser válido Ej: <code>print(nombre[5])</code> Índices de izq. a der.: 0, 1, 2, ... Índices de der. a izq.: -1, -2, -3, ...
[a:b]	Crea un nuevo <i>string</i> como una <i>rebanada</i> de un <i>string</i> existente; comenzando en el índice a y terminando uno antes de b. Ej: <code>palabra = texto[5:10]</code>
[:b]	Crea un nuevo <i>string</i> como una <i>rebanada</i> al inicio de un <i>string</i> existente, terminando uno antes de b. Ej: <code>prefijo = texto[:b]</code>
[a:]	Crea un nuevo <i>string</i> como una <i>rebanada</i> al final de un <i>string</i> existente, comenzando en a. Ej: <code>sufijo = texto[a:]</code>
c in s	Retorna <code>True</code> si el caracter o <i>string</i> c está contenido en el <i>string</i> s; de lo contrario retorna <code>False</code> Ej: <code>if "a" in nombre:</code> Puede utilizarse <code>not in</code> para preguntar en negativo

Funciones sobre *strings*

<code>len(s)</code>	Retorna la longitud del <i>string</i> s, es decir, la cantidad de caracteres que contiene Ej: <code>largo = len(nombre)</code>
<code>ord(c)</code>	Retorna el código ASCII del caracter c; el código retornado es un número entero Ej: <code>codigo = ord("A")</code>
<code>chr(n)</code>	Retorna un <i>string</i> que contiene el caracter cuyo código ASCII es el entero n Ej: <code>caracter = chr(65)</code>

Métodos sobre *strings*

<code>s.upper()</code>	Retorna un nuevo <i>string</i> , convirtiendo a mayúsculas las letras del <i>string</i> original que puedan ser convertidas; el resto se copian sin cambios Ej: <code>mayusculas = nombre.upper()</code>
<code>s.lower()</code>	Retorna un nuevo <i>string</i> , convirtiendo a minúsculas las letras del <i>string</i> original que puedan ser convertidas; el resto se copian sin cambios Ej: <code>minusculas = nombre.lower()</code>

Diferencia entre funciones y métodos

La característica que diferencia a los métodos de las funciones es la forma de llamarlos. En una función aplicada a un *string*, como `len(s)`, se escribe el nombre de la función y el *string* va dentro de los paréntesis. En un método aplicado a un *string*, se comienza con el *string*, seguido de un punto, seguido del nombre del método y dentro de los paréntesis se incluyen parámetros adicionales. Por ejemplo: `s.upper()`.

Comparaciones entre *strings*

<code>s1 == s2</code>	Retorna <code>True</code> si los <i>strings</i> son exactamente iguales, o <code>False</code> si no lo son. Ej: <code>if nombre == "Ada":</code>
<code>s1 < s2</code>	Retorna <code>True</code> si <code>s1</code> precede a <code>s2</code> según el orden lexicográfico, o <code>False</code> en caso contrario; puede usarse también <code><=</code> para menor o igual.
<code>s1 > s2</code>	La comparación opuesta a <code><</code> ; puede usarse también <code>>=</code> para mayor o igual.

Las comparaciones utilizan el orden de los caracteres en la tabla ASCII. Las comparaciones se hacen desde el primer caracter, hasta encontrar una pareja de caracteres que sean distintos. El resultado es el orden lexicográfico al que estamos acostumbrados cuando ordenamos alfabéticamente.

Recorrido de *strings* con `while`

```
i = 0
while i < len(s):
    print(s[i])
    i = i + 1
```

Recorrido de *strings* con `for`

```
for caracter in s:
    print(caracter)
```

Notas

(1) Los letras mayúsculas y minúsculas son caracteres distintos, pues tienen código ASCII distintos. Por ejemplo, el código ASCII de "A" es 65 mientras que el de "a" es 97.

(2) Los *strings* son inmutables, es decir, no pueden cambiar una vez creados. Si se quiere cambiar el contenido de una variable de tipo *string*, se debe crear un nuevo *string* y asignarlo a la variable.
Ej: `mayusculas = nombre.upper()`