

### 5.1.5. Recorrido de cadenas

Una propiedad interesante de los datos secuenciales es que pueden recorrerse de izquierda a derecha con un bucle **for-in**. Por ejemplo, el siguiente bucle recorre los caracteres de una cadena de uno en uno, de izquierda a derecha:

```
>>> for caracter in "mi_cadena": ↵
...     print caracter ↵
...     ↵
m
i

c
a
d
e
n
a
```

En cada paso, la variable del bucle (en el ejemplo, *caracter*) toma el valor de uno de los caracteres de la cadena. Es lo que cabía esperar: recuerda que el bucle **for-in** recorre uno a uno los elementos de una serie de valores, y una cadena es una secuencia de caracteres.

Tienes una forma alternativa de recorrer los elementos de una cadena: recorriendo el rango de valores que toma su índice e indexando cada uno de ellos. Estudia este ejemplo:

```
>>> a = "mi_cadena" ↵
>>> for i in range(len(a)): ↵
...     print a[i] ↵
...     ↵
m
i

c
a
d
e
n
a
```

La variable *i* toma los valores de `range(len(a))`, en este caso los valores comprendidos entre 0 y 8, ambos inclusive. Con `a[i]` hemos accedido, pues, a cada uno de ellos. Si mostramos tanto *i* como `a[i]`, quizás entiendas mejor qué ocurre exactamente:

```
>>> a = "mi_cadena" ↵
>>> for i in range(len(a)): ↵
...     print i, a[i] ↵
...     ↵
0 m
1 i
2
3 c
4 a
5 d
6 e
7 n
8 a
```

También puedes mostrar los caracteres de la cadena en orden inverso, aunque en tal caso has de hacerlo necesariamente con un bucle **for-in** y un *range*:

```
>>> a = "mi_cadena" ↵
>>> for i in range(len(a)): ↵
...     print a[len(a)-i-1] ↵
...     ↵
a
n
e
d
a
c

i
m
```

..... EJERCICIOS .....

► **157** Intentamos mostrar los caracteres de la cadena en orden inverso así:

```
>>> a = "mi_cadena" ↵
>>> for i in range(len(a), -1): ↵
...     print a[i] ↵
...     ↵
```

¿Funciona?

► **158** Intentamos mostrar los caracteres de la cadena en orden inverso así:

```
>>> a = "mi_cadena" ↵
>>> for i in range(len(a)-1, -1, -1): ↵
...     print a[i] ↵
...     ↵
```

¿Funciona?

► **159** Diseña un programa que lea una cadena y muestre el número de espacios en blanco que contiene.

► **160** Diseña un programa que lea una cadena y muestre el número de letras mayúsculas que contiene.

► **161** Diseña una programa que lea una cadena y muestre en pantalla el mensaje «**Contiene dígito**» si contiene algún dígito y «**No contiene dígito**» en caso contrario.

.....