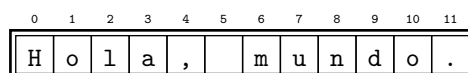


5.1.4. Indexación

Podemos acceder a cada uno de los caracteres de una cadena utilizando un operador de *indexación*. El índice del elemento al que queremos acceder debe encerrarse entre corchetes. Si a es una cadena, $a[i]$ es el carácter que ocupa la posición $i+1$. Debes tener en cuenta que el primer elemento tiene índice cero. Los índices de la cadena 'Hola, mundo.' se muestran en esta figura:



```
>>> 'Hola, mundo.' [0] ↵
'H'
>>> 'Hola, mundo.' [1] ↵
'o'
>>> a = 'Hola, mundo.' ↵
>>> a [2] ↵
'l'
>>> a [1] ↵
'o'
>>> i = 3 ↵
>>> a [i] ↵
'a'
>>> a [len(a)-1] ↵
'.'
```

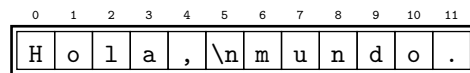
Observa que el último carácter de la cadena almacenada en la variable a no es $a[\text{len}(a)]$, sino $a[\text{len}(a)-1]$. ¿Por qué? Evidentemente, si el primer carácter tiene índice

0 y hay $len(a)$ caracteres, el último ha de tener índice $len(a)-1$. Si intentamos acceder al elemento $a[len(a)]$, Python protesta:

```
>>> a[len(a)] ↵
Traceback (innermost last):
  File "<stdin>", line 1, in ?
IndexError: string index out of range
```

El error cometido es del tipo *IndexError* (error de indexación) y, en el texto explicativo que lo detalla, Python nos informa de que el índice de la cadena está fuera del rango de valores válidos.

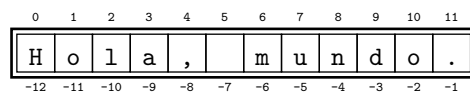
Recuerda que las secuencias de escape codifican caracteres simples, aunque se expresen con dos caracteres. La cadena 'Hola, \nmundo.', por ejemplo, no ocupa 13 casillas, sino 12:



También puedes utilizar índices negativos con un significado especial: los valores negativos acceden a los caracteres de derecha a izquierda. El último carácter de una cadena tiene índice -1 , el penúltimo, -2 , y así sucesivamente. Analiza este ejemplo:

```
>>> a = 'Ejemplo' ↵
>>> a[-1] ↵
'o'
>>> a[len(a)-1] ↵
'o'
>>> a[-3] ↵
'p'
>>> a[-len(a)] ↵
'E'
```

De este modo se simplifica notablemente el acceso a los caracteres del final de la cadena. Es como si dispusieras de un doble juego de índices:



..... EJERCICIOS

► **156** La última letra del DNI puede calcularse a partir de sus números. Para ello sólo tienes que dividir el número por 23 y quedarte con el resto. El resto es un número entre 0 y 22. La letra que corresponde a cada número la tienes en esta tabla:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

Diseña un programa que lea de teclado un número de DNI y muestre en pantalla la letra que le corresponde.

(Nota: una implementación basada en tomar una decisión con **if-elif** conduce a un programa muy largo. Si usas el operador de indexación de cadenas de forma inteligente, el programa apenas ocupa tres líneas. Piensa cómo.)

.....