

UVA 5

Strings

Prof. Federico Meza



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO
DE INFORMÁTICA

Longitud e índices

```
frase = 'Solo vuela el que se atreve a hacerlo'
```

- a. Longitud, ¿con qué función se obtiene?

R/ `37`

- b. Índices válidos

R/ `0` a `36`

Alternativamente: `-37` a `-1`

- c. Índice del último carácter, en términos de la longitud

R/ `len(frase) - 1`

Longitud e índices

```
frase = 'Solo vuela el que se atreve a hacerlo'
```

d. ¿Cómo podemos recorrer los índices del string?:

```
i = 0
while i < ??? :
    i += 1
```

```
R/ len(frase)
```

Conversión de tipos (*casting*)

a. Convertir "23515" a tipo `int`

R/ `int("23515")`

b. Convertir "veintitrés mil quinientos quince" a tipo `int`

R/ No se puede, directamente

ASCII y orden lexicográfico

a. ¿Resultado de: `'perro' < 'gato' ?`

R/ False

a. ¿Resultado de: `'perro' == 'Perro' ?`

R/ False

a. ¿Resultado de: `'perro' < 'Perro' ?`

R/ False, las mayúsculas van primero en ASCII

Tabla ASCII

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[59	3B	;	91	5B	[123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

Tabla ASCII (extendida)

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	Ł	224	E0	α
129	81	ù	161	A1	í	193	C1	ł	225	E1	β
130	82	é	162	A2	ó	194	C2	ŧ	226	E2	Γ
131	83	â	163	A3	ú	195	C3	ţ	227	E3	π
132	84	ä	164	A4	ñ	196	C4	—	228	E4	Σ
133	85	å	165	A5	Ñ	197	C5	†	229	E5	σ
134	86	ä	166	A6	ª	198	C6	‡	230	E6	μ
135	87	ç	167	A7	º	199	C7	‡	231	E7	ι
136	88	ê	168	A8	¿	200	C8	Ł	232	E8	φ
137	89	ë	169	A9	ˆ	201	C9	Ɔ	233	E9	θ
138	8A	è	170	AA	˜	202	CA	Ł	234	EA	Ω
139	8B	ı	171	AB	½	203	CB	Ɔ	235	EB	δ
140	8C	î	172	AC	¼	204	CC	Ɔ	236	EC	∞
141	8D	ï	173	AD	ı	205	CD	=	237	ED	φ
142	8E	Ä	174	AE	«	206	CE	Ł	238	EE	ε
143	8F	Å	175	AF	»	207	CF	Ł	239	EF	Ω
144	90	Ê	176	B0	ˆ	208	D0	Ł	240	F0	≡
145	91	æ	177	B1	ˆ	209	D1	Ɔ	241	F1	±
146	92	Æ	178	B2	ˆ	210	D2	Ɔ	242	F2	≥
147	93	ø	179	B3	ı	211	D3	Ł	243	F3	≤
148	94	ó	180	B4	ı	212	D4	Ö	244	F4	ƒ
149	95	ò	181	B5	ı	213	D5	Ɔ	245	F5	
150	96	û	182	B6	ı	214	D6	Ɔ	246	F6	+
151	97	ü	183	B7	ı	215	D7	†	247	F7	≈
152	98	ÿ	184	B8	ı	216	D8	‡	248	F8	≈
153	99	Û	185	B9	ı	217	D9	‡	249	F9	·
154	9A	Ü	186	BA	ı	218	DA	ı	250	FA	·
155	9B	Ÿ	187	BB	ı	219	DB	ı	251	FB	√
156	9C	£	188	BC	ı	220	DC	ı	252	FC	π
157	9D	¥	189	BD	ı	221	DD	ı	253	FD	z
158	9E	Pts	190	BE	ı	222	DE	ı	254	FE	■
159	9F	f	191	BF	ı	223	DF	ı	255	FF	

Inmutabilidad

a. ¿Qué imprime?

```
nombre = 'Juan Carlos Bodoque'  
nombre.upper()  
print(nombre)
```

R/ Juan Carlos Bodoque (¿por qué?)

b. ¿Cuál es la forma correcta?

```
nombre = 'Juan Carlos Bodoque'  
nombre = nombre.upper()  
print(nombre)
```

R/ JUAN CARLOS BODOQUE

Inmutabilidad

c. ¿Qué hace?

```
nombre = 'Juan Carlos Bodoque'  
print(nombre[3])
```

R/ n

d. ¿Y esto qué hace?

```
nombre = 'Juan Carlos Bodoque'  
nombre[3] = 'A'
```

R/ Produce un error (¿por qué?)

Pertenencia

a. `vocales = 'aeiou'`

`'e' in vocales`

R/ `True`

b. `'E' in 'Perro'`

R/ `False`

c. `'pollo' in 'Vaca Pollo Cerdo Tomate Lechuga Repollo Zapallo'`

R/ `True`

Recuperación de elementos

```
texto = 'gato grande, negro y gordo'
```

- a. `texto[4]`
R/ ' ' (espacio en blanco)
- b. `len(texto[5:8])`
R/ 3
- c. `texto[:4]`
R/ 'gato'
- d. `texto[-5:]`
R/ 'gordo'

Recorrido de strings: `for`

a. ¿Qué hace el siguiente programa?

```
texto = 'gato grande, negro y gordo'
for x in texto:
    if x in 'aeiou':
        print(x)
```

b. ¿Cómo lo hacemos con `while` en vez de `for`?

¿Cuándo usamos `for` y cuándo `while`?

- Usar `for` para recorrer un *string* es más fácil, pues no usamos los índices. La instrucción nos va entregando los caracteres uno por uno, para implementar nuestro algoritmo.
- Pero a veces necesitamos los índices, para saber la posición de los caracteres que estamos procesando. En ese caso tenemos que usar `while`.

Ejercicio

Escriba un programa que lea como entrada un texto y una letra, e imprima la cantidad de veces que la letra aparece en el texto.

```
Ingrese palabra: paralelepipedo
```

```
Ingrese letra: a
```

```
Aparece 2 veces
```

```
Ingrese palabra: Paralelepipedo
```

```
Ingrese letra: e
```

```
Aparece 3 veces
```

```
Ingrese palabra: Paralelepipedo
```

```
Ingrese letra: f
```

```
Aparece 0 veces
```

Ejercicio

Escriba un programa que lea como entrada un texto e imprima la cantidad de palabras que contiene.

Ingrese palabra: **gato grande, negro y gordo**

Contiene 5 palabras

Ingrese palabra: **hola**

Contiene 1 palabras

Ejercicio

Escriba un programa que indique las letras que coinciden (la misma letra en la misma posición) en dos *strings* leídos como entrada.

Por ejemplo, "amorosos" y "amortiza" coinciden en: "amor"; por otra parte, "conformidad" y "contorno" coinciden en "conor".

Observe que los strings pueden tener distintos largos.

Ejercicio

[25 %] En el básquetbol existen tres diferentes tipos de anotaciones:

- el tiro libre (L), que vale un punto,
- el doble (D), que vale dos puntos, y
- el triple (T), que vale tres puntos.

Un partido de básquetbol está dividido en varios períodos.

Usted debe escribir un programa que reciba como entrada una única línea, que contenga todas las anotaciones realizadas por un equipo de básquetbol durante un partido. Las anotaciones de períodos distintos deben ir separadas por un espacio. Como salida, debe mostrar la cantidad de puntos obtenidos en cada período y los puntos totales, siguiendo el formato del ejemplo.

```
Anotaciones: DDTDLLDD DDLDT TDTLLD DDDDD
15 puntos en el periodo 1
10 puntos en el periodo 2
12 puntos en el periodo 3
10 puntos en el periodo 4
```