

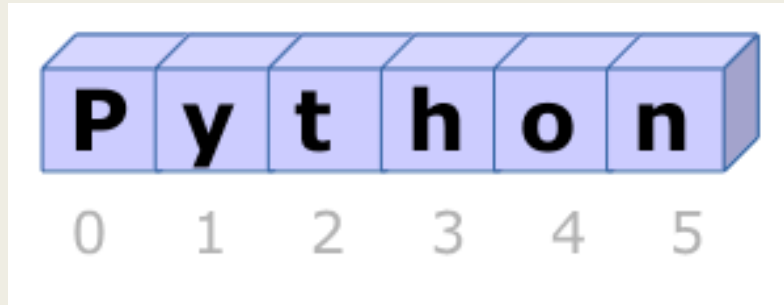
A thick black L-shaped frame surrounds the text. It starts at the top left, goes right, then down, then right again at the bottom right.

INTRODUCCIÓN A LA PROGRAMACIÓN STRINGS

Rodrigo López A.

rilopez3@uc.cl

Strings en



- Un String es una secuencia de caracteres.
- Hasta el momento los hemos usado para recibir datos y mostrar mensajes, pero sus usos son mucho más amplios.
- Las próximas clases veremos como al poder calcular el largo de un texto, obtener segmentos, recorrer sus componentes mediante un ciclo, entre otros, abre una amplia gama de posibilidades.

Largo y contenido de un String

- Largo de un String: utilizando la función provista por Python `len()`, es posible obtener la cantidad de caracteres.

`len('palabra') = 7`

`len('a b') = 3`

`len('') = 0`

- Python nos permite recorrer todos los caracteres de una cadena de manera muy sencilla, usando un ciclo.

```
For i in 'palabra':  
    print(i)
```

p

a

l

a

b

r

a

Indexing

- Para obtener el carácter en la posición i de un String, Python nos provee de una notación con corchetes: escribiremos $a[i]$ para hablar de la posición i -ésima de la cadena a .

$a = \text{'palabra'}$

$a[0] = \text{'p'}$

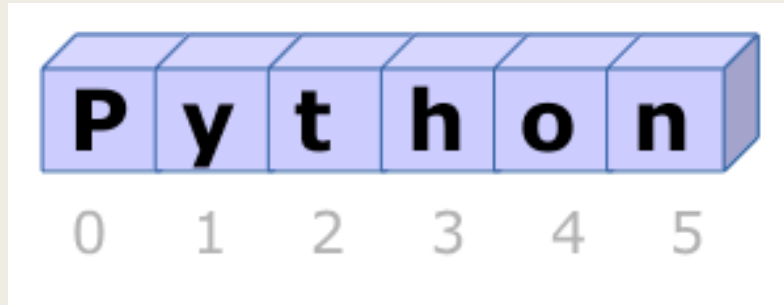
$a[2] = \text{'l'}$

$a[\text{len}(a)] = \text{ERROR !}$

- Las distintas posiciones de una cadena a se llaman índices.
- La primera posición es el índice 0.
- Los índices son números enteros que pueden tomar valores entre $-\text{len}(a)$ y $\text{len}(a) - 1$.
- Los índices negativos proveen una notación que permite indexar los caracteres en orden inverso:
 - $a[-1]$ es el último carácter de a
 - $a[-2]$ es el penúltimo carácter de a
 - $a[-\text{len}(a)]$ es el primer carácter de a

-6	-5	-4	-3	-2	-1
0	1	2	3	4	5
+---+---+---+---+---+					
a	b	c	d	e	f
+---+---+---+---+---+					

Strings en



Slicing: una notación especial para obtener segmentos de una cadena.

- De forma similar a los parámetros de la función range(inicio, termino, salto)
- Es posible obtener un segmento de un string de la forma palabra[inicio : termino]

ejemplo = 'frase de texto'

ejemplo[:] = ejemplo[0 : len(ejemplo)] = ejemplo[0 :] = 'frase de texto'
ejemplo[1 : 3] = 'ra' ejemplo[-4 : -1] = 'ext'

- Es posible agregar un 3 parámetro que permite definir el salto
palabra[inicio: termino: salto]

ejemplo[: : -1] = 'otxet ed esarf' ejemplo[: : 2] = 'faed et'

ejemplo[0:3:2] = ?? ejemplo[0:0] = ?? ejemplo[2:1] = ?? ejemplo[2:1:-1] = ??

Composición de Strings

- Como hemos visto los String son cadenas de caracteres, por lo cual es posible que un string sea parte de otro string.
- Para verificar esta situación es posible utilizar la sentencia `in` , la cual nos entregara un valor verdadero si se cumple y falso cuando no sea así.

`'a' in 'abcde' = True`

`'ae' in 'abcde' = False`

`'bcd' in 'abcde' = True`

`'a b' in 'yxza bcde' = True`

`'b'+ 'c' in 'abcde' = ??`

`'ab' in 'aBcDe' = ??`

`'cion' in 'explicación' = ??`

Métodos String

- `STRING.upper()`: convierte todos los caracteres del string a letra **mayúscula**.
- `STRING.lower()`: convierte todos los caracteres del string a letra **minúscula**.
- `STRING.isupper()`: devuelve **True** si todos los caracteres del string están en mayúscula.
- `STRING.islower()`: devuelve **True** si todos los caracteres del string están en minúscula.

- `STRING.isalpha()`: devuelve **True** si todos los caracteres del string son letras.
- `STRING.isalnum()`: devuelve **True** si todos los caracteres del string son letras o números.
- `STRING.isdecimal()`: devuelve **True** si todos los caracteres del string son números.
- `STRING.isspace()`: devuelve **True** si todos los caracteres del string son espacios vacíos.
- `STRING.istitle()`: devuelve **True** si la primera letra de cada palabra comienza con mayúscula con el resto en minúscula.
- `STRING.startswith(var)`: devuelve **True** si el string comienza con la variable var
- `STRING.endswith(var)`: devuelve **True** si el string termina con la variable var