

# strings

Si quisiéramos saber **¿cuál es la primera letra de un `string`?**

¿Cómo podríamos hacerlo?

En realidad, un `string` es una secuencia de caracteres.

Por ejemplo, el `string` "hola" en realidad es una concatenación de los caracteres "h"."o"."l"."a".

Carácter	→	h	o	l	a
Posición	→	0	1	2	3

## Métodos para cadenas de caracteres

Notemos que los caracteres se empiezan a contar desde el 0, y no desde el 1. Esto es sumamente importante, ya que al programar en general uno empieza a contar desde el 0 y no desde el 1.

CÓDIGO	RESULTADO
<pre>s = "hola" print(s[0])</pre>	<pre>h</pre>

En este caso, nuestra variable `string` que tiene el texto "hola" se llama `s`. Podemos extraer la primera letra en la posición 0 ocupando `[0]`, y así para cualquier posición.

Notemos que para extraer la letra de cualquier posición, ocupamos la siguiente notación:

```
nombre_variable_texto[posición]
```

¿Qué salida genera el siguiente código?

```
texto1 = "Los datos"  
texto2 = "Móvil"
```

```
print(texto1[3] + texto2[1])
```



Carácter	P	A	L	A	B	R	A
Posición	0	1	2	3	4	5	6

Es importante notar que la posición del último carácter está dada por `largo_texto` menos 1. Esto es porque las posiciones se empiezan a contar desde el 0.

El largo de la palabra en Python es 7. Por lo tanto, si queremos extraer la última letra, tenemos que buscar el carácter que está en la posición 6 (largo de la palabra menos 1).

# Comprobar cadenas

Para comprobar si una determinada frase o carácter está presente en una cadena, podemos usar la palabra clave **in**.

```
txt = "Alumno Inacap de programación"  
print("programación" in txt)
```

¿Qué salida genera el siguiente código?



## Método Slice

Esta acción nos permite extraer una secuencia de caracteres de un string. Se ocupa de la siguiente manera.

`s[i:j]` Esta acción extrae los caracteres que se encuentran desde la posición i (posición inicial) hasta la posición j-1 (posición final).

```
string_de_ejemplo = "texto de ejemplo"  
primera_palabra = string_de_ejemplo[0:5]  
print(primera_palabra)
```

¿Qué salida genera el siguiente código?

T	E	X	T	O		D	E		E	J	E	M	P	L	O
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

# Funciones de strings

## len()

Para determinar el número de caracteres en una cadena se utiliza la función **len**.

```
mensaje4 = 'hola' + ' ' + 'mundo'
```

```
print(len(mensaje4))
```

```
s="Texto"
```

```
largo_del_texto=len(s)
```



¿Qué salida genera el siguiente código?

## find()

`s.find(x)`

Retorna la posición de x en s.

Permite buscar una sub-cadena en una cadena de caracteres, el resultado será el índice de inicio de la misma

```
mensaje5="Hola Mundo"
```

```
mensaje5a=mensaje5.find("Mundo")
```

```
print(mensaje5a)
```



¿Qué salida genera el siguiente código?

# Funciones de strings

## count()

`s.count(x)`

Cuenta apariciones de x en s.

Cuenta las apariciones de un carácter o expresión en una cadena.

```
s="palabra"  
print(s.count("a"))
```



```
txt="Amo las manzanas, las manzanas  
son mi fruta favorita"  
x=txt.count("manzanas", 10, 24)  
print(x)
```

¿Qué salida genera el siguiente código?

## replace()

`s.replace(x, y)`

Reemplaza x por y en el string s.

Reemplaza un carácter por otro en una cadena.

```
s="palabra"  
print(s.replace("a","b"))
```

```
x="papapepa"  
print(x.replace("pa","tx"))
```



¿Qué salida genera el siguiente código?

# Funciones de strings

## `s.upper()`

Transforma todos los caracteres del string a mayúsculas.

```
s="Texto"  
print(s.upper())
```



¿Qué salida genera el siguiente código?

## `s.lower()`

Transforma todos los caracteres del string a minúsculas.

```
s="Texto"  
print(s.lower())
```



¿Qué salida genera el siguiente código?

## `strip()`

Remueve espacios y otros caracteres indeseables al principio y al final del string.



```
s="\nPALABRA"  
print(s.strip())
```

¿Qué salida genera el siguiente código?



inacap.cl