



FACULTAD DE INGENIERÍA



PYTHON

Manejo de Cadenas de Caracteres

Lic. Gustavo Bianchi

Junio 2023

Contenido

- Estructura Interna, composición y organización.
- Cómo acceder a los caracteres que la componen.
- Distintos modos de asignar cadenas.

¿Qué es una cadena de caracteres?

- Una cadena de caracteres es una secuencia ordenada de caracteres.
- El primer carácter ocupa la posición 0.
- Es **INMUTABLE**, su contenido no puede ser modificado.

¿Cómo las asignamos?

```
>>> nombre = 'Juan'
```

```
>>> apellido = "Perez"
```

Pueden estar
encerradas entre
comillas simples ó
dobles

¿Cómo se concatenan?

```
>>> nombre + apellido
```

```
'JuanPerez'
```

```
>>> apellido + ', ' + nombre
```

```
'Perez, Juan'
```

Se pueden
concatenar
mediante el
signo +

Posicionamiento: ¿Cómo accedemos a los caracteres?

Se puede acceder a cada uno de los caracteres de la cadena, mediante un índice.

```
>>> palabra = "Algoritmo"
```

```
>>> palabra[0]
```

```
'A'
```

```
>>> palabra[2]
```

```
'g'
```

```
>>> palabra[-1]
```

```
'o'
```

```
>>> palabra[-3]
```

```
't'
```

*El primer carácter ocupa la posición 0
En este caso, voy a obtener la letra “A”*

*También puedo acceder desde la derecha
de la cadena, siendo la posición del último
carácter: -1*

Particularidades: Carácter de exclusión '\'

Si las comillas deben formar parte de la cadena, entonces se deben combinar comillas simples y dobles, ó anteponer el carácter de exclusión '\'

```
>>> linea = 'Ella dijo: "No fue mi culpa"'
```

ó

```
>>> linea = "Ella dijo: \"No fue mi culpa\""
```

Carácter de Exclusión

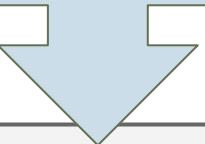
```
>>> linea
```

```
'Ella dijo: "No fue mi culpa"'
```

Particularidades: Cadenas de varias líneas

Si queremos almacenar varias líneas de texto, debemos encerrarlas entre triples comillas, ya sea simples, ó dobles

```
>>> texto = """Bello es mejor que feo.  
Explícito es mejor que implícito.  
Simple es mejor que complejo.  
Complejo es mejor que complicado."""
```



Carácter de Nueva Línea

```
>>> texto
```

```
'Bello es mejor que feo. \nExplícito es  
mejor que implícito. \nSimple es mejor  
que complejo. \nComplejo es mejor que  
complicado.'
```

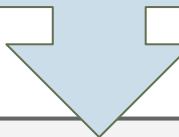
```
>>> print(texto)
```

```
Bello es mejor que feo.  
Explícito es mejor que implícito.  
Simple es mejor que complejo.  
Complejo es mejor que complicado.
```

Particularidades: Alternativa cadena de varias oraciones

Otra alternativa cuando queremos asignar un texto compuesto por varias oraciones, es encerrar el texto entre paréntesis y cada cadena encerrada entre comillas.

En este caso, el resultado será una concatenación de todas las cadenas.



```
>>> texto = ("Texto de la oración 1. "
              "Texto de la oración 2. "
              "Texto de la oración 3.")
```

```
>>> texto
```

'Texto de la oración 1. Texto de la oración 2. Texto de la oración 3.'

```
>>> print(texto)
```

Texto de la oración 1. Texto de la oración 2. Texto de la oración 3.

www.ingenieria.uba.ar



/ingenieriauba



/FIUBAoficial