

A3 – Métodos de la clase str y manipulación básica de cadenas

Objetivo de la actividad

Que los equipos (**2 a 3 personas**) comprendan y apliquen métodos fundamentales de la clase str en Python.

Introducción conceptual

Conceptos fundamentales sobre objetos str y sus métodos

1. ¿Qué es un objeto str?

En Python, una cadena de texto (string) es un **objeto de la clase str**.

Esto significa que:

- Tiene **propiedades** (como su longitud).
- Tiene **métodos** (funciones internas) que permiten manipularla.
- Es **inmutable**: cada vez que aplicas un método, se genera una **nueva cadena**, nunca se modifica la original.

2. ¿Qué significa que un método reciba el objeto que lo invoca como primer argumento implícito?

Cuando escribimos:

```
texto.lower()
```

Python internamente interpreta algo equivalente a:

```
str.lower(texto)
```

Es decir:

- El **primer argumento** del método es **el objeto que lo llama**.
- No lo escribimos explícitamente porque Python lo pasa automáticamente.

Por eso decimos que el objeto es el **primer argumento implícito**.

Ejemplo:

```
"Hola".replace("H", "M")
```

Internamente es:

```
str.replace("Hola", "H", "M")
```

3. ¿Qué parámetros adicionales recibe cada método?

Además del objeto que lo invoca, cada método puede recibir **cero, uno o varios parámetros**.

Los parámetros permiten personalizar el comportamiento del método.

4. ¿Qué retorna cada método?

Cada método de str retorna un tipo de dato específico, dependiendo de su propósito **otra cadena (str), listas (list[str]), booleanos (True o False) o enteros (como .count())**.

Métodos relevantes

Método	Parámetros adicionales	Retorno	Uso típico	Ejemplo
<code>.strip()</code>	Ninguno o caracteres a eliminar	str	Quitar espacios al inicio y final	" hola ".strip() → "hola"
<code>.lstrip()</code>	Ninguno o caracteres a eliminar	str	Quitar espacios solo al inicio	" hola".lstrip() → "hola"
<code>.rstrip()</code>	Ninguno o caracteres a eliminar	str	Quitar espacios solo al final	"hola ".rstrip() → "hola"
<code>.lower()</code>	Ninguno	str	Convertir a minúsculas	"Hola".lower() → "hola"
<code>.upper()</code>	Ninguno	str	Convertir a mayúsculas	"Hola".upper() → "HOLA"
<code>.capitalize()</code>	Ninguno	str	Primera letra en mayúscula	"hola mundo".capitalize() → "Hola mundo"
<code>.title()</code>	Ninguno	str	Capitalizar cada palabra	"hola mundo".title() → "Hola Mundo"
<code>.split()</code>	Separador opcional	list[str]	Dividir texto en palabras	"a b c".split() → ["a","b","c"]
<code>.join(lista)</code>	Lista de cadenas	str	Unir elementos con un separador	" ".join(["a","b"]) → "a b"
<code>.replace(a, b)</code>	Texto a buscar, texto a reemplazar	str	Sustituir texto	"hola".replace("h","m") → "mola"
<code>.startswith(x)</code>	Prefijo	bool	Verificar inicio	"Hola".startswith("H") → True
<code>.endswith(x)</code>	Sufijo	bool	Verificar final	"archivo.txt".endswith(".txt") → True
<code>.count(x)</code>	Subcadena a contar	int	Contar ocurrencias	"banana".count("a") → 3
<code>.find(x)</code>	Subcadena	int	Buscar posición (o -1)	"hola".find("o") → 1
<code>.index(x)</code>	Subcadena	int	Buscar posición (error si no existe)	"hola".index("o") → 1
<code>.isdigit()</code>	Ninguno	bool	Verificar si son solo dígitos	"123".isdigit() → True
<code>.isalpha()</code>	Ninguno	bool	Verificar si son solo letras	"Hola".isalpha() → True
<code>.isalnum()</code>	Ninguno	bool	Letras o números	"Hola123".isalnum() → True
<code>.swapcase()</code>	Ninguno	str	Invertir mayúsculas/minúsculas	"HoLa".swapcase() → "hOlA"
<code>.center(n)</code>	Ancho total	str	Centrar texto	"hola".center(10) → " hola "

Retos a resolver

◆ *Reto 1 — Convertir una frase a snake_case*

Descripción: Dada una frase, convertirla a minúsculas y reemplazar los espacios por guiones bajos.

Dato de prueba obligatorio: "Hola Mundo desde Python "

Resultado esperado: hola_mundo_desde_python

◆ *Reto 2 — Contar cuántas palabras tiene una frase*

Descripción: Separar la frase en palabras y contar cuántas hay.

Dato de prueba obligatorio: "Los métodos de string son muy útiles"

Resultado esperado: 6

◆ *Reto 3 — Obtener las iniciales de una frase*

Descripción: Tomar la primera letra de cada palabra, concatenarlas y convertirlo todo a mayúsculas.

Dato de prueba obligatorio: "Universidad Autónoma de Nuevo León"

Resultado esperado: UADNL

◆ *Reto 4 — Invertir el orden de las palabras*

Descripción: Invertir el orden de las palabras sin alterar su contenido.

Dato de prueba obligatorio: "Programar en Python es divertido"

Resultado esperado: divertido es Python en Programar

◆ *Reto 5 — Encontrar la palabra más larga*

Descripción: Identificar la palabra con mayor longitud dentro de una frase.

Dato de prueba obligatorio: "Aprender programación requiere práctica constante"

Resultado esperado: programación

Entregables

Cada equipo deberá entregar en MS Teams:

1. Archivo .py

Debe contener:

- Comentarios con los nombres de los integrantes
- Código para resolver **los 5 retos**
- Cada reto separado y claramente identificado
- Uso de los **datos de prueba obligatorios**

2. Screenshot de la ejecución

Debe mostrar:

- La ejecución del programa
- Los resultados correctos para los datos de prueba
- Que todos los retos funcionan