



# Clase 4: Manipulación de Cadenas de Texto en Python



## 1. Características de las Cadenas

Hasta ahora hemos visto dos características de las variables: el nombre y el tipo de datos que almacenan. Uno de los tipos de datos que podemos usar es el de cadenas o `strings`. Para crear una variable de tipo cadena, simplemente le asignamos un valor de texto:

```
texto = "Hola Mundo"
print(texto)
```

Para verificar el tipo de dato, usamos la función `type()`:

```
print(type(texto)) # <class 'str'>
```

En Python, las cadenas se representan con la clase `str`. A diferencia de otros lenguajes, podemos utilizar tanto un solo carácter como una cadena de texto más larga. Vamos a ver un ejemplo:

```
caracter = 'a'
print(type(caracter)) # <class 'str'>
```

## 2. Representación de Cadenas 📖

Existen tres maneras de representar cadenas en Python:

- **Comillas Dobles:**

```
texto = "Hola Mundo"
print(texto)
```

- **Comillas Simples:**

```
texto = 'Hola Mundo'
print(texto)
```

- **Comillas Triples** (simples o dobles):

```
texto = """Hola Mundo"""
print(texto)
```

Las comillas triples permiten usar saltos de línea dentro de la cadena:

```
texto = """Hola
Mundo"""
print(texto)
```

## 3. Indexación de Cadenas

Las cadenas tienen una propiedad llamada indexación, que nos permite acceder a caracteres específicos dentro de la cadena. Los índices empiezan en 0:

```
texto = "Hola Mundo"
print(texto[0]) # H
print(texto[1]) # o
print(texto[2]) # l
```

Para acceder a los caracteres desde el final de la cadena, usamos índices negativos:

```
print(texto[-1]) # o
print(texto[-2]) # d
```

## 4. Operaciones con Cadenas

- **Concatenación:**

Puedes sumar cadenas con el operador

 :

```
nombre = "Carlos"
apellido = "Mendoza"
nombre_completo = nombre + " " + apellido
print(nombre_completo) # Carlos Mendoza
```

- **Repetición:**

Puedes repetir una cadena usando el operador

 :

```
texto = "Hola "
texto_repetido = texto * 3
print(texto_repetido) # Hola Hola Hola
```

## 5. Métodos de Cadenas

Las cadenas tienen métodos específicos que puedes usar para manipular el texto. Por ejemplo:

- **Convertir a Minúsculas:**

```
texto = "Hola Mundo"
print(texto.lower()) # hola mundo
```

- **Convertir a Mayúsculas:**

```
texto = "Hola Mundo"
print(texto.upper()) # HOLA MUNDO
```

- **Eliminar Espacios:**

```
texto = "  Hola Mundo  "
print(texto.strip()) # Hola Mundo
```

## 6. Prácticas Recomendadas

Para mantener un código limpio y legible, sigue las buenas prácticas, como usar comillas consistentes y agregar espacios para mejorar la legibilidad.

```
texto = "Hola Mundo"
print(texto.lower()) # hola mundo
```

## 7. Aplicaciones Avanzadas

Las cadenas también se usan en áreas avanzadas como el procesamiento del lenguaje natural (NLP). La limpieza y manipulación de texto son cruciales en estos casos.

Para explorar más métodos y funciones de cadenas, te recomiendo revisar la documentación oficial de Python. 