

1. upper(): Convierte todos los caracteres de la cadena a mayúsculas. Forma de uso: cadena.upper() Ejemplo básico:

python

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

Ejemplo avanzado:

python

```
def gritar(texto):
    return f"¡¡{texto.upper()}!!"

print(gritar("atención")) # "¡¡ATENCIÓN!!"
```

2. lower(): Convierte todos los caracteres de la cadena a minúsculas. Forma de uso: cadena.lower() Ejemplo básico:

python

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

Ejemplo avanzado:

python

```
nombres = ["ANA", "Juan", "MARÍA", "pedro"]
nombres_normalizados = [nombre.lower() for nombre in nombres]
print(nombres_normalizados) # ["ana", "juan", "maría", "pedro"]
```

3. strip(): Elimina los espacios en blanco al principio y al final de la cadena. Forma de uso: cadena.strip() Ejemplo básico:

python

```
texto = "  Hola, mundo  "
print(texto.strip()) # "Hola, mundo"
```

Ejemplo avanzado:

python

```
def limpiar_datos(lista):  
    return [item.strip().lower() for item in lista if item.strip()]  
  
datos = [" Python ", " Java ", " ", "C++ "]  
print(limpiar_datos(datos)) # ["python", "java", "c++"]
```

4. `replace(old, new)`: Reemplaza todas las ocurrencias de una subcadena por otra. Forma de uso: `cadena.replace(viejo, nuevo)` Ejemplo básico:

python

```
texto = "Hola, mundo"  
print(texto.replace("mundo", "Python")) # "Hola, Python"
```

Ejemplo avanzado:

python

```
def censurar(texto, palabras_censuradas):  
    for palabra in palabras_censuradas:  
        texto = texto.replace(palabra, "*" * len(palabra))  
    return texto  
  
frase = "Este es un ejemplo de texto con palabras censurables"  
censuradas = ["ejemplo", "censurables"]  
print(censurar(frase, censuradas)) # "Este es un ***** de texto con palabras *****"
```

5. `split(sep)`: Divide la cadena en una lista de subcadenas. Forma de uso: `cadena.split(separador)` Ejemplo básico:

python

```
texto = "manzana,banana,cereza"  
print(texto.split(",")) # ["manzana", "banana", "cereza"]
```

Ejemplo avanzado:

python

```
def analizar_frase(frase):
    palabras = frase.lower().split()
    return {
        "num_palabras": len(palabras),
        "palabra_mas_larga": max(palabras, key=len),
        "palabras_unicas": len(set(palabras))
    }

resultado = analizar_frase("El rápido zorro marrón salta sobre el perro perezoso")
print(resultado)
# {'num_palabras': 9, 'palabra_mas_larga': 'perezoso', 'palabras_unicas': 9}
```

6. join(iterable): Une los elementos de un iterable en una sola cadena. Forma de uso: separador.join(iterable) Ejemplo básico:

python

```
frutas = ["manzana", "banana", "cereza"]
print(", ".join(frutas)) # "manzana, banana, cereza"
```

Ejemplo avanzado:

python

```
def generar_lista_html(items):
    elementos = "".join(f"<li>{item}</li>" for item in items)
    return f"<ul>{elementos}</ul>"

tareas = ["Comprar leche", "Pasear al perro", "Hacer ejercicio"]
print(generar_lista_html(tareas))
# "<ul><li>Comprar leche</li><li>Pasear al perro</li><li>Hacer ejercicio</li></ul>"
```

7. startswith(prefix): Verifica si la cadena comienza con el prefijo especificado. Forma de uso: cadena.startswith(prefijo) Ejemplo básico:

python

```
texto = "Hola, mundo"
```

```
print(texto.startswith("Hola")) # True
```

Ejemplo avanzado:

python

```
def categorizar_archivo(nombre):  
    if nombre.startswith(("IMG", "DCIM")):  
        return "imagen"  
    elif nombre.startswith("DOC"):  
        return "documento"  
    else:  
        return "otro"  
  
archivos = ["IMG001.jpg", "DOC123.pdf", "archivo.txt"]  
categorias = [categorizar_archivo(archivo) for archivo in archivos]  
print(categorias) # ['imagen', 'documento', 'otro']
```

8. `endswith(suffix)`: Verifica si la cadena termina con el sufijo especificado. Forma de uso: `cadena.endswith(sufijo)` Ejemplo básico:

python

```
texto = "Hola, mundo"  
print(texto.endswith("mundo")) # True
```

Ejemplo avanzado:

python

```
import os  
  
def obtener_archivos_por_extension(directorio, extensiones):  
    archivos = os.listdir(directorio)  
    return [archivo for archivo in archivos if archivo.lower().endswith(extensiones)]  
  
archivos_imagen = obtener_archivos_por_extension(".", (".jpg", ".png", ".gif"))  
print(archivos_imagen)
```

9. find(sub): Busca una subcadena y devuelve el índice de su primera aparición. Forma de uso: cadena.find(subcadena) Ejemplo básico:

python

```
texto = "Hola, mundo"
print(texto.find("mundo")) # 6
```

Ejemplo avanzado:

python

```
def extraer_entre_etiquetas(texto, etiqueta):
    inicio = texto.find(f"<{etiqueta}>") + len(etiqueta) + 2
    fin = texto.find(f"</{etiqueta}>")
    return texto[inicio:fin] if inicio > -1 and fin > -1 else None

html = "<body><h1>Título</h1><p>Párrafo de ejemplo</p></body>"
print(extraer_entre_etiquetas(html, "h1")) # "Título"
```

10. count(sub): Cuenta el número de veces que aparece una subcadena. Forma de uso: cadena.count(subcadena) Ejemplo básico:

python

```
texto = "Hola, hola, hola mundo"
print(texto.count("hola")) # 2
```

Ejemplo avanzado:

python

```
def analizar_frecuencia_palabras(texto):
    palabras = texto.lower().split()
    return {palabra: palabras.count(palabra) for palabra in set(palabras)}

frase = "El perro persigue al gato y el gato persigue al ratón"
print(analizar_frecuencia_palabras(frase))
# {'el': 2, 'perro': 1, 'persigue': 2, 'al': 2, 'gato': 2, 'y': 1, 'ratón': 1}
```

11. `isalpha()`: Verifica si todos los caracteres de la cadena son alfabéticos. Forma de uso: `cadena.isalpha()` Ejemplo básico:

python

```
texto = "Python"
print(texto.isalpha()) # True
```

Ejemplo avanzado:

python

```
def es_pangrama(frase):
    alfabeto = set('abcdefghijklmnopqrstuvwxyz')
    return all(letra in frase.lower() for letra in alfabeto if letra.isalpha())

pangrama = "El veloz murciélago hindú comía feliz cardillo y kiwi"
print(es_pangrama(pangrama)) # True
```

12. `rstrip()`: Elimina los espacios en blanco (o caracteres especificados) al final de la cadena. Forma de uso: `cadena.rstrip()` Ejemplo básico:

python

```
texto = "Hola, mundo   "
print(texto.rstrip()) # "Hola, mundo"
```

Ejemplo avanzado:

python

```
def limpiar_puntuacion(texto):
    return texto.rstrip('.,:;!?'')

frases = ["Hola, mundo.", "¿Cómo estás?", "Bien, gracias!"]
limpio = [limpiar_puntuacion(frase) for frase in frases]
print(limpio) # ['Hola, mundo', '¿Cómo estás', 'Bien, gracias']
```

13. `isdigit()`: Verifica si todos los caracteres de la cadena son dígitos. Forma de uso: `cadena.isdigit()` Ejemplo básico:

python

```
numero = "12345"
print(numero.isdigit()) # True
```

Ejemplo avanzado:

python

```
def extraer_numeros(texto):
    palabras = texto.split()
    return [int(palabra) for palabra in palabras if palabra.isdigit()]

frase = "Tengo 3 manzanas y 2 peras"
print(extraer_numeros(frase)) # [3, 2]
```

14. `capitalize()`: Convierte el primer carácter a mayúscula y el resto a minúsculas. Forma de uso: `cadena.capitalize()` Ejemplo básico:

python

```
texto = "hola, mundo"
print(texto.capitalize()) # "Hola, mundo"
```

Ejemplo avanzado:

python

```
def formatear_nombres(nombres):
    return ' '.join(nombre.capitalize() for nombre in nombres.split())

nombres = "juan CARLOS lópez"
print(formatear_nombres(nombres)) # "Juan Carlos López"
```

15. `title()`: Convierte la primera letra de cada palabra a mayúscula. Forma de uso: `cadena.title()` Ejemplo básico:

python

```
texto = "bienvenido a python"
print(texto.title()) # "Bienvenido A Python"
```

Ejemplo avanzado:

python

```
import re

def titulo_personalizado(texto):
    excepciones = {'de', 'del', 'la', 'las', 'los', 'y', 'e', 'o', 'u'}
    palabras = texto.lower().split()
    return ' '.join(palabra if palabra in excepciones else palabra.capitalize()
                    for palabra in palabras)

frase = "la historia de las ciencias y las artes"
print(titulo_personalizado(frase)) # "La Historia de las Ciencias y las Artes"
```

16. `zfill(width)`: Rellena la cadena con ceros a la izquierda hasta alcanzar la longitud especificada. Forma de uso: `cadena.zfill(ancho)` Ejemplo básico:

python

```
numero = "42"
print(numero.zfill(5)) # "00042"
```

Ejemplo avanzado:

python

```
def formatear_codigo(codigo, longitud=6):
    return f"PROD-{codigo.zfill(longitud)}"

codigos = ["1", "42", "1337"]
formateados = [formatear_codigo(codigo) for codigo in codigos]
print(formateados) # ['PROD-000001', 'PROD-000042', 'PROD-001337']
```

17. `partition(sep)`: Divide la cadena en tres partes basadas en el separador. Forma de uso: `cadena.partition(separador)` Ejemplo básico:

python


```
email = "usuario@ejemplo.com"
print(email.partition("@")) # ('usuario', '@', 'ejemplo.com')
```

Ejemplo avanzado:

python

```
def parse_nombre_archivo(nombre):
    nombre, separador, extension = nombre.partition('.')
    return {
        "nombre": nombre,
        "extension": extension if extension else None
    }

archivos = ["documento.txt", "imagen.jpg", "archivo"]
parsed = [parse_nombre_archivo(archivo) for archivo in archivos]
print(parsed)
# [{'nombre': 'documento', 'extension': 'txt'},
#  {'nombre': 'imagen', 'extension': 'jpg'},
#  {'nombre': 'archivo', 'extension': None}]
```