# Documentación del Script scraper.py

## Introducción

Este script en Python realiza scraping de la página web de Suecia Barbershop (`https://sueciabarbershop.cl/`) para extraer los títulos (`<h1>`) presentes en la página y guarda estos títulos en un archivo CSV.

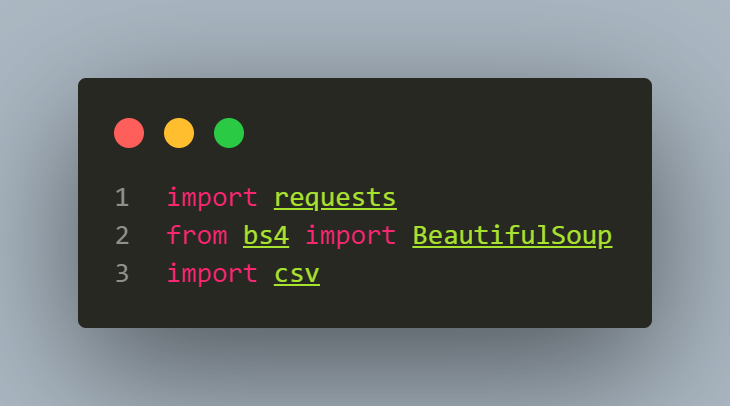
## Requisitos Previos

Python 3.x  
Bibliotecas: `requests`, `BeautifulSoup` (de `bs4`), `csv`  
  
Puedes instalar las bibliotecas necesarias utilizando pip:

pip install requests beautifulsoup4

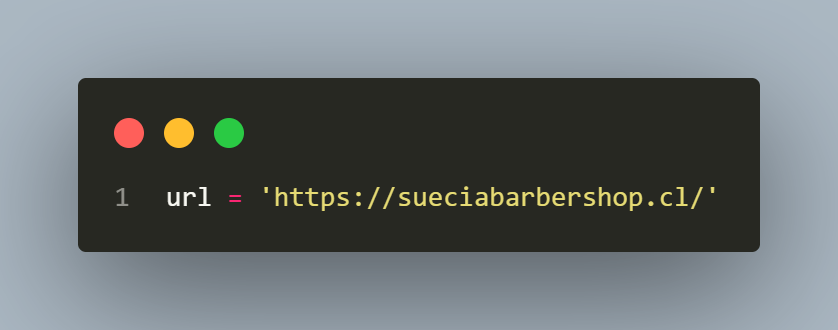
## Descripción del Código

### 1. Importación de Bibliotecas



- `requests`: Para realizar solicitudes HTTP.  
- `BeautifulSoup` de `bs4`: Para parsear el HTML de la página web.  
- `csv`: Para manejar la escritura de datos en archivos CSV.

### 2. Definición de la URL



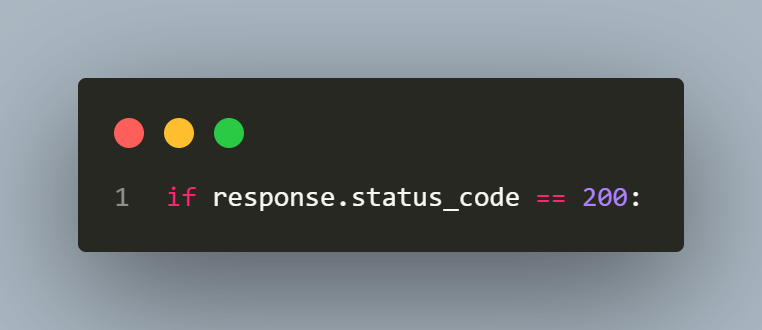
- `url`: Contiene la dirección de la página web de la que se desea extraer información.

### 3. Realizar la Solicitud HTTP



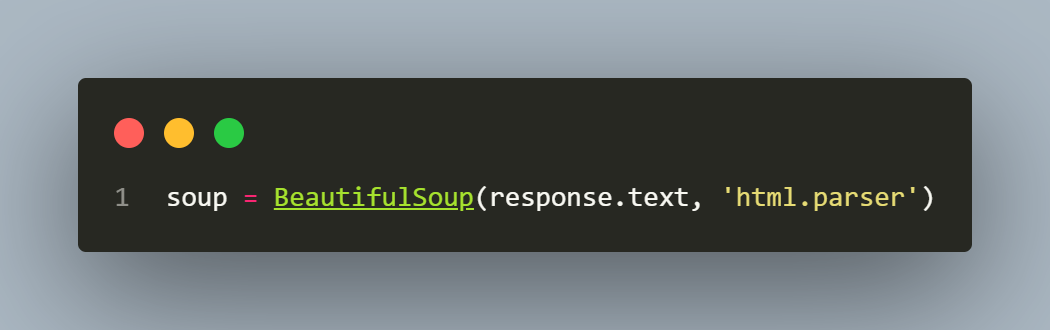
- `response`: Guarda la respuesta de la solicitud GET realizada a la URL.

### 4. Verificación del Estado de la Solicitud



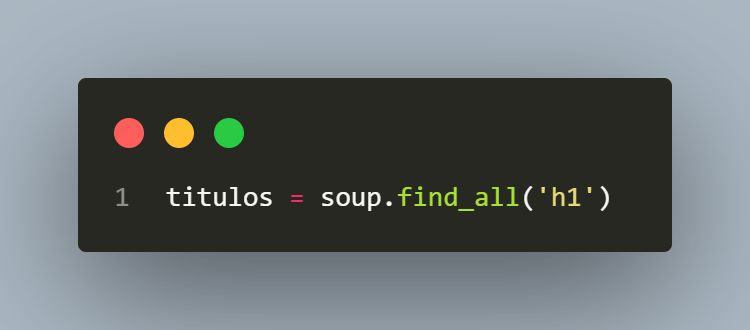
- Verifica si la solicitud fue exitosa (código de estado 200).

### 5. Parseo del HTML



- `soup`: Contiene el HTML parseado de la página web.

### 6. Extracción de Datos Específicos



- `titulos`: Lista de todos los elementos `<h1>` encontrados en la página.

### 7. Escritura de Datos en CSV



- Abre (o crea) un archivo llamado `datos.csv` para escritura.  
- `fieldnames`: Define el nombre del campo para el CSV.  
- `writer`: Objeto que maneja la escritura en el archivo CSV.  
- `writer.writeheader()`: Escribe el encabezado del CSV.  
- Para cada título en la lista `titulos`, escribe una fila en el CSV con el texto del título.

### 8. Manejo de Errores

- Si la solicitud no es exitosa, imprime un mensaje de error con el código de estado HTTP.

## Ejecución del Script

Para ejecutar el script, simplemente corre el archivo `scraper.py` en tu entorno de desarrollo o línea de comandos:  
  
python scraper.py

El script generará un archivo `datos.csv` en el mismo directorio, que contendrá los títulos (`<h1>`) extraídos de la página web especificada.

## Consideraciones

- Asegúrate de tener permiso para realizar scraping en la página web.  
- El script está diseñado para extraer solo elementos `<h1>`. Puedes modificarlo para extraer otros elementos ajustando el método `soup.find\_all()`.  
- Manejo de posibles excepciones adicionales puede ser necesario para robustecer el script en entornos de producción.