

Hoja de trucos: API y recopilación de datos

| Paquete/Método | Descripción | Ejemplo de código |
|-------------------------------------|---|---|
| Accediendo al atributo del elemento | Accede al valor de un atributo específico de un elemento HTML. | Sintaxis: <code>atributo = elemento[(atributo)]</code> Ejemplo: <code>href = link_element[(href)]</code> |
| BeautifulSoup() | Analiza el contenido HTML de una página web utilizando BeautifulSoup. El tipo de analizador puede variar según el proyecto. | Sintaxis: <code>soup = BeautifulSoup(html, (html.parser))</code> Ejemplo: <code>html = (https://api.example.com/data) soup = BeautifulSoup(html, (html.parser))</code> |
| delete() | Envía una solicitud DELETE para eliminar datos o un recurso del servidor. Las solicitudes DELETE eliminan un recurso específico en el servidor. | Sintaxis: <code>response = requests.delete(url)</code> Ejemplo: <code>response = requests.delete((https://api.example.com/delete))</code> |
| find() | Encuentra el primer elemento HTML que | Sintaxis: <code>element = soup.find(tag, attrs)</code> |

| | | |
|----------------|--|--|
| | coincide con la etiqueta y atributos especificados. | Ejemplo: <pre>first_link = soup.find((a), {{class}): (link)})</pre> |
| find_all() | Encuentra todos los elementos HTML que coinciden con la etiqueta y atributos especificados. | Sintaxis: <pre>elements = soup.find_all(tag, attrs)</pre> Ejemplo: <pre>all_links = soup.find_all((a), {{class}): (link)})</td></pre> |
| findChildren() | Encuentra todos los elementos hijos de un elemento HTML. | Sintaxis: <pre>children = element.findChildren()</pre> Ejemplo: <pre>child_elements = parent_div.findChildren()</pre> |
| get() | Realiza una solicitud GET para recuperar datos de una URL especificada. Las solicitudes GET se utilizan típicamente para leer datos de una API. La variable de respuesta | Sintaxis: <pre>response = requests.get(url)</pre> Ejemplo: |

| | | |
|----------------------|---|--|
| | contendrá la respuesta del servidor, que puedes procesar más adelante. | <pre>response = requests.get((https://api.example.com/data))</pre> |
| Headers | Incluye encabezados personalizados en la solicitud. Los encabezados pueden proporcionar información adicional al servidor, como tokens de autenticación o tipos de contenido. | <p>Sintaxis:</p> <pre>headers = {(HeaderName): (Value)}</pre> <p>Ejemplo:</p> <pre>base_url = (https://api.example.com/data) headers = {(Authorization): (Bearer YOUR_TOKEN)} response = requests.get(base_url, headers=headers)</pre> |
| Importar bibliotecas | Importa las bibliotecas de Python necesarias para el web scraping. | <p>Sintaxis:</p> <pre>from bs4 import BeautifulSoup</pre> |
| json() | Analiza datos JSON de la respuesta. Esto extrae y trabaja con los datos devueltos por la API. El método response.json() convierte la respuesta JSON en una estructura de datos de Python (generalmente un diccionario o lista). | <p>Sintaxis:</p> <pre>data = response.json()</pre> <p>Ejemplo:</p> <pre>response = requests.get((https://api.example.com/data)) data = response.json()</pre> |
| next_sibling() | Encuentra el siguiente elemento hermano en el DOM. | <p>Sintaxis:</p> <pre>sibling = element.find_next_sibling()</pre> |

| | | |
|--------|---|--|
| | | <p>Ejemplo:</p> <pre>next_sibling = current_element.find_next_sibling()</pre> |
| parent | <p>Accede al elemento padre en el Modelo de Objetos del Documento (DOM).</p> | <p>Sintaxis:</p> <pre>parent = element.parent</pre> <p>Ejemplo:</p> <pre>parent_div = paragraph.parent</pre> |
| post() | <p>Envía una solicitud POST a una URL especificada con datos. Crea o actualiza solicitudes POST utilizando recursos en el servidor. El parámetro de datos contiene los datos que se enviarán al servidor, a menudo en formato JSON.</p> | <p>Sintaxis:</p> <pre>response = requests.post(url, data)</pre> <p>Ejemplo:</p> <pre>response = requests.post((https://api.example.com/submit), data={(key): (value)})</pre> |

| | | |
|------------------------|--|---|
| put() | <p>Envía una solicitud PUT para actualizar datos en el servidor. Las solicitudes PUT se utilizan para actualizar un recurso existente en el servidor con los datos proporcionados en el parámetro de datos, típicamente en formato JSON.</p> | <p>Sintaxis:</p> <pre>response = requests.put(url, data)</pre> <p>Ejemplo:</p> <pre>response = requests.put('https://api.example.com/update', data={key: value})</pre> |
| Parámetros de consulta | <p>Envía parámetros de consulta en la URL para filtrar o personalizar la solicitud. Los parámetros de consulta especifican condiciones o límites para los datos solicitados.</p> | <p>Sintaxis:</p> <pre>params = {param_name: value}</pre> <p>Ejemplo:</p> <pre>base_url = 'https://api.example.com/data' params = {"page": 1, "per_page": 10} response = requests.get(base_url, params=params)</pre> |
| select() | <p>Selecciona elementos HTML del HTML analizado utilizando un selector CSS.</p> | <p>Sintaxis:</p> <pre>element = soup.select(selector)</pre> <p>Ejemplo:</p> <pre>titles = soup.select('h1')</pre> |
| status_code | <p>Verifica el código de estado HTTP de la respuesta. El código de estado HTTP indica el resultado de la</p> | <p>Sintaxis:</p> <pre>response.status_code</pre> |

| | | |
|-------|--|---|
| | <p>solicitud (éxito, error, redirección). El código de estado HTTP se puede usar para el manejo de errores y la toma de decisiones en tu código.</p> | <p>Ejemplo:</p> <pre>url = "https://api.example.com/data" response = requests.get(url) status_code = response.status_code</pre> |
| | <p>etiquetas para <code>find()</code> y <code>find_all()</code></p> <p>Especifica cualquier etiqueta HTML válida como parámetro de etiqueta para buscar elementos de ese tipo. Aquí hay algunas etiquetas HTML comunes que puedes usar con el parámetro de etiqueta.</p> | <p>Ejemplo de etiqueta:</p> <ul style="list-style-type: none"> - <code>(a)</code>: Encontrar etiquetas de ancla () . - <code>(p)</code>: Encontrar etiquetas de párrafo ((p)). - <code>(h1), (h2), (h3), (h4), (h5), (h6)</code>: Encontrar etiquetas de encabezado del nivel 1 al 6 ((h1),n (h2)). - <code>(table)</code>: Encontrar etiquetas de tabla () . - <code>(tr)</code>: Encontrar etiquetas de fila de tabla () . - <code>(td)</code>: Encontrar etiquetas de celda de tabla ((td)). - <code>(th)</code>: Encontrar etiquetas de celda de encabezado de tabla ((td)). - <code>(img)</code>: Encontrar etiquetas de imagen ((img)). - <code>(form)</code>: Encontrar etiquetas de formulario ((form)). - <code>(button)</code>: Encontrar etiquetas de botón ((button)). |
| texto | <p>Recupera el contenido de texto de un elemento HTML.</p> | <p>Sintaxis:</p> <pre>text = element.text</pre> <p>Ejemplo:</p> <pre>title_text = title_element.text</pre> |



Skills Network