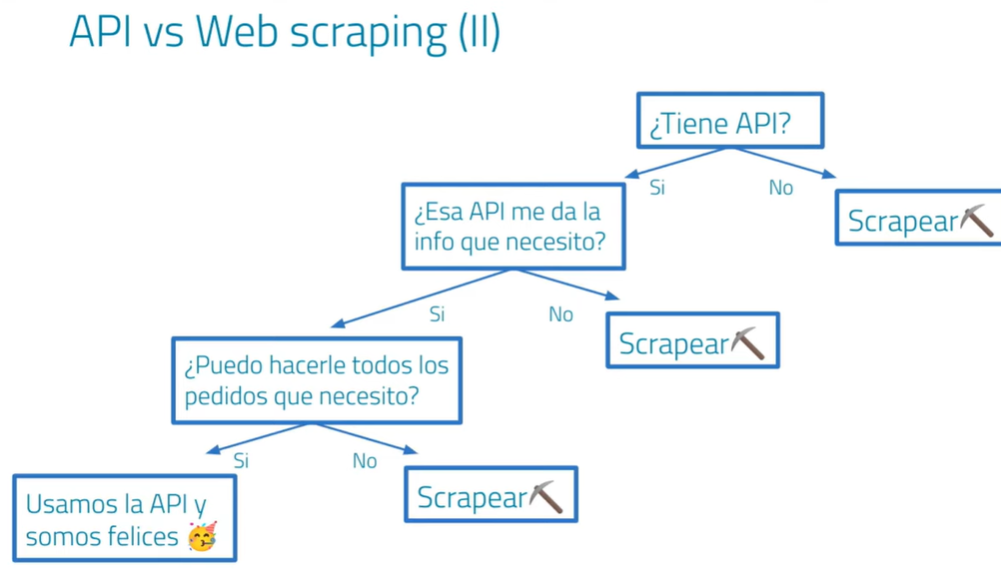
FUENTE: <https://www.youtube.com/watch?v=yKi9-BfbfzQ&t=472s&ab_channel=freeCodeCampEspa%C3%B1ol>

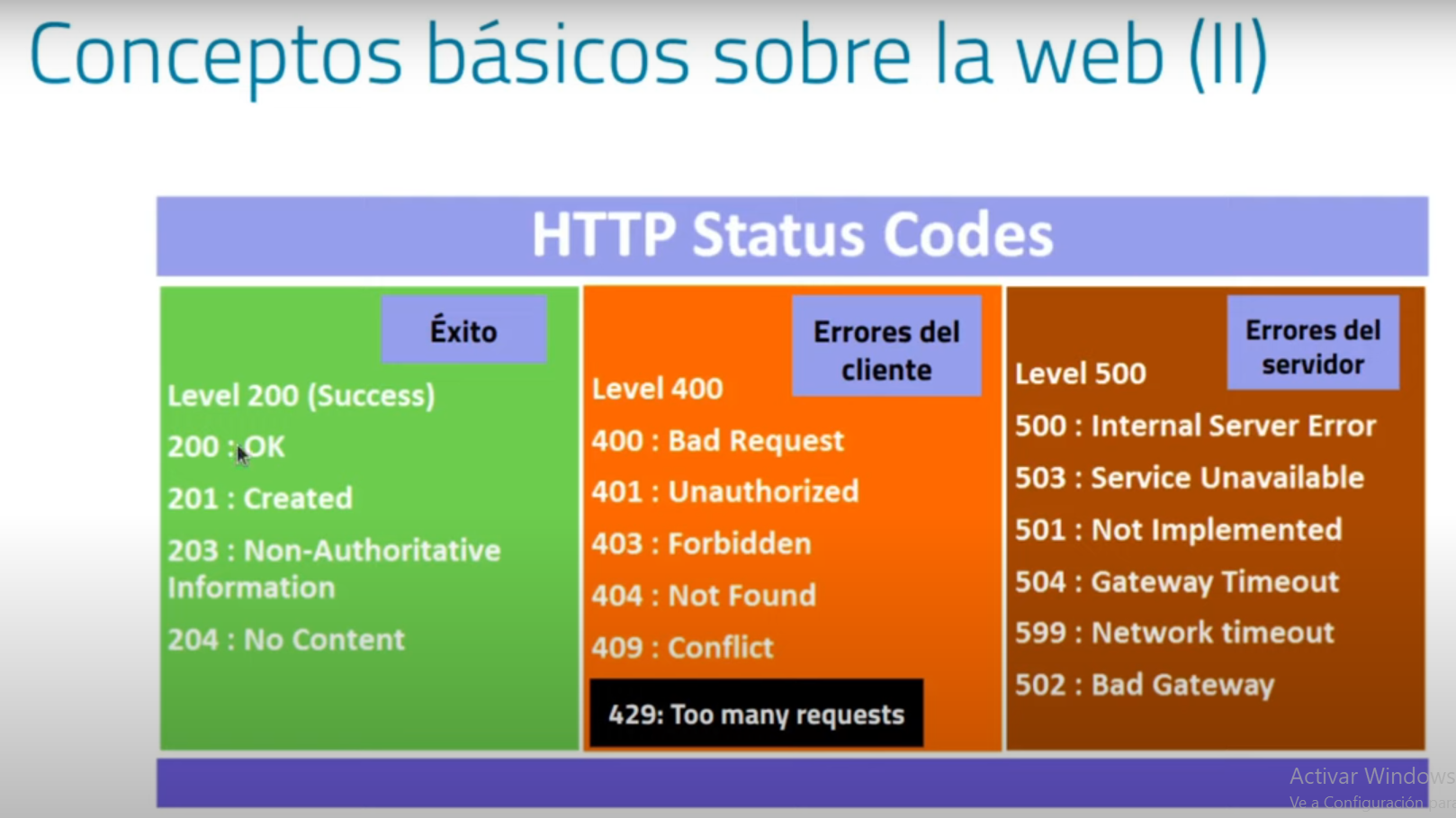
Cuando hablamos de **web scraping** nos referimos a buscar información concreta en una web ya sean imágenes, audios… Es **información sintetizada**. De lo contrario se llama **web crawling**, que sería tener toda la información de un sitio.

Legalidad: si es información libre no debería de haber problema.

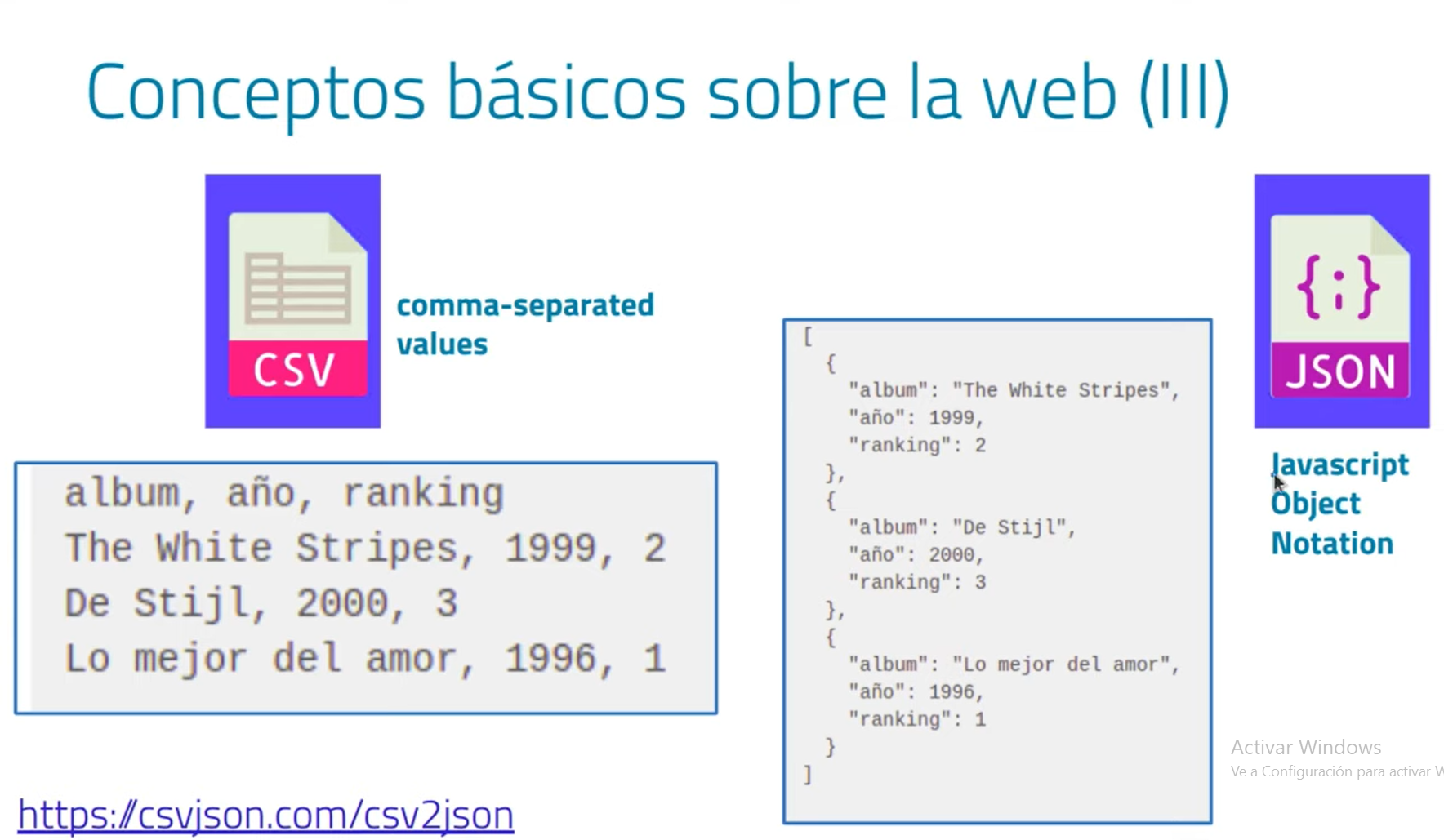
Servidores: si la página donde pedimos información tiene servidores pequeños y nosotros pedimos mucha información podríamos afectar a su buen funcionamiento.







Hay muchos más códigos que hay que aprender.



XML ha sido cada vez menos usado, ganando terreno JSON:





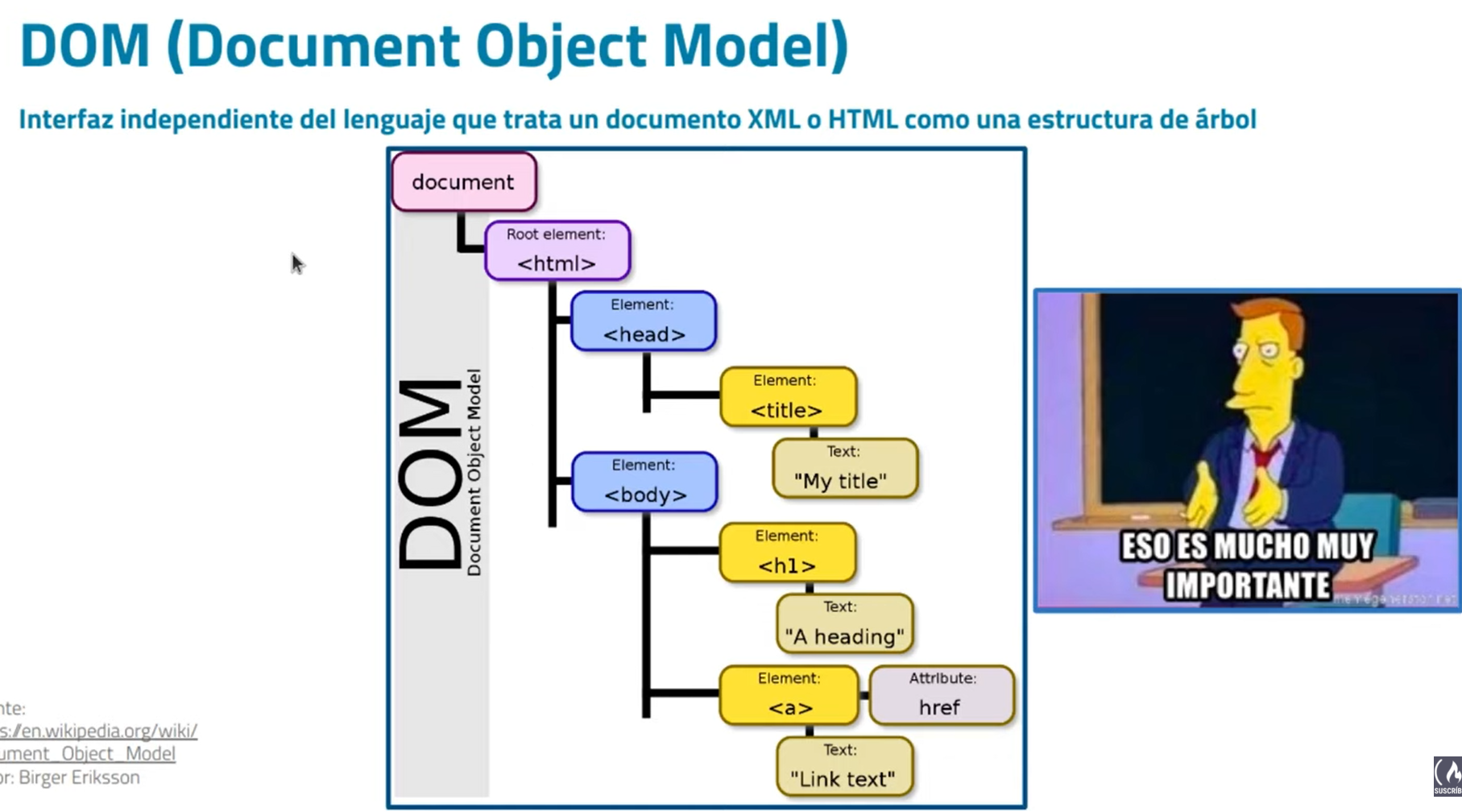
Ejemplo de HTML y CSS:

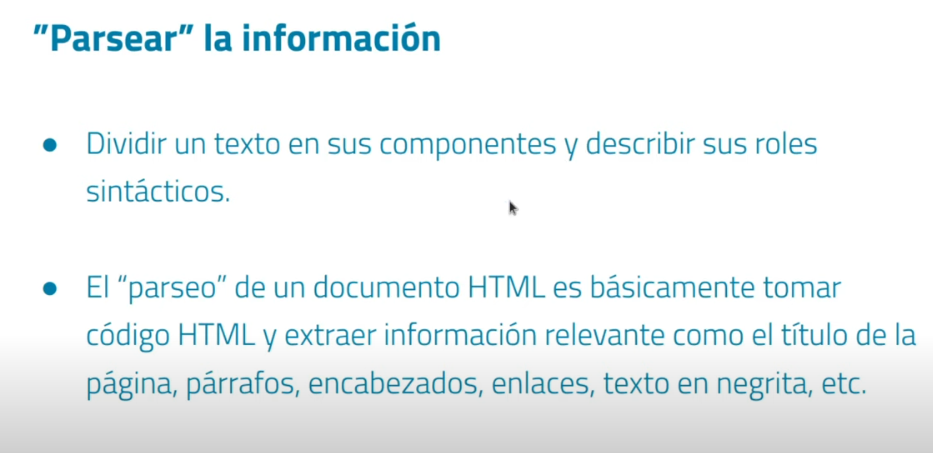


Es importante tener ciertas nociones de todos estos lenguajes para poder hacer un buen “Scrapeo”.

DOM

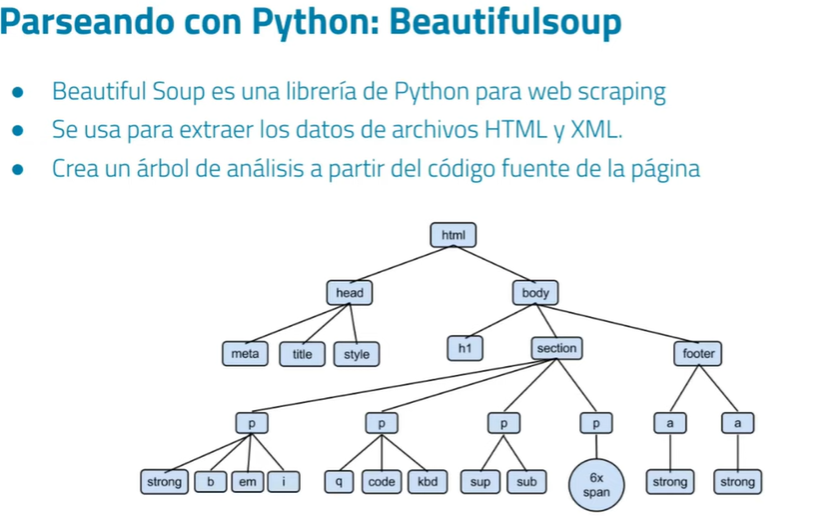
Esto es bastante importante para escrapear puesto que cada vez que escrapeemos una página tendremos que lidiar con estructuras similares.





El proceso de un escrapeo de forma resumida sería algo así:





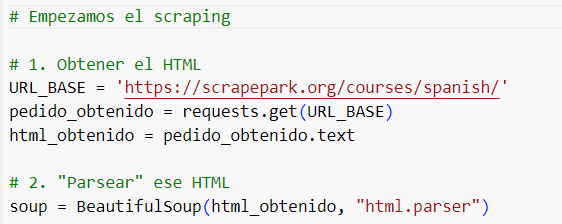
A partir de ahora, vamos a ayudarnos en el aprendizaje desde el DRIVE, donde tenemos códigos y ejemplos.

[https://colab.research.google.com/drive/1nNst9ru2\_M11SR44fvKoWK4Vj2BCU5Ba#scrollTo=HFbPvfyu3yaS](https://colab.research.google.com/drive/1nNst9ru2_M11SR44fvKoWK4Vj2BCU5Ba%23scrollTo=HFbPvfyu3yaS)

Las API’s suelen ser la forma más sencilla de escrapear, pero muchas webs no tienen API’s que nos permitan ver su información. Para ello podemos usar la librería BeautifulSoup.

La mejor forma de entender a fondo una librería siempre es con su documentación. Así que dejo aquí su enlace: [https://beautiful-soup-4.readthedocs.io/en/latest/](https://www.google.com/url?q=https%3A%2F%2Fbeautiful-soup-4.readthedocs.io%2Fen%2Flatest%2F)

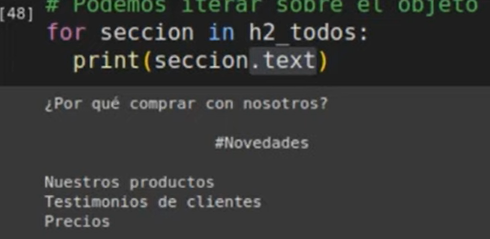
En el primer ejemplo para empezar a escrapear:



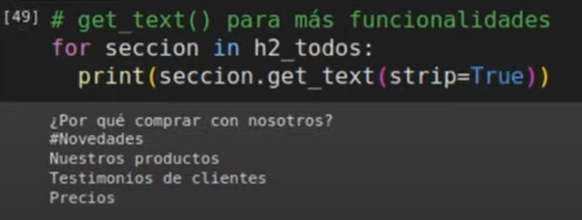
En el paso uno simplemente obtenemos el HTML de esa página en formato text(sin etiquetas). En el segundo paso queremos parsear ese HTML obtenido y usamos el “motor” html.parser. Hay otros “motores” y cada uno te puede servir en un determinado momento. Por norma general se usa este.

Con el atributo .text te pasa el texto sin modificar, este muchas veces te puede venir alterado por espacios de más o cosas así. Entonces podemos usar el método **get\_text**(). Con este método podemos sacar el texto del HTML(sin etiquetas) y modificarlo para que el texto sea mas legible. Por ejemplo:

Con text:



Con Get\_text():



Con Strip=True le decimos que nos limpie los espacios en blanco.

**Método Find**(): ver DRIVE.

**Metodo Find\_all**(): ver DRIVE.

**Tablas**: Un problema que suele ser común al scrapear es que hay tablas o objetos que no aparecen en el HTML. Esto es porque hay un tipo de etiqueta en HTML llamada IFRAME, que lo que hace es crear como una nueva página dentro de la página.

Si vamos a la tabla de precios de <https://scrapepark.org/spanish/> y miramos su código fuente podemos ver como nos indica un enlace.

