**Scraping utilizando Python**

**y el framework scrapy.**

Este trabajo trata de una araña web realizada con Python2.7 y el framework Scrapy, con el objetivo de sacar información de la versión digital del periódico **El país** para luego tratarla.

Lenguajes utilizados: Python2.7.14 y XPath 2.0.2 (de apoyo).

Framework: Srapy 1.5

Editor de texto: Sublime Text Build 3143

Extensión de ayuda: XPath helper (en Chrome)

**Manual de desarrollo**

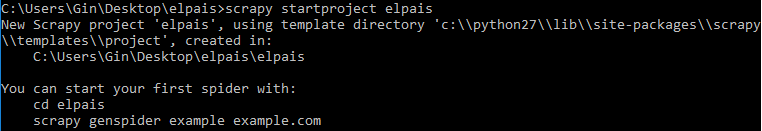
1. **Instalación**

Descargamos ‘python-2.7.14’ de [www.python.org/downloads/](http://www.python.org/downloads/) e instalar seleccionando:

Con la herramienta Pip, instalamos Scrapy:

1. **Creación de la araña**

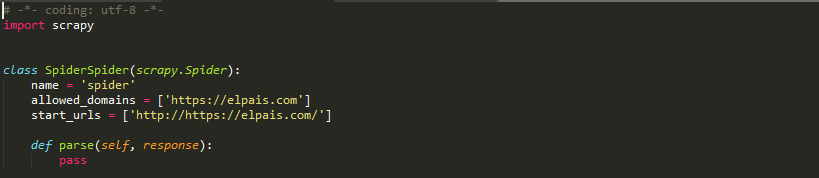
Vamos a la carpeta donde vayamos a tener nuestra araña y generamos el proyecto con Scrapy:



Vamos a la carpeta ‘Spiders’ y generamos nuestra primera araña



Tras crear la araña los principales archivos con los que trabajaremos son: spider.py y settings.py. Nos queda una clase, nuestra araña, con la siguiente estructura:



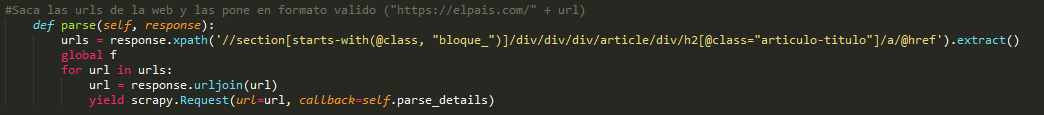
Name: El nombre con el que llamamos a nuestra araña.

Allowed\_domains: Dominios admitidos.

Start\_urls: Url por la que empieza a recorrer la web.

Para hacer una araña que recorra la página de El país y saque sus artículos de portada, seguimos los siguientes pasos:

Creamos un bucle que recoja las urls de los artículos, ignorando la publicidad.

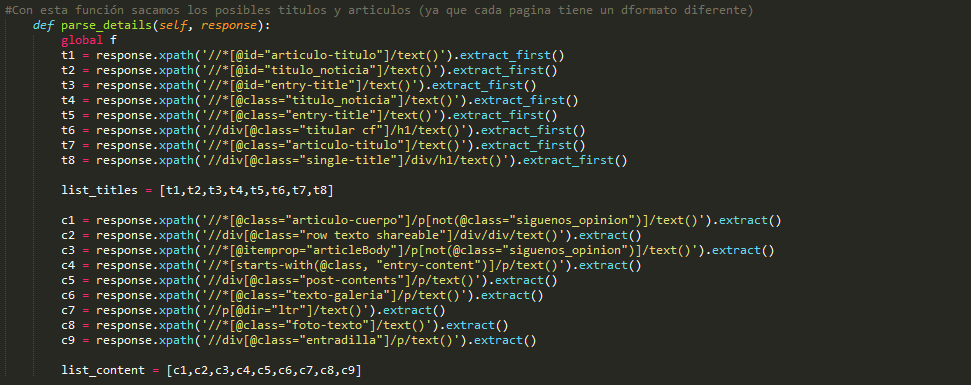


En urls sacamos el XPath que selecciona todas las urls de los artículos y las recorremos.

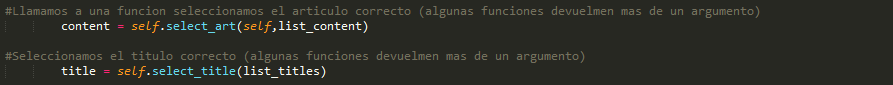
Urljoin: Nos concatena <https://elpais.com/> con la url que hemos sacado (si es necesario)

Yield: Aquí realizamos una petición a la url que toca y llamamos al método “parse\_details”.

Creamos un segundo método “parse\_details”, que va a iniciarse con cada Url que encontremos y recogerá el título y el artículo en sí.



Debido a que el país redirige a diferentes webs, no tenemos un formato común para sacar el título ni el artículo, por lo que necesitamos probar con los XPath de todas las webs (se verá más abajo).



Unimos los resultados de las <p> de artículos (generalmente).

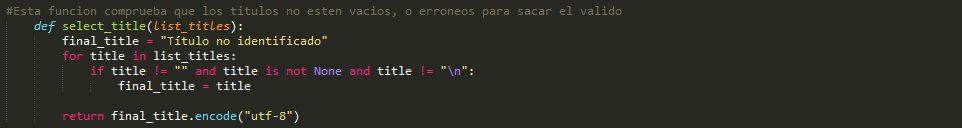


Escribimos título más artículo, separándolos por “|”.

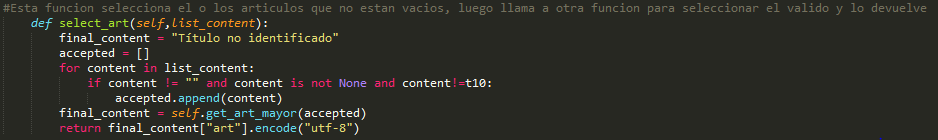


Esta función coge como argumento los resultados de todos los XPaths y los mete a una lista. De esta lista, sacamos el que no es un String vacío, un salto de línea o igual a nulo.

El elegido lo codificamos en utf-8 para que nos permita utilizar los caracteres especiales del lenguaje español.

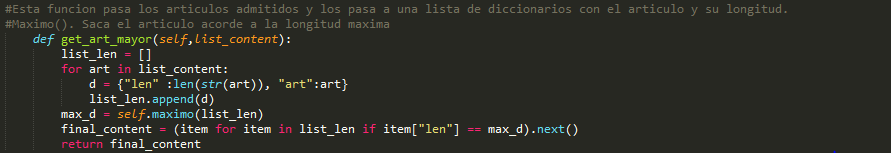


Con los artículos seguimos el mismo proceso, pero al ser más extensos, algunos XPath devuelven más de una solución. Así que, lo que nos devuelve es una lista de posibles artículos que, mediante otras funciones, nos devuelve el más extenso. El elegido lo codificamos en utf-8 para que nos permita utilizar los caracteres especiales del lenguaje español.

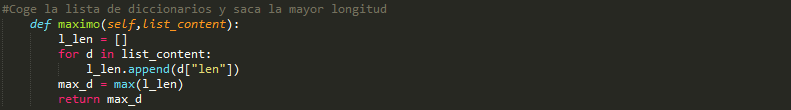


El primero de ellos recibe la lista de posibles y la pasa a una lista de diccionarios donde, cada uno de ellos, está compuesto por la longitud del artículo y el artículo en sí.

Tras la función máximo(), conseguimos la mayor longitud de la lista. Recorremos la lista y devolvemos el artículo correspondiente.



Aquí, recibimos la lista de diccionarios de artículos posibles. Metemos en otra lista la longitud de todos y, devolvemos la que sea mayor, ya que los artículos no válidos no llegan a las 5 líneas.



1. **Consulta de información**

Teniendo los datos, podemos pasar a obtener información que nos sea de utilidad.

Para poner un ejemplo contaremos el número de iteraciones de una palabra en los artículos, para ver la relevancia de la misma.



Esta función abre el resultado obtenido por nuestra araña y lo guarda en una lista. Separamos esta lista por espacios para extraer cada una de las palabras y poder ver las repeticiones de la palabra elegida.

Debido a que separamos por espacios, los caracteres especiales se solapan con las palabras, pero sin interferir en el tratamiento de las mismas. Por ello lo que hacemos es ver si la palabra que nosotros hemos seleccionado está contenida en cada uno de los elementos, palabras, de la lista. Para hacer un tratamiento más fino, convertimos a mayúsculas tanto la palabra buscada como las contenidas en la lista, una vez hecho esto devolvemos el número de repeticiones.