

Máster en Internet de las cosas

Web Scraping

Rafael Caballero Roldán

*Departamento de Sistemas
Informáticos y Computación*

UCM

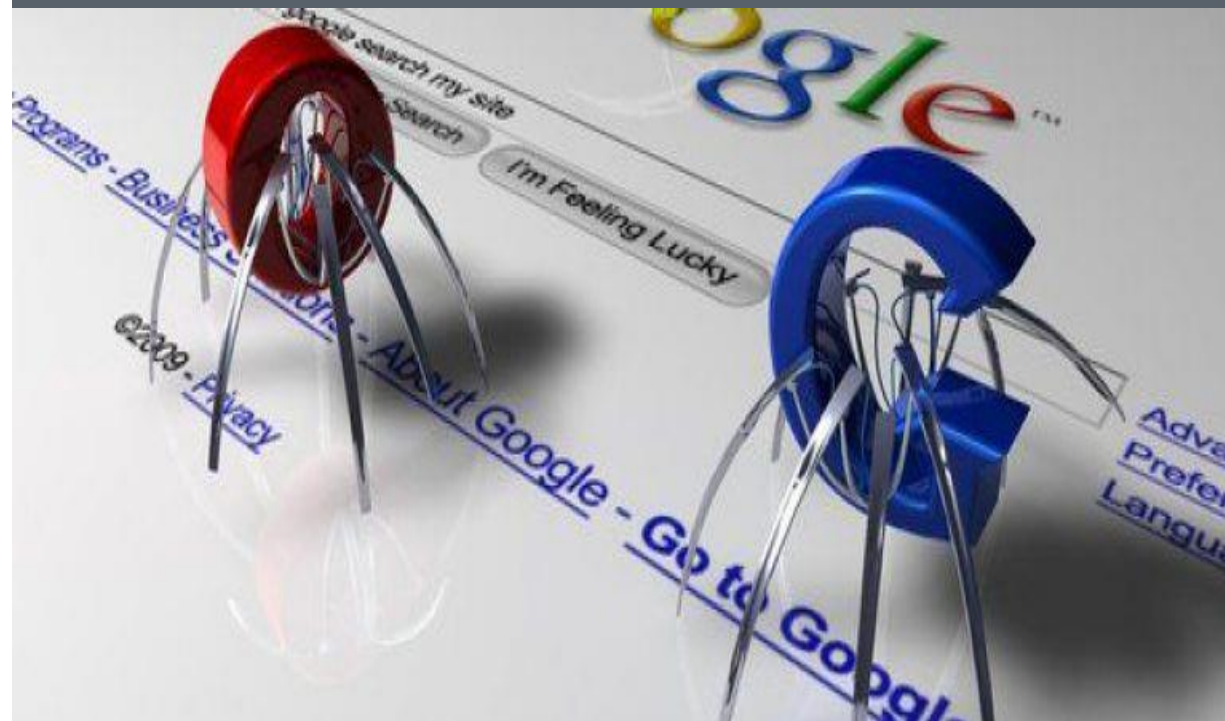
Web Scraping

Introducción

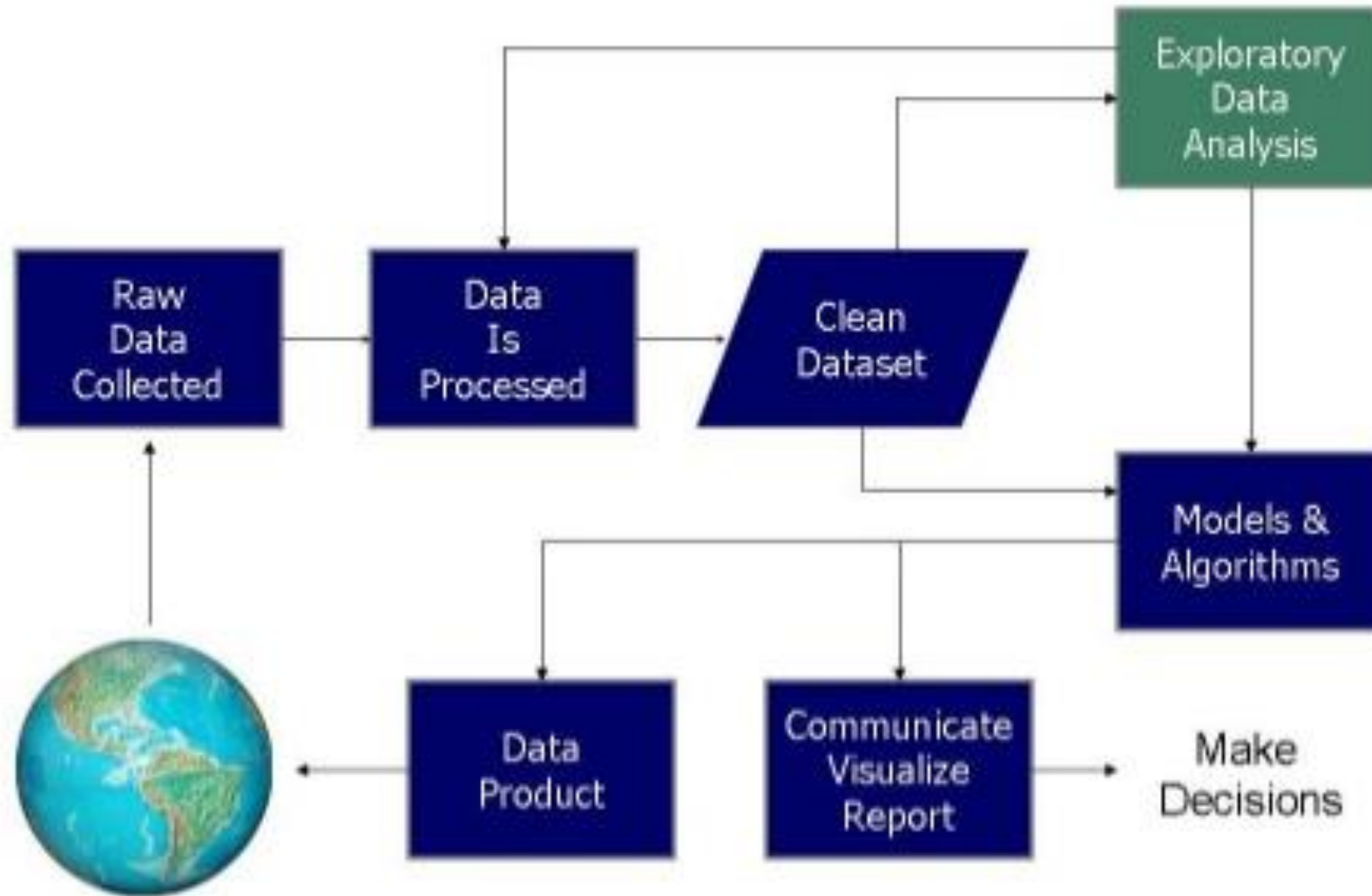
XPath

Selenium

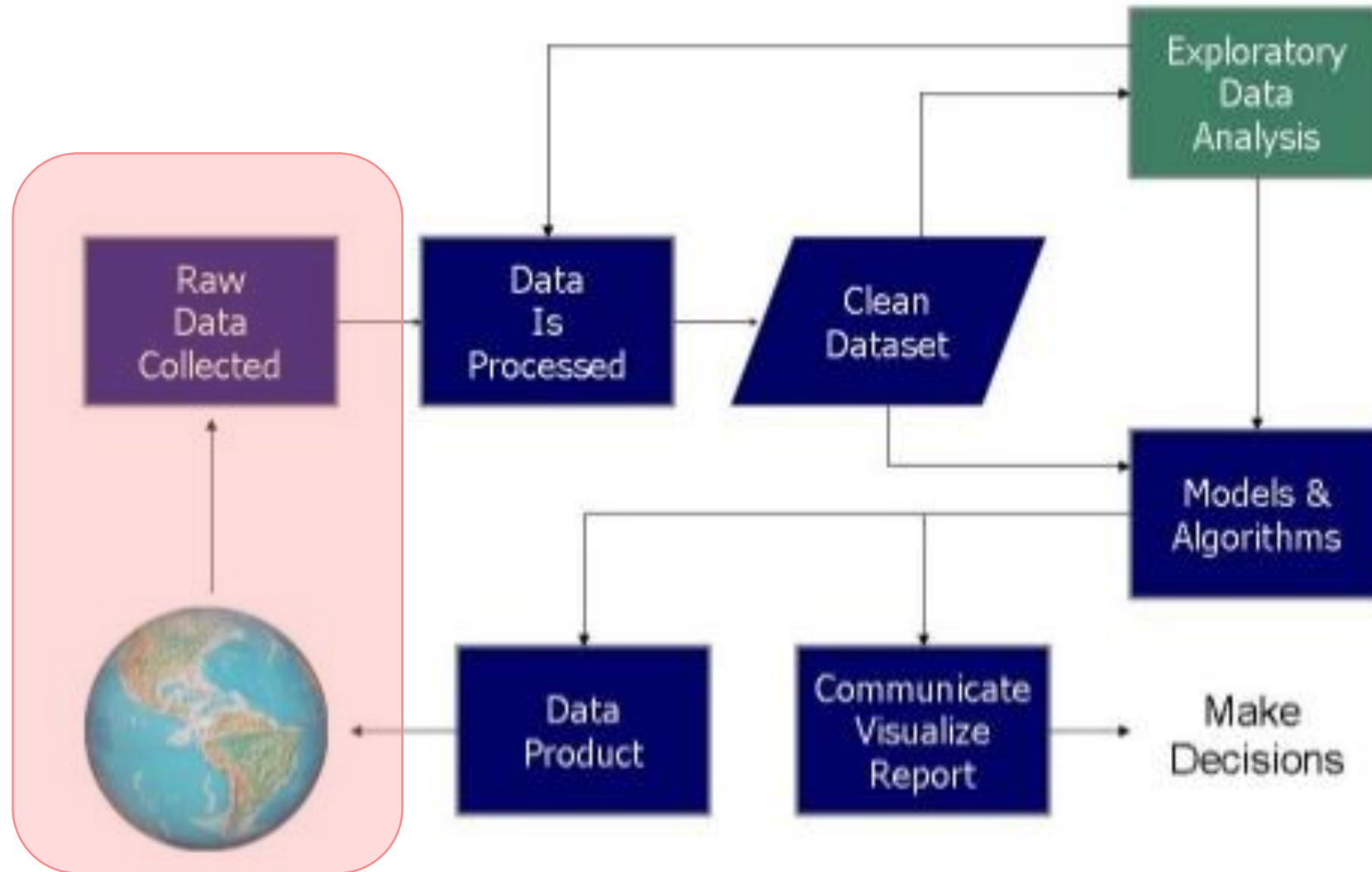
Ejemplo



Ciclo de análisis de datos



Ciclo de análisis de datos



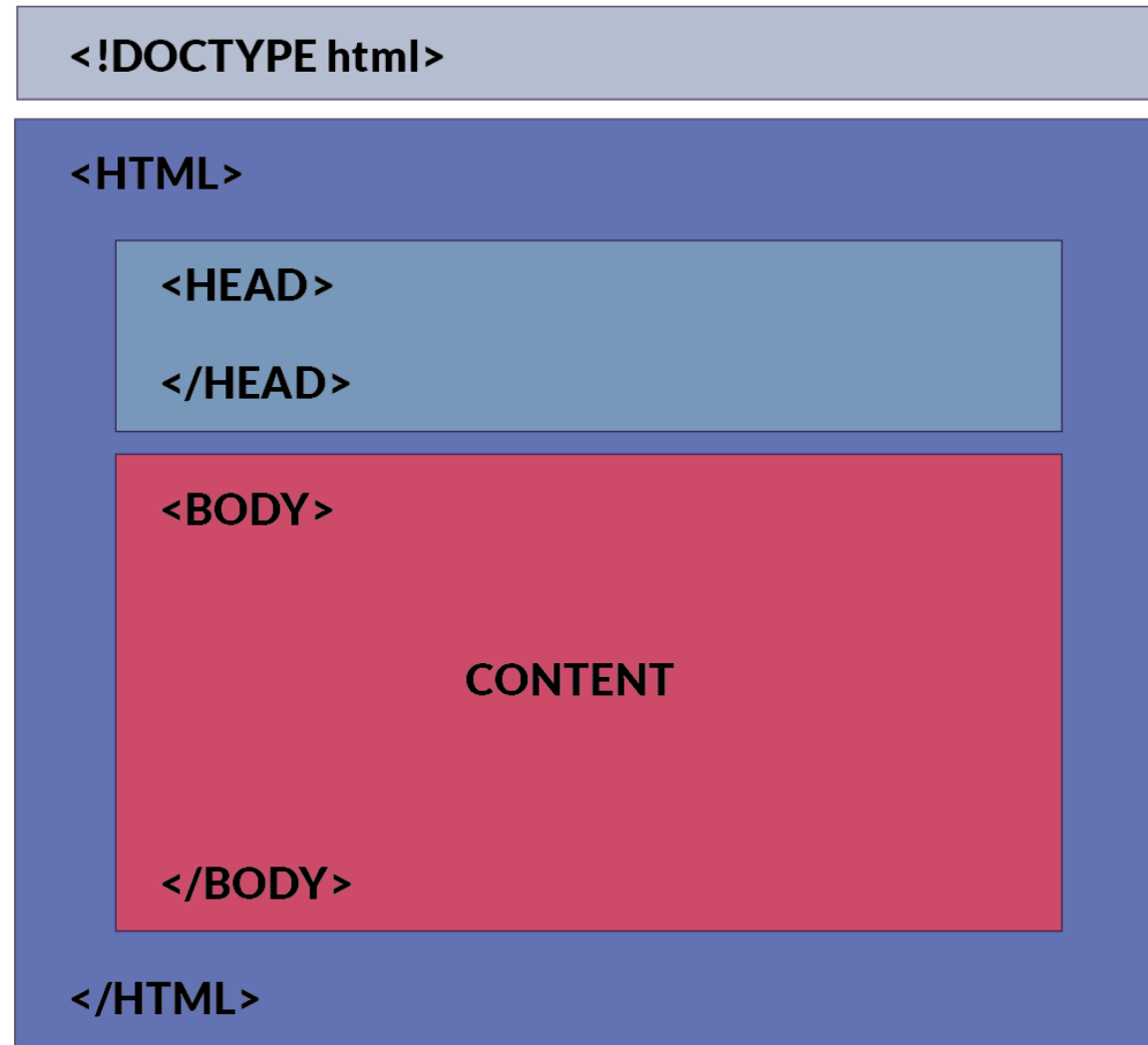
Web scraping

- Es una fuente de datos de gran utilidad
- Muy habitual en IoT: páginas web que muestran datos obtenidos de sensores medioambientales, meteorológicos, etc.
- A menudo **no es fácil** “capturar” estos datos de forma automática, para hacerlo hay que examinar el código en el que está escrita la página web: **HTML**

Web scraping vs Web crawling

- **Web scraping** es básicamente el proceso de coger datos de una o varias webs previamente seleccionadas
- **Web crawling**: se supone que se consulta una gran cantidad de páginas, a menudo siguiendo links entre una y otra, buscando una determinada información. Google es el ejemplo de crawler por excelencia

Páginas HTML: estructura básica



Páginas HTML + CSS

HTML Markup

```
<html>
<body>
  <h1 class="blue">Hello World!</h1>
  <h1>I'm in the middle...</h1>
  <h1 class="blue">Goodbye World!</h1>
</body>
</html>
```

+

CSS

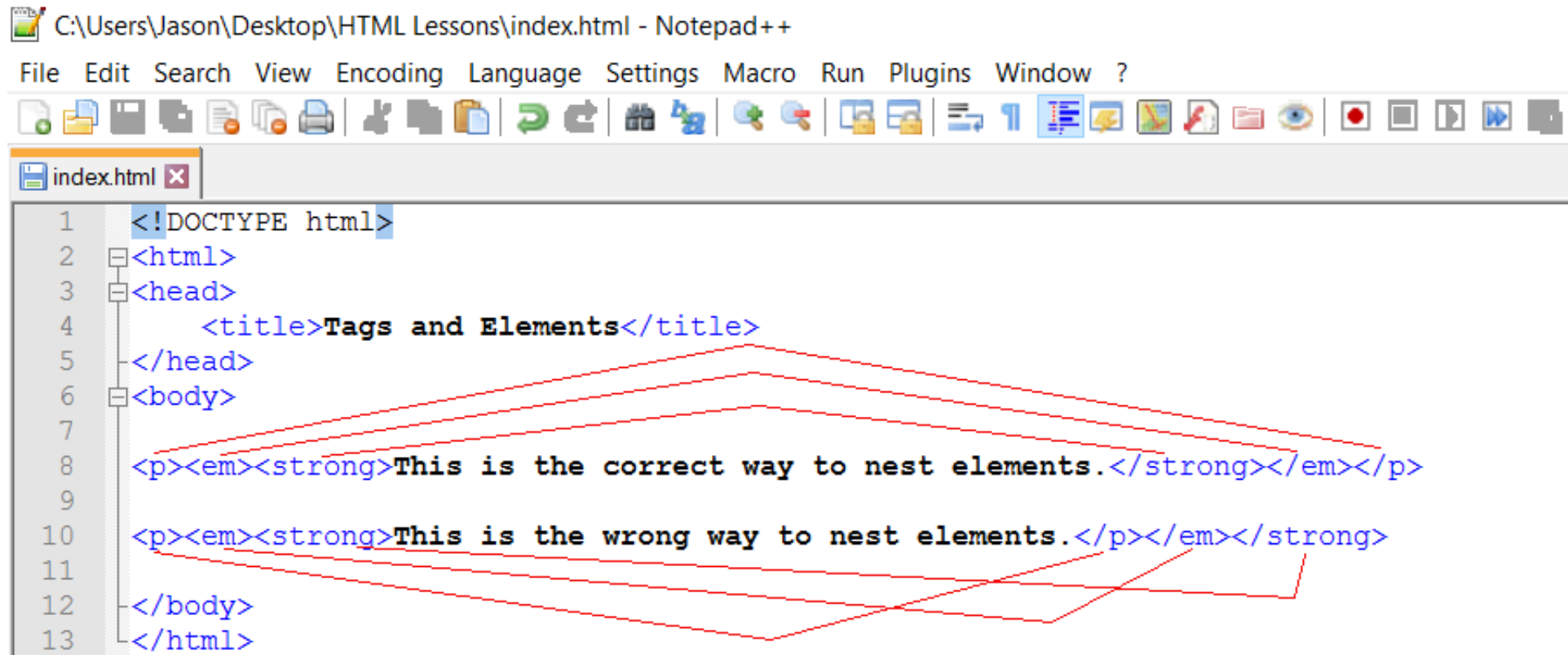
```
h1 {
  font-size: 16pt;
  font-style: italic;
  text-align: center;
  color: #FF0000;
}

.blue {
  color: #003366;
}
```

Resulting Page Render



HTML: etiquetas anidadas

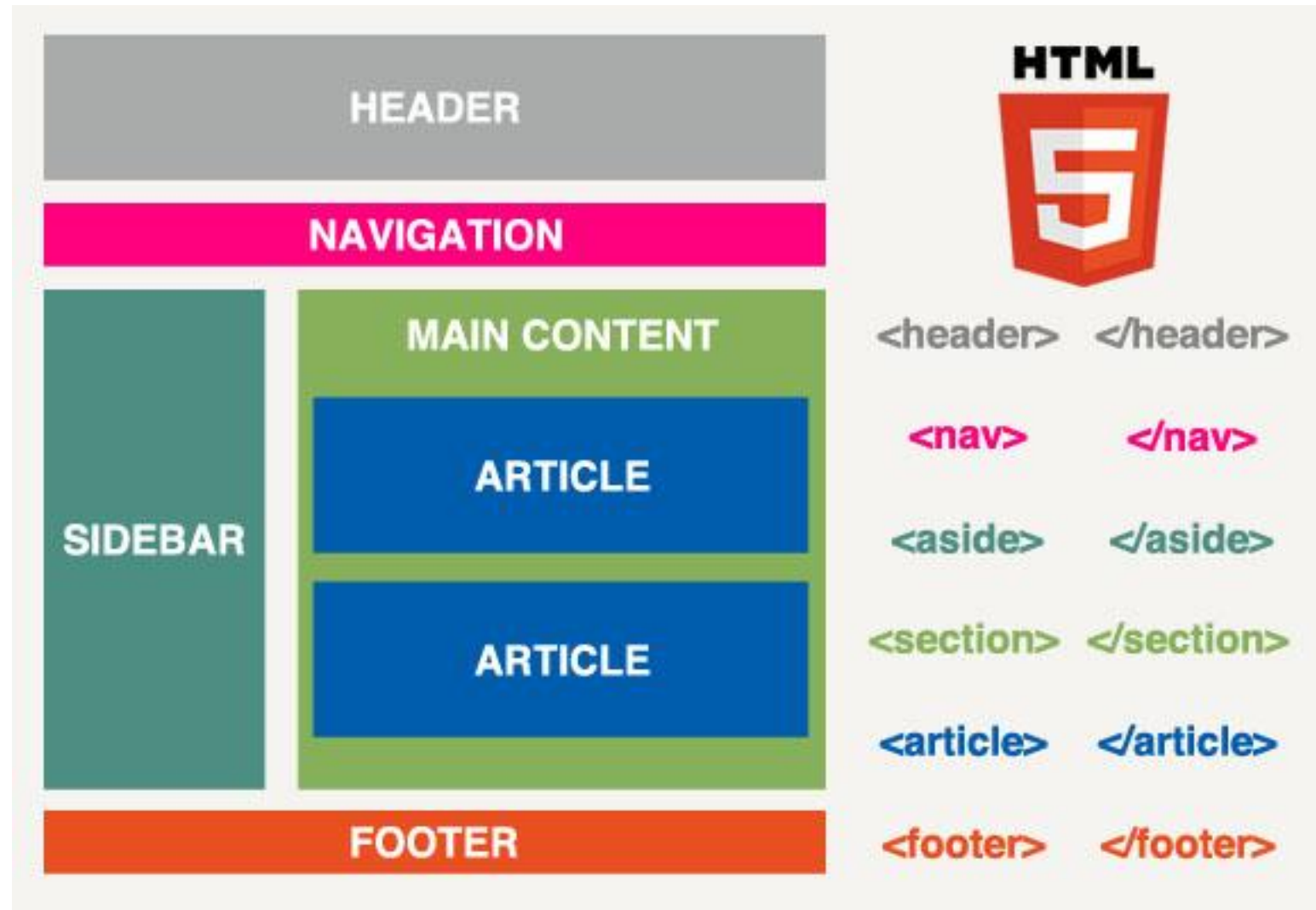


The screenshot shows a Notepad++ window with the file path `C:\Users\Jason\Desktop\HTML Lessons\index.html`. The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations and editing. The active tab is `index.html`. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Tags and Elements</title>
5 </head>
6 <body>
7
8 <p><em><strong>This is the correct way to nest elements.</strong></em></p>
9
10 <p><em><strong>This is the wrong way to nest elements.</p></em></strong>
11
12 </body>
13 </html>
```

Red lines are drawn on the code to illustrate the nesting structure. For the first paragraph (line 8), lines connect the opening `<p>` to its closing `</p>`, the opening `` to its closing ``, and the opening `` to its closing ``. For the second paragraph (line 10), lines connect the opening `<p>` to its closing `</p>`, the opening `` to its closing ``, and the opening `` to its closing ``, demonstrating an incorrect nesting order.

Páginas web: punto de vista de diseño



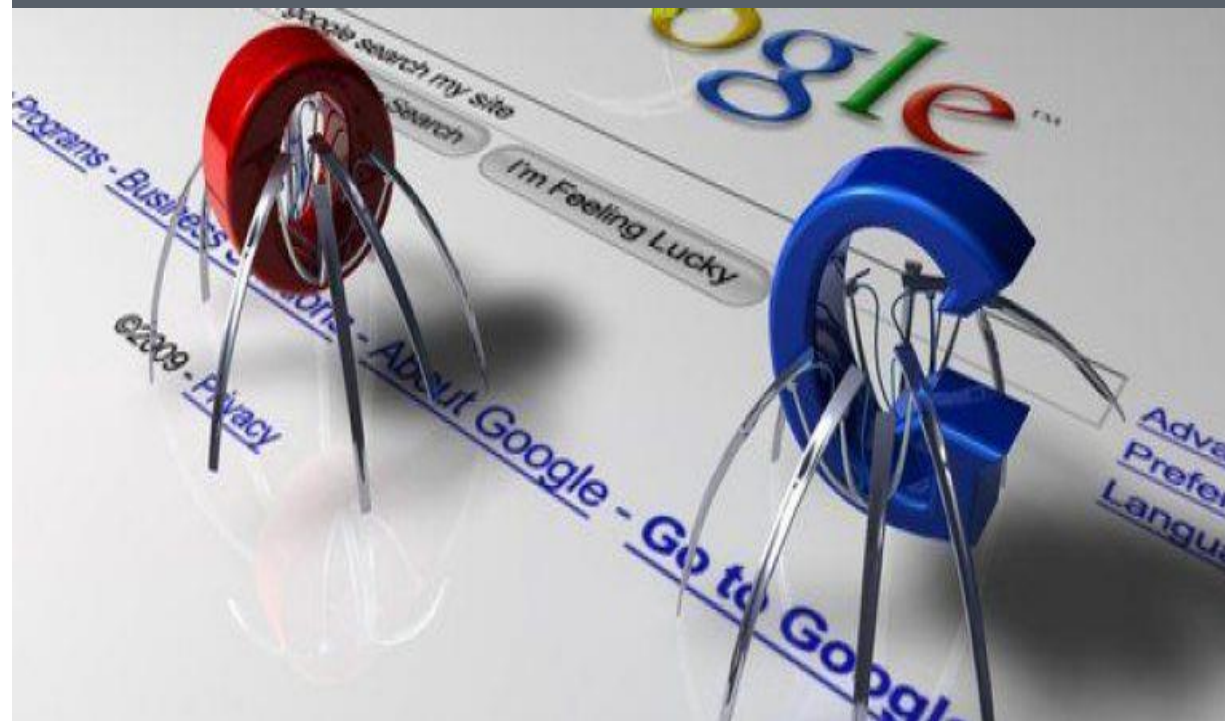
Web Scraping

Introducción

XPath

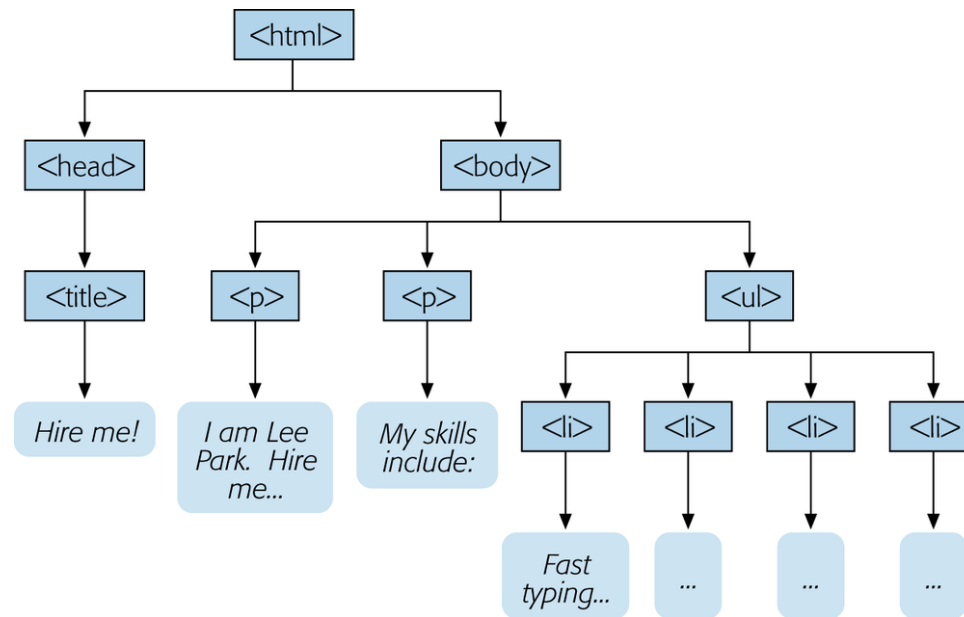
Selenium

Ejemplo



XPath: motivación

La idea detrás de XPath es ver la estructura de la página como un árbol



El lenguaje especifica cómo moverse por la página de una forma similar a las carpetas de un SO (cd)

Xpath: documento XML ejemplo

```
<?xml version="1.0" encoding="UTF-8"?>

<bookstore>

  <book>
    <title lang="en">Harry Potter</title>
    <price>29.99</price>
  </book>

  <book>
    <title lang="en">Learning XML</title>
    <price>39.95</price>
  </book>

</bookstore>
```

Expresiones XPath

Expression	Description
<i>nodename</i>	Selects all nodes with the name " <i>nodename</i> "
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

Expresiones XPath: Ejemplo

bookstore	Selects all nodes with the name "bookstore"
/bookstore	Selects the root element bookstore Note: If the path starts with a slash (/) it always represents an absolute path to an element!
bookstore/book	Selects all book elements that are children of bookstore
//book	Selects all book elements no matter where they are in the document
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
//*[contains(@lang, 'eng')]	Selects nodes with attribute lang and value 'eng'

Web Scraping

Introducción

XPath

Selenium

Ejemplo



Tipos de páginas para Scraping

1. **Estáticas:** al poner la URL en el navegador obtenemos la información como parte de la página obtenida. A su vez puede que los datos
 - a) Sean parte de la página
 - b) Estén en un fichero que hay que analizar (XML, PDF)
2. **Dinámicas:** la página incluye un formulario que se debe rellenar para llegar a la página con los datos. A su vez pueden distinguirse:
 - a) A través de URL: los datos del formulario generan una nueva URL que incluye los datos del formulario (tipo 1)
 - b) El formulario simplemente genera una nueva página

Aparte hay que tener en cuenta características específicas de páginas concretas (marcos, imágenes dinámicas, etc).

Selenium

- Una **librería** para navegar páginas WEB
- Muy útil para **hacer tests de regresión** de servicios WEB
- Se pueden cargar URLs y navegarlas
- Además permite introducir **datos en formularios**
 - → muy útil porque a menudo las páginas incluyen formularios que hay que rellenar antes de acceder a los datos

Selenium

- Disponible para Java, Python y otros lenguajes

<https://github.com/corywalker/selenium-crawler>

- En el caso de Java basta con añadir al proyecto el .jar con la librería

- Un buen tutorial de web scraping en Python con Selenium:

<http://www.discoversdk.com/blog/web-scraping-with-selenium>



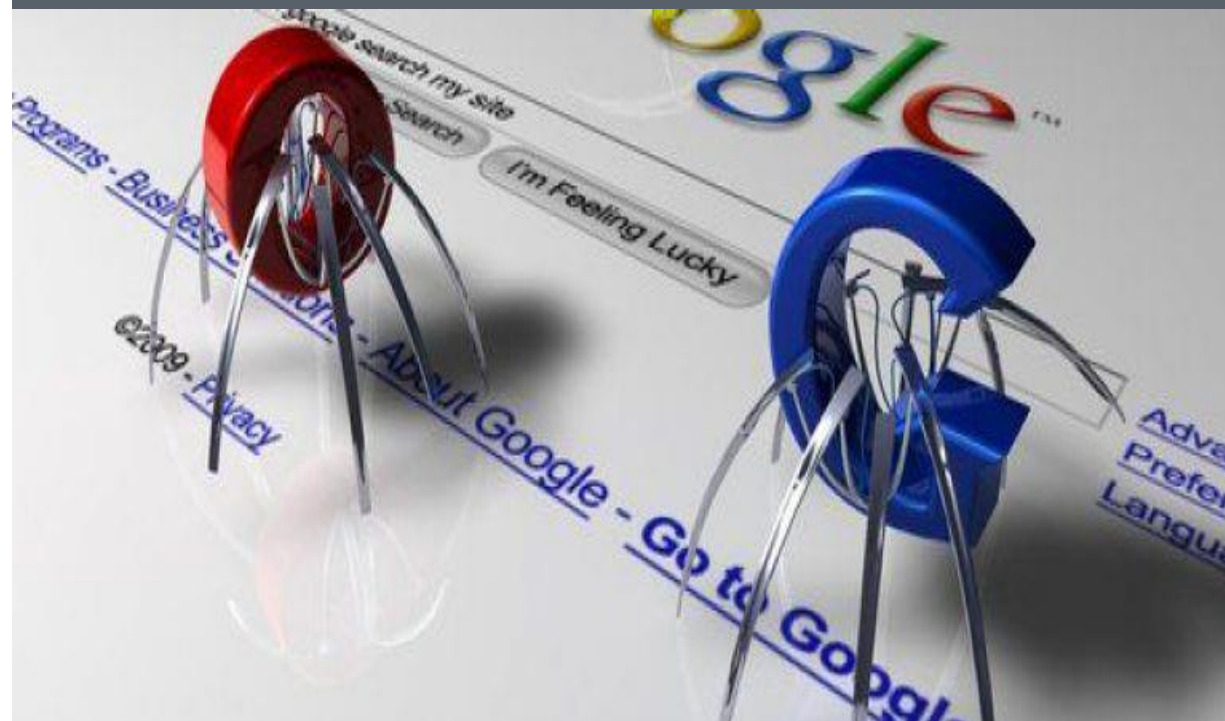
Web Scraping

Introducción

XPath

Selenium

Ejemplo



¡Gracias!

