Web scrapping:-

Web scraping is the process of extracting and collecting data from websites and storing it on a local machine or in a database.

Data Extraction with Selenium - Locating Elements

The first step of extracting the data is to locate the elements. Selenium offers a variety of find_element methods to help locate elements on a page:

- find_element_by_id - Finds element by the id attribute
- find_element_by_name - Finds element by element name attribute
- find_element_by_xpath - Finds element by XPath (Recommended)
- find_element_by_css_selector - Find element by using a CSS selector(Recommended)
- find_element_by_link_text - Find <a> elements by matching its text
- find_element_by_partial_link_text - Find <a> elements by matching its text *partially*
- find_element_by_tag_name - Finds element by the tag name
- find_element_by_class_name - Finds element by the class attribute

All these method return one instance of WebElement.

XPath

XPath is a syntax language that helps find a specific object in DOM. XPath syntax finds the elements from the root element either through an absolute path or by using a relative path. e.g.:

- / : Select child element. /html/body/div/p[1] will find the first p which is in a div tag, which in turn is a child of body element. This means that if a something will not be selected.
- //: Select all descendant element from the current element. //p will find all p elements, whether they are in a div or not.
- **[@attributename='value']**: It looks for a specific attribute with a specific value. This can also be used as [@attributename] to search for the presence of this attribute, irrespective of the value.
- XPath functions such as contains() can be used for a partial match

For example, on the web page http://books.toscrape.com, if we want to locate the link to the Humor on the navigation pane, this can be done using the contains function. Note that the text() contains white space. That's why text()="Humor" will not work. This will need to contains functions.