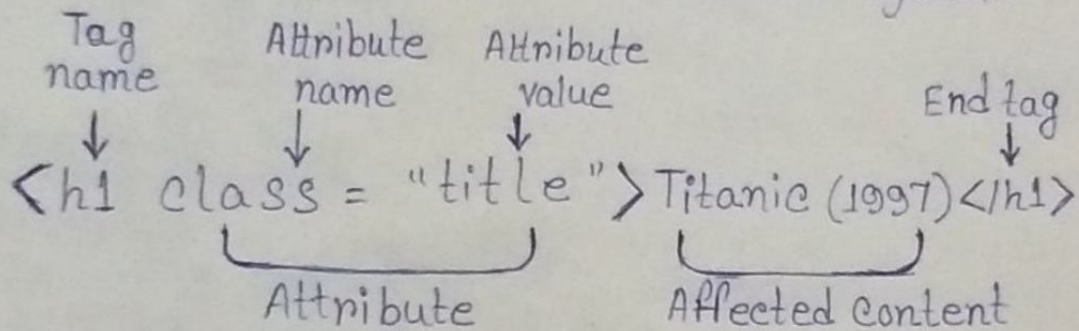


# Web Scrapping Cheat Sheet

Web Scrapping is the process of extracting data from a website. Before studying BeautifulSoup and Selenium, it's good to review some HTML basics first.

## HTML for Web Scrapping

Let's take a look at the HTML element syntax.



← HTML Element →

This is a single HTML element, but the HTML code behind a website has hundreds of them.

HTML code example

```
<article class = "main-article">
```

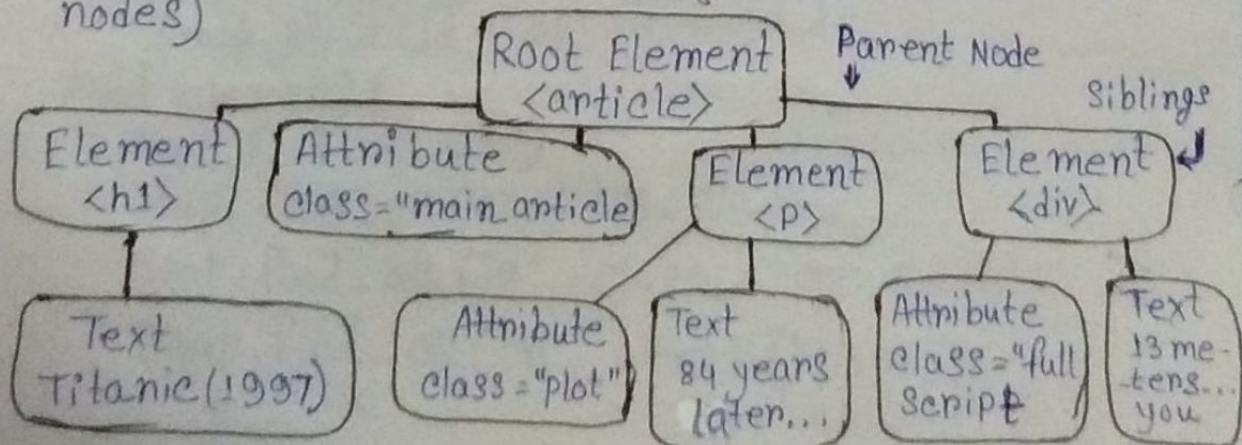
```
<h1> Titanic (1997) </h1>
```

```
<p class = "plot"> 84 years later... </p>
```

```
<div class = "full-script"> 13 meters. you... </div>
```

```
</article>
```

The HTML code is structured with "nodes". Each rectangle below represents a node (element, attribute and text nodes)





- "Siblings" are nodes with the same parent.
- A node's children and its children's children are called its "descendants". Similarly, a node's parent and its parent's parent are called its "ancestors".
- it's recommended to find element in this order.
  - a. ID
  - b. Class name
  - c. Tag name
  - d. xpath

## Beautiful Soup

### Workflow

Importing the Libraries

```
from bs4 import BeautifulSoup
import requests
```

### Fetch the pages

```
result = requests.get("www.google.com")
result.status_code # get status code
result.headers # get the headers
```

### Page Content

```
content = result.text
```

### Create Soup

```
soup = BeautifulSoup(content, "lxml")
```

### HTML in a readable format

```
print(soup.prettify())
```

### Find an element

```
soup.find(id="specific-id")
```

### Find elements

```
soup.find_all("a")
```

```
soup.find_all("a", "css-class")
```

```
soup.find_all("a", class_="my-class")
```

```
soup.find_all("a", attrs={"class":  
"my-class"})
```

Get inner text

Sample = element.get\_text()

Sample = element.get\_text(strip=True  
Separator = ' ')

Get specific attributes

Sample = element.get('href')

## X Path

We need to learn XPath to scrape with selenium or scrapy.

XPath Syntax

An XPath usually contains a tag name, attribute name, and attribute value.

// tagName[@Attribute Name = "value"]

Let's check some examples to locate the article, title, and transcript elements of the HTML code we used before.

// article[@class = "main-article"]

// h1

// div[@class = "full-script"]

## X Path Functions and Operators

XPath functions

// tag[contains(@Attribute Name, "value")]

X Path Operators; and, or

// tag[(expression 1) and (expression 2)]



## XPath Special Characters

/	Selects the children from the node set on the left side of this character
//	Specifies that the matching node set should be located at any level within the document
•	Specifies the current context should be used (refers to present node)
..	Refers to a parent node
*	A wildcard character that selects all elements or attributes regardless of name
@	Select an attribute
()	Grouping an XPath expression
[n]	Indicates that a node with index "n" should be selected