CSS Selectors - Part 2

- Locating different elements using Relative CSS Selectors (Demonstrate at http://compendiumdev.co.uk/selenium/basic_web_page.html)
  - HTML page - html
  - HTML Head - head
  - HTML Title - title
  - HTML Body - body
  - p tags - p
  - p tags inside body - body p
  - p tags inside html - html p
  - Locate p tag having id 'para2' - p[id='para2']
  - Locate p tag having class 'main' - p[class='main']
  - Locate elements having id 'para1' - [id='para1']
  - Locate elements having class 'sub' - [class='sub']
  - Using # for locating elements by ids
    - p tag having id 'para1' - p#para1
    - p tag having id 'para2' - p#para2
    - Locate elements having id 'para2' - #para2
  - Using . for locating elements by class
    - p tag having class 'main' - p.main
    - p tag having class 'sub' - p.sub
    - Locate elements having class 'main' - .main
  - (Demonstrate at http://omayo.blogspot.in/)
    - Locate input tag having value='blue' - input[value='blue']
    - Locate elements having value='blue' - [value='blue']
    - Locate all the input tags - input
    - Locate all the elements having 'value' as attribute - [value]
    - Locate all the elements having 'id' as attribute - [id]
    - Locate all the elements having 'name' as attribute - [name]
    - Locate all the elements having 'href' as attribute - [href]
    - Locate all the elements having 'src' as attribute - [src]
    - Locate all the img tags having 'src' as attribute - img[src]
  - (Demonstrate at http://compendiumdev.co.uk/selenium/basic_web_page.html)
    - Locate all the p tags having 'id' as attribute - p[id]
    - Locate all the elements having 'id' as attribute - [id]
    - Locate all the p tags having 'class' as attribute - p[class]
    - Locate all the elements having 'class' as attribute - [class]
  - (Demonstrate at http://compendiumdev.co.uk/selenium/basic_web_page.html)
    - *:first-child
      - Locate the first child inside body tag - body > *:first-child
      - Locate the first child inside head tag - head > *:first-child
      - Locate the first child inside html tag - html > *:first-child
      - Locate the first p tag - p:first-child
      - Locate the first p tag having id 'para1' - p[id='para1']:first-child
    - *:last-child
      - Locate the last child inside body tag - body > *:last-child
      - Locate the last child inside head tag - head > *:last-child
      - Locate the last child inside html tag - html > *:last-child
      - Locate the last p tag - p:last-child
      - Locate the last p tag having id 'para2' - p[id='para2']:last-child
    - *:nth-child
      - Locate the second child inside the html tag - html > *:nth-child(2)
      - Locate the first child inside the html tag - html > *:nth-child(1)
      - Locate the first child inside the body tag - body > *:nth-child(1)
      - Locate the second child inside the body tag - body > *:nth-child(2)
      - Locate the second p child inside the body tag - body > p:nth-child(2)
      - Locate the second child inside the body tag - p:nth-child(2)
      - Locate the second child having id 'para2' - p[id='para2']:nth-child(2)
      - Locate the second child having p tag and who's ancestor is html tag - html p:nth-child(2)
  - (Demonstrate at http://omayo.blogspot.in/)
    - textarea[id='ta1'] , button[id='but2'] - Works as or operator
    - * - All the elements will get highlighted
    - head > * - All the elements under head tag will get highlighted

        - body > * - All the elements under body tag will get highlighted
    - (Demonstrate at http://compendiumdev.co.uk/selenium/basic_web_page.html)
        - p[class^='ma'] - starts with
        - p[class$='ub'] - ends with
        - p[class*='ai'] - contains
    - (Demonstrate at http://compendiumdev.co.uk/selenium/basic_web_page.html)
        - p[id='para1'][class='main'] - works as and operator
        - p:not([id='para1'])
        - p:not([id='para1'])[class='sub']
        - p:not([id='para1']):not([class='main'])
    - (Demonstrate at http://compendiumdev.co.uk/selenium/basic_web_page.html)
        - Following Sibling having tag p - p[id='para1']+p
        - Following sibling having any tag - head+*
        - Demonstrate at http://book.theautomatedtester.co.uk/chapter2
            - Following sibling having link tag - title + link
    - (Demonstrate at http://omayo.blogspot.in/)
        - Locate disabled elements - *:disabled
        - Locate enabled elements - *:enabled
        - Locate selected checkbox or radio options or drop down field options etc - *:checked (Need not be default selected)
    - Easy way to generate CSS Selectors using Firepath
        - Firepath can automatically generate CSS Selectors and is the easiest way
        - If you feel that the CSS Selector returned by Firepath is not good, you can use your above knowledge in creating good CSS Selectors on your own.
            - Example: Generated CSS Selector for 'Log In' button on facebook.com
                - #loginbutton input can be used as good CSS Selector instead

---