limitations of partial link text

-> Element should be a link

eg: <span> Inbox(7) </span>

-> link ~~should~~ (or) element should be partially changing

eg : <span> 7 </span>

(or)

<a> 7 </a>

5/9/22

css selector: (cascading style sheet)

-> In selenium css selector is one of the Locator, the syntax is as followed:

Syn: tag [Attribute Name = Attribute value]

eg: a[id = 'd1']

a[name = 'n1']

a[class = 'C1']

a [href = 'https://www.jspiders.com/']

This are called as css Expression.

Note:

We can check the css Expression in the browser by using following steps:

-> Inspect any element

-> press ctrl+F

-> Type the above expressions

-> 1 of 1 = one matching elements

-> 1 of 3 -> multiple matching

0 of 0 -> No matching element

eg: css selector

```
class Democssselector{
static{
 system.setproperty(" webdriver.chrome.driver", "./driver/chromedriver.exe")
}
  PSVM() {
 WebDriver driver = new chromeDriver();
 driver.get("file:///C:/Users/Admin/Desktop/Demo.html");
 driver.findElement(By.cssselector ("a[id='d1']")).click();
  driver.navigate().back();
  driver.findElement(By.css selector("a[name='n1']")).click();
  driver.navigate().back();
  driver.find Element (By.css selector ("a[class='c1']")).click();
  driver.navigate ().back();
   driver.find Element(By.cssselector ("a[href='https://www.jspiders.com
                                             /']")).clic
  }
}
```

Note:

→ The limitations of css selector is we cannot use text. (or) It does not support text.

→ If we do any syntax mistake while writting css expressions (or) X-Path then will get invalid selector ~~expo~~ Exception.

X-Path

→ It is the path of element in a html tree is called as X-Path.

→ while writting X-path it starts with dot(.) which also represents current html documente (or) current web.page. using dot(.) is not mandatory

html tree:

```
html
  ↳ body
      ↳ a
```

X-Path

```
./html/body/a  }
   (or)        } → X-path expression.
/html/body/a   }
```

using X-path in selenium

```
driver.findElement (By.Xpath ("html/body/a")).click();
```

note:
we navigate from parent element to child element we use single (/)

/ → represents child.

Types of X-Path

→ Absolute X-Path

→ Relative X-Path

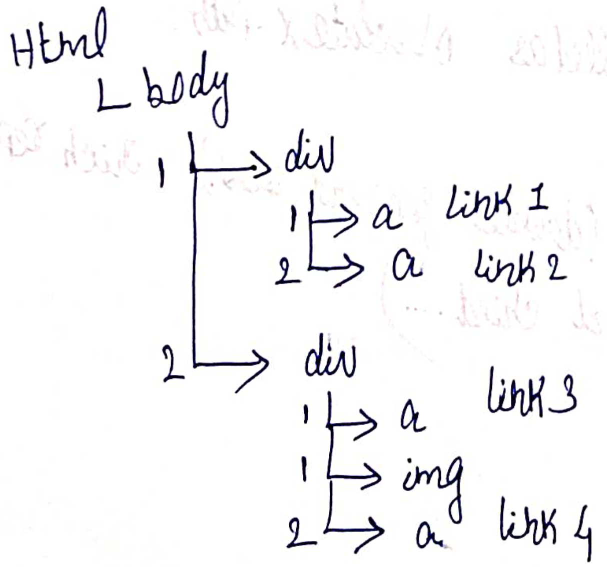
Relative X-path:

a) X-path by attribute
b) X-Path by text function.
c) X-path by contains function
d) traversing xpath
e) independent dependent Xpath
f) xpath by group index.

Absolute X-path

starting from the html tag to the required (or) desired element is called as absolute X-path.

X-path index



In X-path we can use an index which starts from 1, then if there is another element under the same parent with the same tag then index become 2 and so on.

X-Path

link 1 → /html/body/div[1]/a[1]

link 2 → /html/body/div[1]/a[2]

link 3 → /html/body/div[2]/a[1]

link 4 → /html/body/div[2]/a[2]

link 1 & 2 → /html/body/div[1]/a

link 3 & 4 → /html/body/div[2]/a

link 1 & 3 → /html/body/div/a[1]

link 2 & 4 → /html/body/div/a[2]

image → /html/body/div[2]/img (or) /html/body/div/img

link 1,2,3,4 → /html/body/div/a

All the above X-path are called as Absolute X-path.

Relative X-path:

Relative X-path starts with //(double forward slash) which represents descendants (child, grandchild, great grand child....)

link 1 → //div[1]/a[1]

link 2 → //div[1]/a[2]

link 3 → //div[2]/a[1]

link 4 → //div[2]/a[2]

link 1 & 2 → //div[1]/a

link 3 & 4 → //div[2]/a

link 1 & 3 → //div/a[1]

link 2 & 4 → //div/a[2]

image → //div[2]/img (or) //div/img.