

3/9/12

Example for tag name locator

```
class DemoLocators {
```

```
static {
```

```
    System.setProperty("webdriver.chrome.driver", ".\\driver\\chromedriver.exe");
```

```
}
```

```
    public void {
```

```
        WebDriver driver = new ChromeDriver();
```

```
        driver.get("file:///C:/Users/Admin/Desktop/demo.html");
```

```
        WebElement e = driver.findElement(By.tagName("a"));
```

```
        e.click();
```

```
    }
```

```
}
```

sample html code:

```
<html>
```

```
<body>
```

```
<a id="d1" name="n1" class="c1" href="https://www.jspiders.com/">Google </a> <br>
```

```
<a id="d1" name="n1" class="c1" href="https://www.jspiders.com/">Google </a>
```

```
</body>
```

```
</html>
```

Note:

In order to create the static web page type the above html code in the notepad & save it as demo.html on the desktop.

Interview
1) What is the return type of find element method?

WebElement.
2) If the locator is not matching with any of the element, then what find element does?

It throws ~~no search element~~ exception.
NoSuchElementException

3) If the specified locator is matching with multiple elements, then what find element does?

It takes first matching element. (or) It returns the address of first matching element.

4) What is the argument accepted by find element()? By type.

Example for id, name, class name, locator:

```
class Locators {
```

```
static {
```

```
System.setProperty("webdriver.chrome.driver", ".\\driver\\chromedriver.exe");
```

```
}
```

```
psvm() throws InterruptedException {
```

```
WebDriver driver = new ChromeDriver();
```

```
driver.get("file:///C:/users/Admin/Desktop/demo.html");
```

```
Thread.sleep(2000);
```

```
driver.findElement(By.tagName("a")).click();
```

```
Thread.sleep(2000);
```



```
driver.navigate().back();
```

```
Thread.sleep(2000);
```

```
driver.findElement(By.id("d1")).click();
```

```
Thread.sleep(2000);
```

```
driver.navigate().back();
```

```
Thread.sleep(2000);
```

```
driver.findElement(By.name("n1")).click();
```

```
Thread.sleep(2000);
```

```
driver.navigate().back();
```

```
Thread.sleep(2000);
```

```
driver.findElement(By.className("c1")).click();
```

```
} }
```

Link text & Partial Link text:

Both link text & partial link text locators are used to find the link. If we try it on any other type of element we get no ~~such~~ ^{such} element exception.

eg. driver.findElement(By.linkText("Google")).click();

If the text of link is changing partially we can use partial link text.

eg: {a...}>Inbox(1) → html code

driver.findElement(By.partialLinkText("Inbox")).click();

↳ selenium code

limitations of partial link text

→ Element should be a link

eg: `inbox(7)`

→ link ~~should~~ (or) element should be partially changing

eg: ` 7 `
(or)

`<a> 7 `