

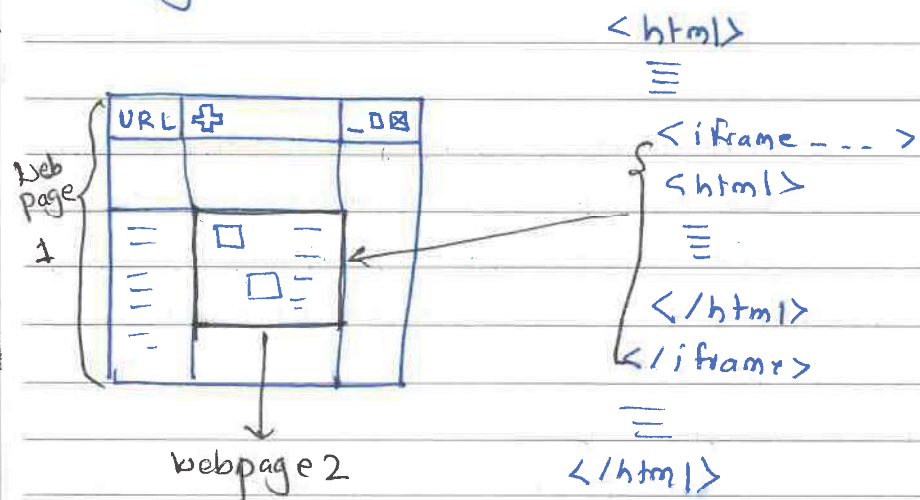
```
public class FrameHandling {
    p.s.v.m (S.L.J.A.)
}
```

system.setProperty()

HANDLING FRAME :-

Embedding a webpage inside another web page is known as 'Frame concept'.

This is done by using an html tag `<iframe>`
For e.g.



In order to identify the required element of the webpage is a part of frame or not, just

right click on that element. if in the context menu you find a word 'called' 'frame' then that indicates the required element is present inside frame.

By default, the control will be in the main page hence webdriver cannot perform any operation on the elements present inside the frame.

In order to do this, first we should make webdriver to switch to required frame by using below mentioned statement -

`driver.switchTo().frame(args);`

int
String index or String Name or WebElement element

The above frame method is a overloaded method of 'targetLocator' interface. It can take any one of the below mentioned argument -
i> int index
ii> string name
iii> WebElement element

After switching to the Frame, the find element() method will identify the elements present inside the frame only.

In order to access the element present in the main page, we should use below mentioned statement,

```
driver.switchTo().defaultContent();
```

Foreg.

```
public class FrameHandling {
```

```
    p.s.v.m (SEJa)
    {
```

```
        system.setProperty("webdriver.chrome.driver",
```

```
        "path" + chromedriver.exe");
```

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

```
        driver.manage().window().maximize();
```

```
        driver.get("http://jqueryui.com/draggable/");
```

```
        /* Create an instance of actions class */
```

```
        Actions actions = new Actions(driver);
```

```
        /* Switch to required frame using index number */
```

```
        driver.switchTo().frame(0);
```

or

```
        /* identify the frame */
```

```
        WebElement FrameObject = driver.findElement(By.
        cssSelector("demo-frame"));
```

```
        /* switch to frame using webElement */
        driver.switchTo().frame(FrameObject);
```

```
        WebElement drag = driver.findElement(By.id
        ("draggable"));
```

```
        /** Do drag & drop with offset values /
        actions.dragAndDrop(drag, 100, 150).perform();
```

```
        /* switch back to the main page */
```

```
        driver.switchTo().defaultContent();
```

```
        driver.findElement(By.cssSelector(".logo")).click();
        }
```


HANDLING POPUPS :-

As per web applⁿ, pop-ups are categorised into two categories -

- 1) JAVA script popup
- 2) HTML popup (hidden division popups)

i) Handling Java script popup -

As the name indicates, these pop-ups are developed using java script hence there will be no html code visible & also these popups are right click protected.

These popups are categories in 3 subcategories

- 1) Alerts popup - which will have some text along with single button that can be 'OK' or 'cancel' or 'close'.

ii) Confirmation popup -

This popup will also have some text along with more than one button & might also contains a checkbox.

Prompt popup -

This popup also contains some text, input field along with some buttons.

In order to handle any of the above popups, we need to use below mentioned statement.
`driver.switchTo().alert()`

In alert we can perform following operations like,
In order to click on 'OK' or 'Yes' button, we can use `accept()` method of alert interface.
`driver.switchTo().alert().accept()`

In order to click on cancel or 'no' button, use `dismiss()` method of alert interface.
`driver.switchTo().alert().dismiss()`

In order to retrieve text from a alert, we can use `getText()` of alert interface.
`driver.switchTo().alert().getText()`

In order to type something on input field present in java script popup, we can use `sendKeys()` method of alert interface.
`driver.switchTo().alert().sendKeys()`

Note :-

After every action on alert, the control will automatically switch back to the main page, hence if you want to perform another action on java script popup then you need to switch again into this popup.

```
public class Handling JavaScriptPopups {  
    p.s.v.m {
```

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

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

```
        driver.manage().window().maximize();
```

```
        driver.get("https://www.tutorialspoint.com/  
        javascript/java-script-dialog-boxes.htm");
```

```
        /* create an instance of actions class */
```

```
        Actions actions = new Actions(driver);
```

```
        /* Scroll down the page */
```

```
        driver.findElement(By.cssSelector("body")).
```

```
        sendKeys(keys.PAGE-DOWN);
```

```
        /* Identify all the frame having alert button */
```

```
        List<WebElement> frameList = driver.findElements  
        (By.xpath("//iframe[@class='result']"));
```

```
        /* Handling alert popups */
```

```
        /* Switch to first frame */
```

```
        driver.switchTo().frame(frameList.get(0));
```


/* identify & click on alert button */

```
driver.findElement(By.xpath("//input[@value='Click Me']")).click();
```

/* get the alert popup text & click on ok button */

```
s.o.p(driver.switchTo().alert().getText());
driver.switchTo().alert().accept();
```

// OR

/* Store Alert instance */

```
// Alert alert = driver.switchTo().alert();
```

```
// s.o.p(alert.getText());
```

```
// alert.accept();
```

/*

/* Switch back to the main page */

```
driver.switchTo().defaultContent();
```

/* Handling confirmation popups */

/* Scroll down the page 2 time */

```
driver.findElement(By.cssSelector("body")).
```

```
sendKeys(Keys.PAGE-DOWN);
```

_____ // (for 2nd scroll down)

/* Switch to the second frame */

```
driver.switchTo().frame(frameList.get(1));
```

/* identify & click on alert button */

```
driver.findElement(By.xpath("//input[@value='Click Me']")).click();
```

/* Store alert instance */

```
Alert alert1 = driver.switchTo().alert();
```

```
s.o.p(alert1.getText());
```

```
alert1.dismiss();
```

/* Switch back to main page */

```
driver.switchTo().defaultContent();
```

/* Handling prompt popups */

scroll

/* Switch down the page */

```
driver.findElement(By.cssSelector("body")).
```

```
sendKeys(Keys.PAGE-DOWN);
```

/* Switch to the 2nd frame */

```
driver.switchTo().frame(frameList.get(2));
```

/* identify & click on alert button */

```
driver.findElement(By.xpath("//input[@value='Click Me']")).click();
```

/* Store alert Instance */

```
Alert alert2 = driver.switchTo().alert();
```

```
s.o.p (alert2.getText());
```

```
alert2.sendKeys(" @spiders ");
```

```
alert2.accept();
```

```
s.o.p (driver.findElement(By.tagName("body")).getText());
```

/* Switch back to Main Page */

```
driver.switchTo().defaultContent();
```

Assignment :-

- 1) Login actitime applⁿ
- 2) Navigate to task page
- 3) Click on add new task & create new task
- 4) select From customer dropdown, select new customer, enter customer name & project name.
- 5) Fill the task name.

6) Select the deadline as 26th Jan

7) Work Type should be Testing.

8) Finally click on create task button.

Validation →

9) Verify thatⁱⁿ the task page, above created task is present or not.

Part 2:

1) Select the check box of newly created task, click on delete button. Get all the information from the pop-up.

11) After deletion verify that whether the task is deleted successfully or not

12) Logout from the applⁿ.