



Selenium

Chapter 04



Neler Öğreneceğiz?



- DropDown
- Synchronization & Selenium Waits
- Alerts
- Basic Authentication
- iFrame
- Test Base





DropDown

Selenium'daki **Select Class** , HTML **SELECT** etiketini uygulamak için kullanılan bir yöntemdir. HTML seçim etiketi, öğeleri seçmek ve seçimlerini kaldırmak için yardımcı yöntemler sağlar. Select sınıfı sıradan bir sınıftır, bu nedenle nesnesini oluşturmak için **new** anahtar sözcüğü kullanılır ve web öğesinin konumunu belirtir.

```
WebElement testDropDown = driver.findElement(By.id("testingDropdown"));
Select dropdown = new Select(testDropDown);
```

Syntax: `Select select = new Select (WebElement webelement);`

```
Select
Select org.openqa.selenium.support.ui
org.openqa.selenium.support.ui
public class Select
extends Object
```



Select Methods

```
Select select = new Select(driver.findElement(By.id("oldSelectMenu")))
```

select.

m	deselectAll()	void
m	deselectByIndex(int index)	void
m	deselectByValue(String value)	void
m	deselectByVisibleText(String text)	void
m	equals(Object o)	boolean
m	getAllSelectedOptions()	List<WebElement>
m	getFirstSelectedOption()	WebElement
m	getOptions()	List<WebElement>
m	getWrappedElement()	WebElement
m	hashCode()	int
m	isMultiple()	boolean
m	selectByIndex(int index)	void
m	selectByValue(String value)	void
m	selectByVisibleText(String text)	void

1

2

3

En yaygın kullanılanlar ;

selectByIndex

selectByValue

selectByVisibleText

Select Methods



SelectByIndex: Argüman olarak index numarası olarak bir tamsayı değeri alır, index 0 dan başlar.

```
selectByIndex(int arg0) : void
```

EXAMPLE

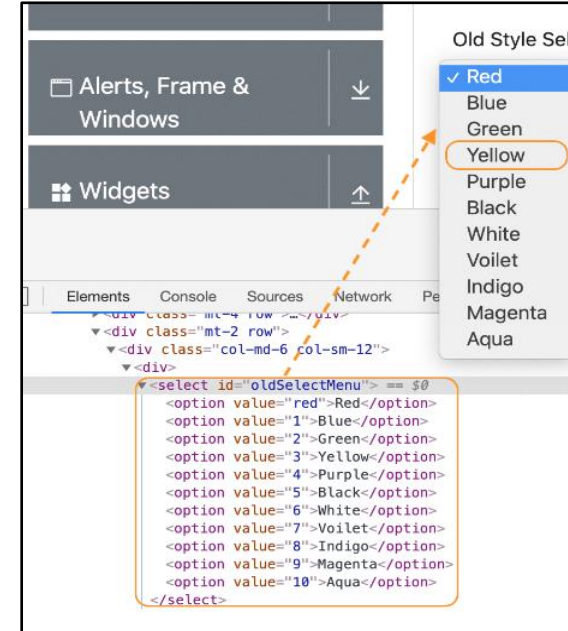
<https://demoqa.com/select-menu>

Create object of the Select class

```
Select se = new Select(driver.findElement(By.xpath("//*[@id='oldSelectMenu']")));
```

Select the option by index

```
se.selectByIndex(3);
```



Select Methods



SelectByValue: Argüman olarak String alır.

```
selectByValue(String arg0) : void
```



<https://demoqa.com/select-menu>

Create object of the Select class

```
Select se = new Select(driver.findElement(By.xpath("//*[@id='oldSelectMenu']")));
```

Select the option with value "6"

```
se.selectByValue("6");
```

```
<select id="oldSelectMenu"> == $0
  <option value="red">Red</option>
  <option value="1">Blue</option>
  <option value="2">Green</option>
  <option value="3">Yellow</option>
  <option value="4">Purple</option>
  <option value="5">Black</option>
  <option value="6">White</option>
  <option value="7">Voilet</option>
  <option value="8">Indigo</option>
  <option value="9">Magenta</option>
  <option value="10">Aqua</option>
</select>
```

Select Methods



SelectByVisibleText: Argüman olarak String alır, açılır menü listesinden bir seçeneğin seçilmesini sağlar.

EXAMPLE

<https://demoqa.com/select-menu>

Create object of the Select class

```
Select se = new Select(driver.findElement(By.xpath("//*[@id='oldSelectMenu']")));
```

Select the option using the visible text

```
se.selectByVisibleText("White");
```

```
<select id="oldSelectMenu"> == $0
  <option value="red">Red</option>
  <option value="1">Blue</option>
  <option value="2">Green</option>
  <option value="3">Yellow</option>
  <option value="4">Purple</option>
  <option value="5">Black</option>
  <option value="6">White</option>
  <option value="7">Voilet</option>
  <option value="8">Indigo</option>
  <option value="9">Magenta</option>
  <option value="10">Aqua</option>
</select>
```

Select Methods



isMultiple()

Select sınıfı, ilk önce açılır listenin birden çok değer seçmemize izin verip vermediğini doğrulayabileceğimiz isMultiple() yöntemini sağlar.

No parameters needed.

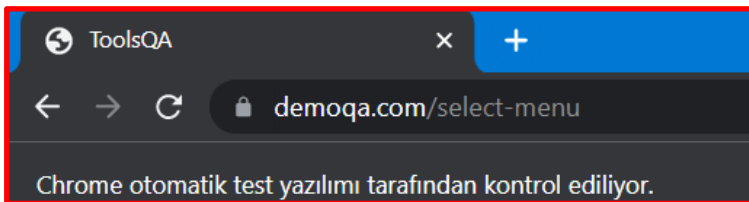
```
isMultiple(): boolean
```

```
//Selecting Items in a Multiple SELECT elements  
driver.get("http://jsbin.com/osebed/2");  
Select fruits = new Select(driver.findElement(By.id("fruits")));  
fruits.selectByVisibleText("Banana");  
fruits.selectByIndex(1);
```




TIME FOR TASK

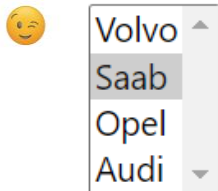
Multiple Select



Go to URL: <https://demoqa.com/select-menu>

Select options in Standard multiple select

Standard multi select





Select Methods

getOptions: Argüman olarak String alır, açılır menü listesinden bir seçeneğin seçilmesini sağlar.

```
getOptions(): List<WebElement>
```

<https://demoqa.com/select-menu>

EXAMPLE

```
Select select = new Select(driver.findElement(By.id("oldSelectMenu")));
```

Get all the options of the dropdown

```
List<WebElement> options = select.getOptions();
```



Select Methods

getFirstSelectedOption(): Açılır menüden ilk seçilen seçeneği döndürür.

```
getFirstSelectedOption(): WebElement
```

EXAMPLE

<https://demoqa.com/select-menu>

```
Select select = new Select(driver.findElement(By.id("oldSelectMenu")));
```

Get the first selected option of the dropdown

```
WebElement firstSelectedOption = select.getFirstSelectedOption();
```



TIME FOR TASK

Get Options Test

Get all the options of the dropdown
Options size
Print all test
Verify the dropdown has option "**Black**"
Print FirstSelectedOptionTest
Select option "**Yellow**"





Select Methods

getAllSelectedOptions(): Açılır listenin tüm seçili seçeneklerini döndürür.

```
getAllSelectedOptions():List<WebElement>
```

EXAMPLE

<https://demoqa.com/select-menu>

```
Select select = new Select(driver.findElement(By.id("oldSelectMenu")));
```

Get all the selected option of the dropdown

```
List<WebElement> selectedOptions = select.getAllSelectedOptions();
```

Select Methods



```
Select select = new Select(driver.findElement(By.id("oldSelectMenu")))
```

select.	
m deselectAll()	void
m deselectByIndex(int index)	void
m deselectByValue(String value)	void
m deselectByVisibleText(String text)	void
m equals(Object o)	boolean
m getAllSelectedOptions()	List<WebElement>
m getFirstSelectedOption()	WebElement
m getOptions()	List<WebElement>
m getWrappedElement()	WebElement
m hashCode()	int
m isMultiple()	boolean
m selectByIndex(int index)	void
m selectByValue(String value)	void
m selectByVisibleText(String text)	void

deselectAll() yöntemini kullanarak tüm seçeneklerin seçimini kaldırabilirsiniz.

Deselect all the options
`select.deselectAll();`

deselectByIndex() yöntemini kullanarak seçeneklerden birinin seçimini kaldırabilirsiniz.

Deselect first value by index
`select.deselectByIndex(1);`

deselectByValue() yöntemini kullanarak açılır menüden bir seçeneğin seçimini kaldırma yöntemini sağlar.

Deselect option with value "6"
`select.deselectByValue("6");`

deselectByVisibleText() yöntemini kullanarak seçeneklerden birinin seçimini kaldırabilirsiniz.

Deselect option with text "White"
`select.deselectByVisibleText("White");`



DropDown – Faker Class

TIME FOR TASK

Go to URL: <https://opensource-demo.orangehrmlive.com/>

Login with Username: Admin

Login with Password: admin123

Click login button

Click on PIM

Click on Employee List

Employee Name → Use Faker Class

ID -> Use Faker Class

Employment Status → select by index: 2

Include → selectByVisibleText: Current and Past Employees

Supervisor Name → Use Faker Class

Job Title ->selectByValue: IT Manager

Sub Unit ->selectByValue: 3

Click search button

Verify text visible : "No Records Found"



DropDown



Go to URL: <https://the-internet.herokuapp.com/dropdown>

Create method `selectByIndexTest` and Select Option 1 using index .

Create method `selectByValueTest` Select Option 2 by value.

Create method `selectByVisibleTextTest` Select Option 1 value by visible text.

Create method `printAllTest` Print all dropdown value.

Verify the dropdown has Option 2 text.

Create method `printFirstSelectedOptionTest` Print first selected option.

Find the size of the dropdown, Print "Expected Is Not Equal Actual" if there are not 3 elements in the dropdown.



HMW



- Launch the browser.
- Open "**<https://demoqa.com/select-menu>**".
- Select the Old Style Select Menu using the element id.
- Print all the options of the dropdown.
- Select 'Purple' using the index.
- After that, select 'Magenta' using visible text.
- Select an option using value.
- Close the browser.



HMW

TIME FOR TASK

- Launch the browser.
- Open "<https://demoqa.com/select-menu>".
- Select the Standard Multi-Select using the element id.
- Verifying that the element is multi-select.
- Select 'Opel' using the index and deselect the same using index.
- Select 'Saab' using value and deselect the same using value.
- Deselect all the options.
- Close the browser.



TIME FOR TASK

Free Work

<https://demo.guru99.com/test/newtours/register.php>

Bu sayfada çalışma yapabilirsiniz.

REGISTER

To create your account, we'll need some basic information about you. This information will be used to send reservation confirmation emails, mail tickets when needed and contact you if your travel arrangements change. Please fill in the form completely.

Contact Information

First Name:

Last Name:

Phone:

Email:

Mailing Information

Address:

City:

State/Province:

Postal Code:

Country:

User Information

User Name:

Password:

Confirm Password:

Gönder

Synchronization & Selenium Waits



Herhangi bir eylemi gerçekleştirmek için iki veya daha fazla bileşen dahil olduğunda, bu bileşenlerin aynı hızda birlikte çalışmasını bekleriz. Bu bileşenler arasındaki paralel olarak çalışacak koordinasyona **senkronizasyon** denir.

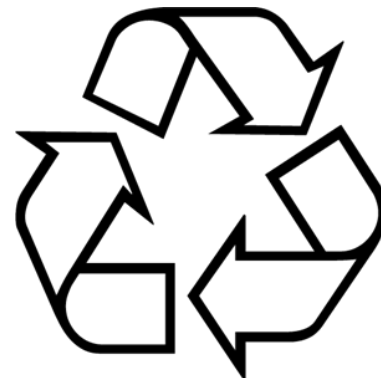
Komut dosyası yürütme ve uygulama arasında senkronizasyon yapmak için uygun işlemleri yaptıktan sonra beklememiz gerekiyor.

Synchronization is very very important for UI automation testing.

NO SUCH ELEMENT EXCEPTION almamak için bu yöntemler uygulanmalıdır.

Senkronizasyon iki kategoriye ayrılabilir:

1. **Unconditional Synchronization(Static Wait)** → Koşulsuz
2. **Conditional Synchronization(Dynamic Wait)** → Koşullu



Selenium & JAVA Bekleme Mekanizması



Unconditional Synchronization Methods

Waiting Mechanism

Conditional Synchronization Methods

Java

Selenium

`Thread.sleep()`

Implicit Wait

Explicit Wait

`driver.manage().timeouts().implicitlyWait()`

`Wait (i)`

`WebDriverWait (c)`

`FluentWait (c)`

Java'ya aittir ve WebElement görüntüldikten sonra bile beklemeye devam eder.

Genel bir beklemedir ve programın başında bir kez bildirildiğinde, tüm WebElement'leri için geçerli olacaktır.

Yoklama süresi ve istisnaları yok sayma konusunda daha fazla esnekliğe sahip WebDriverWait'e benzerdir.

Belirli bir WebElement için açıkça bekler.

Conditional Synchronization Methods



Implicit Wait is defined as a Global Wait timer.

Once Implicit Wait is defined in the script, WebDriver will automatically wait for every element as a specified amount of time.

```
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```
WebElement homeLink = driver.findElement(By.linkText("Home"));  
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(15));  
wait.until(ExpectedConditions.visibilityOf(homeLink));  
wait.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("Home")));
```

Explicit Wait makes your WebDriver to wait for a specific amount of time for a **Target Element** before throwing an exception. Using Explicit Wait you can target a specific element.

Wait for the
WebElement **repeatedly**
at a **regular** interval of
time until the **timeouts**.

```
Wait<WebDriver> waitt = new FluentWait<WebDriver>(driver).  
    withTimeout(Duration.ofSeconds(10)).  
    pollingEvery(Duration.ofMillis(10)).  
    ignoring(NoSuchElementException.class);
```

Alerts



Alertler; görüntüleyen ve belirli işlemler için izin isteyen bazı bilgileri kullanıcıya bildiren küçük açılır kutular/pencerelerdir. Ek olarak, bunları uyarı amaçlı da kullanabiliriz. Bazen kullanıcı uyarı kutusuna birkaç bilgi de girebilir.

Danger! Indicates a dangerous or potentially negative action.



Danger

Success! Indicates a successful or positive action.



Success

Info! Indicates a neutral informative change or action.



Information

Warning! Indicates a warning that might need attention.



Warning

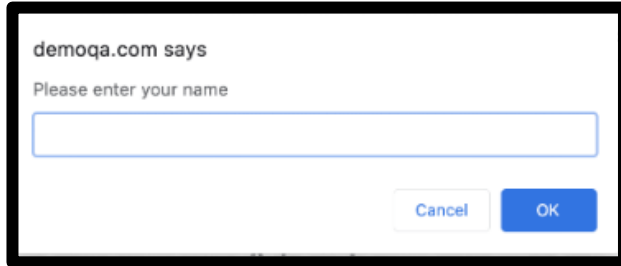


Alerts



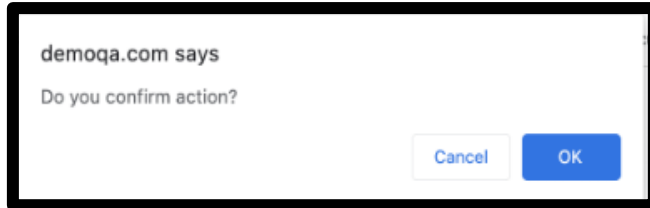
Simple Alert

Bu uyarılar yalnızca **bilgi** amaçlı uyarılardır ve üzerlerinde bir **OK** düğmesi bulunur. Kullanıcılar, uyarı kutusunda görüntülenen mesajı okuduktan sonra **OK düğmesine tıklayabilir**.



Prompt Alert

Bu uyarılar kullanıcının uyarı kutusuna girmesi gereken metin şeklinde bazı **giriş gereksinimleri** vardır. Kullanıcının kullanıcı adını girip **OK** düğmesine basabileceği veya herhangi bir ayrıntı girmeden uyarı kutusunu **iptal** edebilir.



Confirmation Alert

Kullanıcıdan mesaj kutusunu **kabul etme** veya **reddetme şeklinde bir miktar onay alır**. Kullanılabilir metin kutusu olmadığı için kullanıcının hiçbir şey giremeyeceği şekilde istem uyarılarından farklıdır. Kullanıcılar sadece **OK/ Cancel** düğmesine basarak mesajı okuyabilir ve girişleri sağlayabilir.

Alerts



Void accept(): 'OK' button of the alert box.

```
driver.switchTo( ).alert( ).accept();
```

Void dismiss(): 'Cancel' button clicks in the alert box.

```
driver.switchTo( ).alert( ).dismiss();
```

String getText(): Message from the alert box .

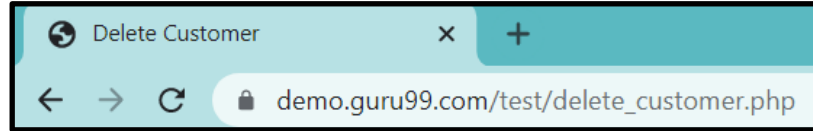
```
driver.switchTo().alert().getText();
```

Void sendKeys (String stringToSend): Ends data to the alert box .

```
driver.switchTo().alert().sendKeys("Text");
```



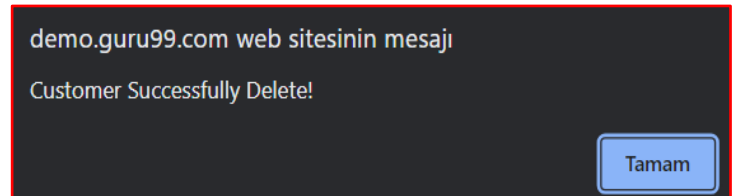
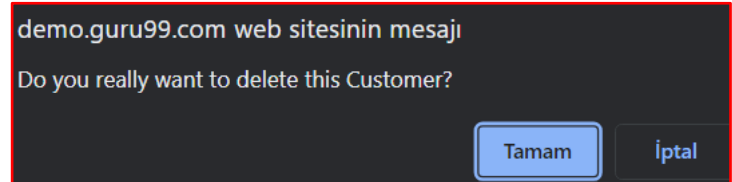
Alerts



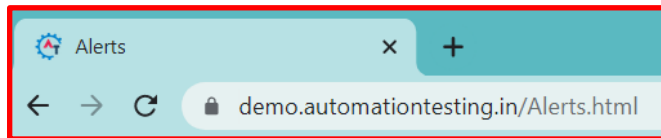
Go to URL: http://demo.guru99.com/test/delete_customer.php
Delete customer ID: 123
Delete 2 alerts that appear.

Delete Customer Form

Customer ID



Alerts



Go to URL: <http://demo.automationtesting.in/Alerts.html>

Click "Alert with OK" and click 'click the button to display an alert box:'

Accept Alert(I am an alert box!) and print alert on console.

Click "Alert with OK & Cancel" and click 'click the button to display a confirm box'

Cancel Alert (Press a Button !)

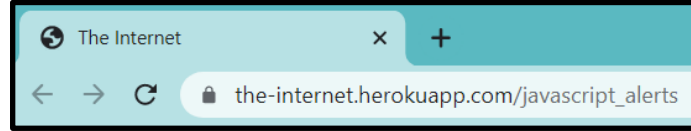
Click "Alert with Textbox" and click 'click the button to demonstrate the prompt box'

And then sendKeys «BootcampCamp» (Please enter your name)

Finally print on console this message "Hello BootcampCamp How are you today" assertion these message.



Free Work



https://the-internet.herokuapp.com/javascript_alerts

Bu sitede pratik yapabilirsiniz.

Click for JS Alert

Click for JS Confirm

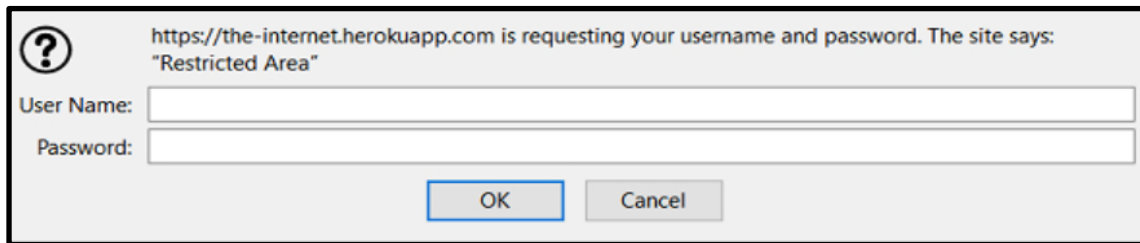
Click for JS Prompt

Basic Authentication



Tarayıcı kimlik doğrulamasını Selenium web sürücüsü ile halledebiliriz.

Kullanıcı **Adı** ve **Şifre** doğru bir şekilde girildikten ve **Tamam düğmesine** tıklandıktan sonra **Tebrikler** yazısının bulunduğu asıl sayfaya yönlendiriliriz. **Uygun kimlik bilgilerine sahip olmalısınız** .



URL ile eklenen kimlik bilgilerini geçmeliyiz. Kullanıcı adı ve şifre, **https://kullanıcıadı:şifre@URL** biçiminde eklenmelidir.

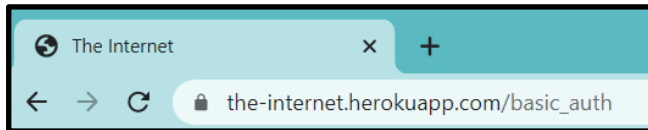
`https://username:password@URL`

URL ile eklenen kimlik bilgilerini geçmeliyiz. Kullanıcı adı ve şifre, **https://kullanıcıadı:şifre@URL** biçiminde eklenmelidir.

SE 4



Basic Authentication



Go to URL: https://the-internet.herokuapp.com/basic_auth

Username: admin

Password: admin

Syntax: `driver.get("https://USERNAME:PASSWORD@URL");`

Authenticate using the above method and test data.

`https://admin:admin@the-internet.herokuapp.com/basic_auth`

Verify the congratulations message.

Basic Auth

Congratulations! You must have the proper credentials.

iFrame



iFrame, bir HTML belgesinin içine gömülü bir HTML belgesidir.

<iframe></iframe>

iFrame, HTML'de bir etiketle tanımlanır.

```
<div id="modal">  
  <iframe id="buttonframe" name="myframe" src="https://seleniumhq.github.io">  
    <button>Click here</button>  
  </iframe>  
</div>
```



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



**No Such Element
Exception**



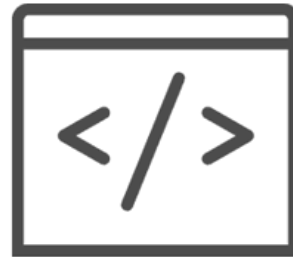
iFrame



iFrame'ler arasında geçiş yapmamız gerektiğini anlamalıyız. iFrame'ler arasında geçiş yapmak için driver'ın **switchTo().frame()** komutunu kullanmalıyız. switchTo().frame() 'i üç şekilde kullanabiliriz.

- **switchTo.frame (int frame Number)**
- **switchTo.frame (string frame Name Or Id)**
- **switchTo.frame (WebElement frame Element)**

Bir sayfada bir iframe varsa, Selenium bir iframe içindeki öğeleri doğrudan göremez.



<iFrame>

iFrame



switchTo.frame (int frame Number): iFrame index'ini yazıp driver'ın o iframe geçmesini sağlanabilir. iFrame index 0'dan başlar.

driver.switchTo().frame(0); → 0. indexteki iframe'e geçmek için → **First frame**

driver.switchTo().frame(1); → 1. indexteki iframe'e geçmek için → **Second frame**

Frame 0

```
<div id="framesWrapper">
  <div>...</div>
  <div id="frame1Wrapper">
    <iframe src="/sample" id="frame1" width="500px" height="350px">
      </iframe> == $0
    </div>
    <div id="frame2Wrapper" class="mt-4">
      <iframe src="/sample" id="frame2" width="100px" height="100px">
        </iframe>
      </div>
    </div>
  </div>
</div>
```


Frame 1

Örneğin sayfada 100 iframe var ise index kullanarak Selenium'da iframe geçebiliriz.


iFrame



switchTo.frame (string frame Name Or Id): switchTo komutuna Name veya ID iletilip driver'ın o iframe geçmesini sağlanabilir.



```
<div>...</div>
  <div id="frame1Wrapper">
    <iframe src="/sample" id="frame1" width="500px" height="350px"> == $0
    #document
  </iframe>
</div>
```



```
<iframe id="a077aa5e" name="a077aa5e" src="ads.html" width="750
px;" height="110px;" scrolling="no">
  #document
</html>
```

```
driver.switchTo().frame("iframe1");
driver.switchTo().frame("a077aa5e");
```

iFrame



switchTo.frame (WebElement frame Element): WebElement kullanarak geçiş yapmak en esnek seçenektir. Tercih edilen locator ile WebElement kullanarak frame bulunabilir ve ona geçiş yapılabilir.

```
//Store the web element
WebElement iframe = driver.findElement(By.cssSelector("#modal>iframe"));

//Switch to the frame
driver.switchTo().frame(iframe);

//Now we can click the button
driver.findElement(By.tagName("button")).click();
```



```
driver.get("https://demoqa.com/frames");
//First find the element using any of locator strategy
WebElement iframeElement = driver.findElement(By.id("IF1"));

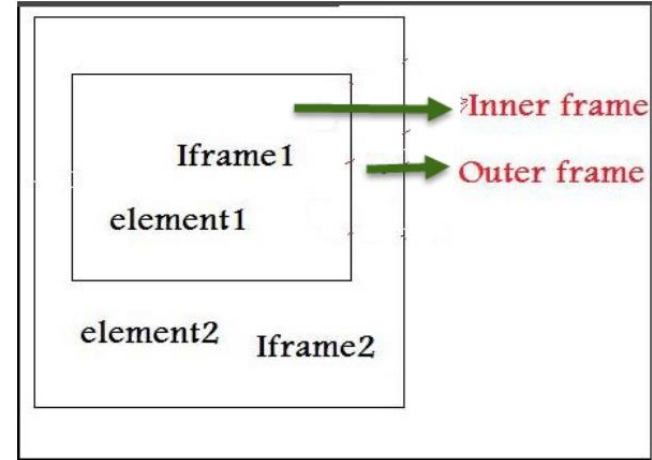
//now use the switch command
driver.switchTo().frame(iframeElement);
```

iFrame



iFrame'den çıkmak için 2 komut vardır.

- **`driver.switchTo().parentFrame();`**
1 üst seviyedeki frame çıkartır.
- **`driver.switchTo().defaultContent();`**
En üstteki frame çıkmak için kullanılır.



NOT:

Birden fazla iframe varsa geçişlerde dikkatli gerekir. Çünkü geçişler her zaman basit olamayabilir. İç içe çerçeveler durumunda,

- İlk başta, iframe'in Index veya ID'sine göre dış çerçeveye geçmeliyiz.
- Dış çerçeveye geçtiğimizde, dış çerçeve içindeki toplam iframe sayısını bulabiliriz.
- Ve bilinen yöntemlerden herhangi biri ile iç çerçeveye geçebiliriz.

iFrame



Number of iframe on a page

iFrame'ler iframe etiketi kullanılarak oluşturulur.

findElements kullanılabilir ve iframe etiketli tüm öğeleri liste olarak döndürebilir.

Listedeki iframe sayısını bulmak için size yöntemi şu şekildedir,

```
List<WebElement> iframeElements=driver.findElements(By.xpath("//iframe"));  
int iframeSize=iframeElements.size();  
System.out.println(iframeSize);
```

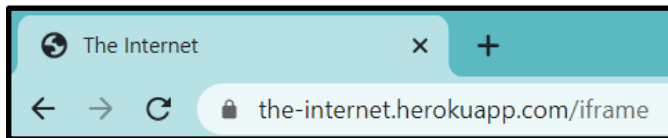
Alternatif olarak, findElements'i xpath/ tagName ile de kullanılabilir.

```
driver.findElements(By.xpath("//iframe")).size();  
driver.findElements(By.tagName("iframe")).size();
```



TIME FOR TASK

iFrame



Go to URL: <https://the-internet.herokuapp.com/iframe>

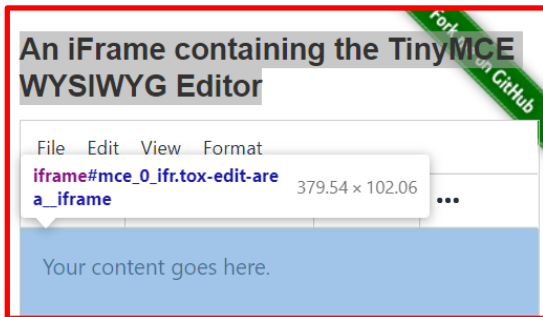
Verify the Bolded text contains “Editor”

Locate the text box

Delete the text in the text box

Type “Hi everyone”

Verify the text Elemental Selenium text is displayed on the page.



```
><iframe id="mce_0_ifr" frameborder="0"
allowtransparency="true" title="Rich Text Area"
class="tox-edit-area_iframe" cd_frame_id="bc8fc3f
e58364e260c6864082fb4fef2">...</iframe> == $0
```



iFrame

TIME FOR TASK

<https://www.jqueryscript.net/demo/Easy-iFrame-based-Twitter-Emoji-Picker-Plugin-jQuery-Emoojis/> sitesine gidiniz.
Web sitesini maximize yapınız.
İkinci emojiye tıklayınız.
Tüm ikinci emoji öğelerine tıklayınız.
Parent iframe geri dönünüz.
Formu doldurun (Formu istediğiniz metinlerle doldurun) apply button'a basınız.



Personalize here:

<small>Text...</small> Bir	<small>Name of smiles...</small> iframe
<small>Name of nature...</small> sorusu	<small>Name of food...</small> bu
<small>Name of activities...</small> kadar	<small>Name of places...</small> eglençeli
<small>Name of objects...</small> olabilir	<small>Name of symbols...</small> sizde de
<small>Name of flags...</small> oyde değil mi ?	<small>Hide emojis...</small> Enter tabs to hide separated by commas

Active tab...
Default: smiles

APPLY



iFrame



Go to URL: <http://demo.guru99.com/test/guru99home/>

Find the number of iframes on the page.

Link to the fourth iframe (JMeter Made Easy) (<https://www.guru99.com/live-selenium-project.html>) click here.

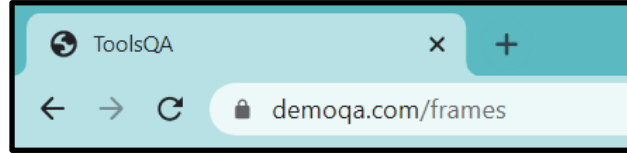
Exit the iframe and return to the main page.

```
<iframe id="a077aa5e" name="a077aa5e" src="ads.html" width="750px;" height="110px;"
scrolling="no"> == $0
#document
  <html>
    <head>...</head>
    <a href="http://www.guru99.com/live-selenium-project.html">
  </a>
</body>
</html>
</iframe>
```




Free Work

TIME FOR TASK



<https://demoqa.com/frames>

Bu sayfada pratik yapabilirsiniz.

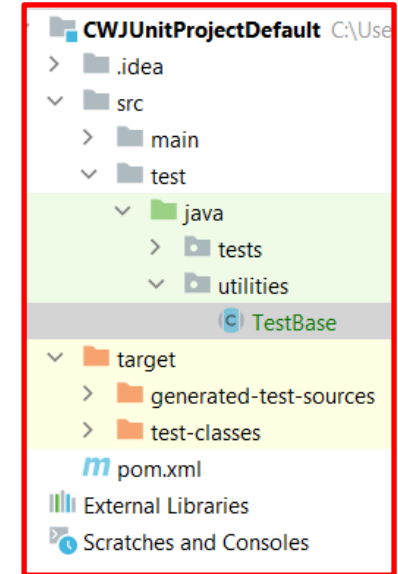
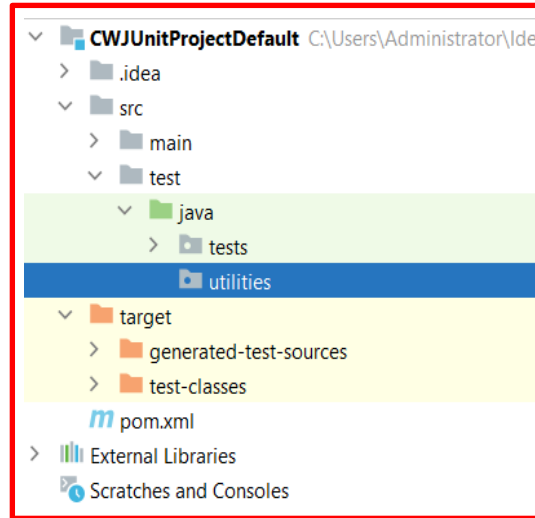


Test Base



Framework'te, utilities [-araçlar-yardımcı program] package'si vardır.
Bu package'de,

- ✓ TestBase
- ✓ Reusable Methods
- ✓ Driver
- ✓ ExcelUtils
- ✓ DatabaseUtils
- ✓ JsUtils vb. olabilir.



Test Base



TestBase Class: Her test class'ında tekrar tekrar yazdığımız setup ve teardown metotları tekrar yazmaktan kurtulmak için Java OOP konsepti kullanarak oluşturduğumuz bir class'dır.

Test Base parent olarak kabul eden class'lar hazırlık için oluşturduğumuz driver, setup() ve teardown() oluşturmak zorunda kalmazlar, kendilerinde olmasa bile parent class'dan kullanılmış olur.

TestBase genelde testlerden farklı bir package'da olduğundan driver'ın kullanabilmesi için **public** veya **protected** yapılması gerekir. Biz hep child class'lardan kullanacağımız için **protected** yapmayı yapmayı tercih ediyoruz.

TestBase, Test classlarını **temiz tutmamıza yardımcı** olur.

```
package utilities;
import io.github.bonigarcia.wdm.WebDriverManager;
import org.junit.After;
import org.junit.Before;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import java.time.Duration;

public abstract class TestBase {

    protected WebDriver driver;

    @Before
    public void setUp(){
        WebDriverManager.chromedriver().setup();
        driver= new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(15));
    }

    @After
    public void tearDown() { driver.quit(); }
}
```

TestBase sınıfı **abstract** olmalıdır, böylece kullanıldığında sadece **extends** edilmesi yeterlidir ve **bu classtan obje** üretilmesi engellenir.



THANKS!

Any questions?

Elly D. - Full-Stack Automation Engineer

Garry T. - Full-Stack Automation Engineer

