



Selenium giriş

SELENIUM GIRIS
OTOMASYON TEST NEDİR?
SELENIUM PROJESİ OLUSTURMA



SELENIUM NEDİR

- Selenium, web uygulamalarını farklı tarayıcılarda ve platformlarda test etmek için ücretsiz (açık kaynaklı) bir araçtır.
- Selenium yalnızca web tabanlı uygulamaları otomasyon yapmaya odaklanır. Mobil ve Windows testi yapmak için eklentiler selenium'a eklenebilir.
- Selenium jar dosyaları ile kurulabilir. Kurulum sırasında jar dosyalarını gördünüz.

Selenium automates browsers. That's it!
What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that.

Boring web-based administration tasks can (and should) also be automated as well.

About Selenium

Selenium is a suite of tools for automating web browsers.

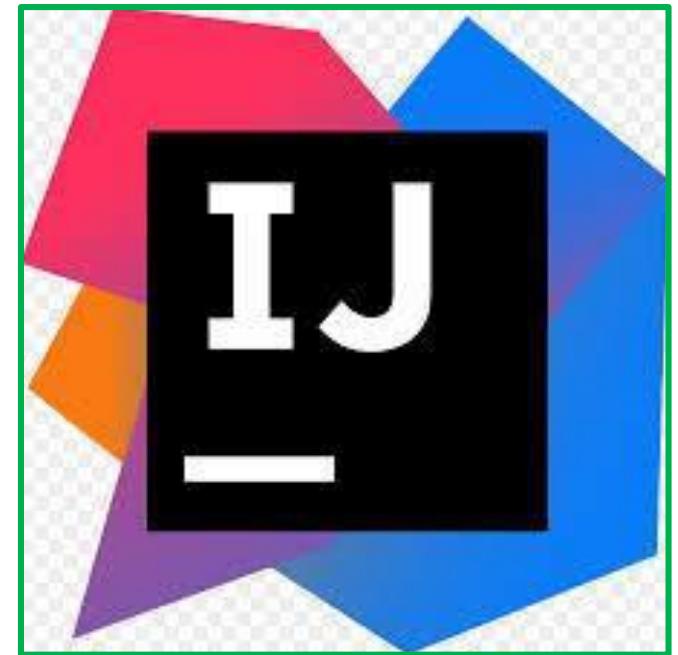
- Selenium, otomasyon yapmak için kendi sınıflarına ve yöntemlerine sahip bir suite'dir..

- 2021'de piyasaya sürülen Selenium 4'ü öğreneceğiz.
- Selenium'u çeşitli programlama dilleri ile yazabilirsiniz java, Python, ruby, .Net vb.



IntelliJ IDEA NEDIR

- IntelliJ IDEA 2000 yılında kurulmuş olan JetBrains isimli yazılım firmasına ait olan, popüler bir Java editörüdür.. (ide)
- IntelliJ IDEA'nın her yönü, geliştirici üretkenliğini en üst düzeye çıkarmak için tasarlanmıştır.
- Akıllı kodlama yardımcı ve ergonomik tasarım ile, kod yazınızı yalnızca verimli değil, aynı zamanda keyifli hale getirir.
- Banyak framework ve plugin ile çalışma imkanı verir
- Akıllı tamamlama özelliği ile kod yazınızı oldukça kolaylaştırır
- IDE ihtiyaçlarınızı tahmin eder ve sıkıcı ve tekrarlayan geliştirme görevlerini otomatikleştirir, böylece büyük resme odaklanabilirsiniz.





SOFTWARE TESTING NEDİR ?

EXPECTED RESULT (beklenen sonucun), **ACTUAL RESULT** (gerçek sonuca) esit olup olmadigini kontrol etme islemidir.

- Eger Expected result = Actual result, ise status PASSED (test basarili)
- Eger Expected result !=Actual result, ise status FAIL (test basarisiz)

Sonuç olarak, olması gereken şeylerin olmadığını veya olmaması gereken şeylerin olduğunu kontrol etmek ve ortaya çıkartmak yazılım testinin amacı olmalıdır.



- Her User Story icin Positive ve Negative Test(ler) yapılmalıdır
- Test, musteri/isletme ihtiyaçlarını karşılamak için yapılır.
- Bir uygulamayı test etmek için onceden belirlenmis user storyler (kullanıcı hikayeleri) ve tanımlanmış acceptance criterias (kabul kriterleri) dikkate alınır.

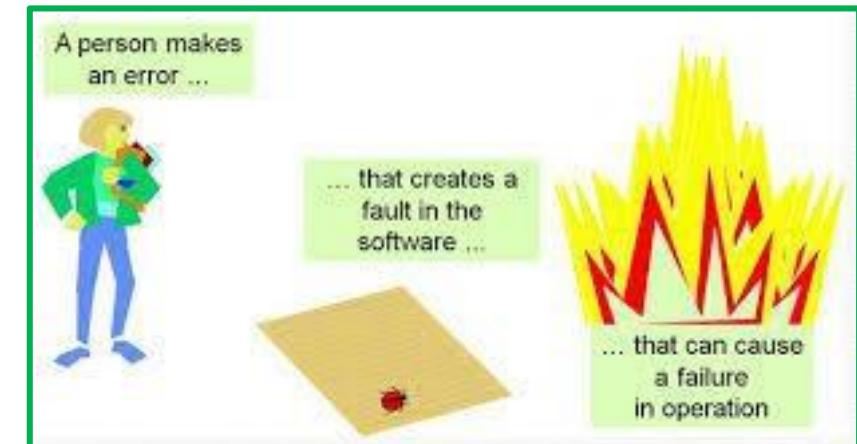


SOFTWARE TESTING NEDEN ONEMLIDIR ?

- İnsanlar hata yaparlar, bu hatalar kodda, yazılımda, sistemde ya da dokümanda defect (Kusur) oluşturur.
- Defect olan kod çalıştırıldığında sistem beklenen fonksiyonları gerçekleştiremez ve başarısız olur.

Bu sebeplerden dolayı;

- Müşteriye sunulmadan önce ürün kalitesinden emin olmak,
- Yeniden çalışma (düzeltme) ve geliştirme masraflarını azaltmak,
- Geliştirme işleminin erken aşamalarında yanlışları saptayarak ileri aşamalara yayılmasını önlemek, böylece zaman ve maliyetten tasarruf sağlamak amacıyla ürün müşteriye sunulmadan önce test edilmesi gerekmektedir.



KISACA : Testing saves money and lives .

Testing para kayiplarini onler ve hayat kurtarir. (Ucak kazalari vs..)



MANUAL(FUNCTIONAL) TESTING NEDİR?

- Manuel test, uygulamayı herhangi bir otomasyon aracı olmadan manuel olarak test etmektir.
- Manuel test kullanıcıları dokümantasyon için sınırlı teknoloji (Excel vb.) kullanır, ancak otomasyon araçları veya dili kullanmazlar.
- Manuel testte insan hatası olabilir.
- Tüm Otomasyon Tester'lar, herhangi bir otomasyon yapmadan önce uygulamayı anlamak için mutlaka manuel test yapmalıdır.
- İyi bir otomasyon tester aynı zamanda iyi bir manuel testerdir.





TEST OTOMATION NEDİR ?



- Bir sistemi bir otomasyon aracı (tool) yardımıyla test etmeye 'Test Otomasyonu' denir.
- Otomasyon test yazılımı test verilerini Test Edilen Sistem'e girebilir, beklenen ve gerçek sonuçları karşılaştırabilir ve ayrıntılı test raporları oluşturabilir.
- Bir test otomasyon tool'u kullanarak, calistirilan test paketini kaydetmek ve gerekiğinde yeniden calistirmak mümkündür. Test paketi otomatik hale getirildikten sonra hiçbir insan müdahalesi gerekmesizin programlandigi zamanda calisabilir.
- Giderek daha popüler hale gelmektedir.



MANUAL TESTING vs AUTOMATION TESTING

Asagida yazili metin sizce neyi ifade etmektedir ?

- A. Test Case
- B. Manuel tester icin test adimlari
- C. Otomasyon ile test yapan kodlar

@hotel

Scenario: TC08 Positive login

```
Given kullanici "HMCUrl" sayfasina gider
And login linkine tiklar
Then kullanici adi olarak "HMCValidUsername" girer
And password olarak "HMCValidPassword" girer
And Login butonuna basar
Then basarili olarak giris yapildigini test eder
And sayfayı kapatır
```



MANUAL TESTING vs AUTOMATION TESTING

MANUAL TESTING

VS

AUTOMATION TESTING

EXECUTION TIME



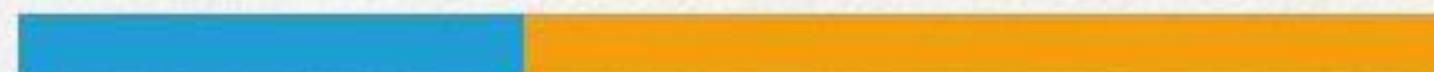
PEOPLE



INFRESTRUCTURE



TOOLS



TURNAROUND TIME

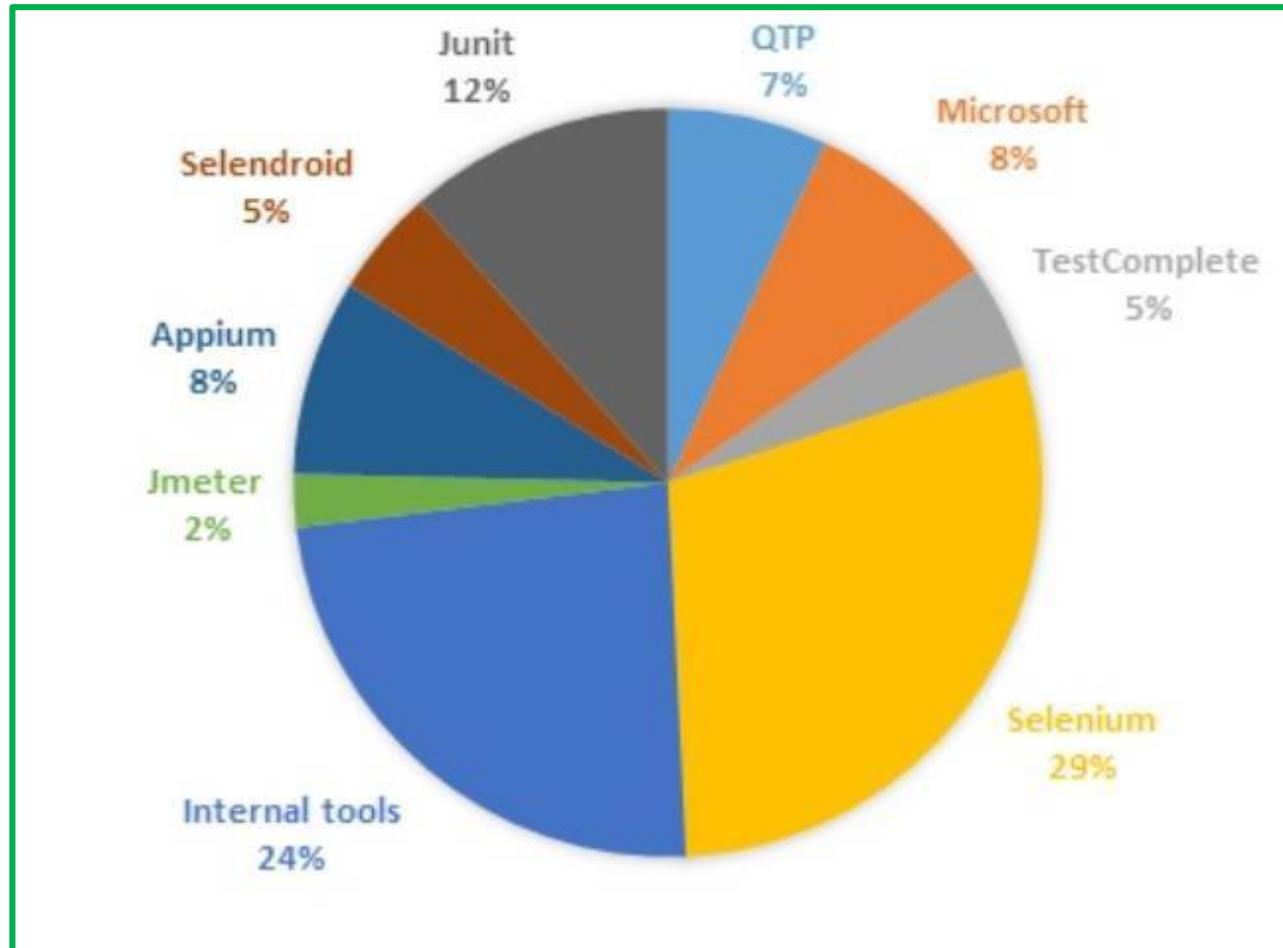


TRAINING





EN COK KULLANILAN TEST TOOL'LARI





Basarılı Bir Otomasyon Testi için Yapılması Gerekenler

- Doğru otomasyon araçlarını(tools) seçin. Genellikle şirketlerin kullandıkları tool'lari vardır, bazi şirketler de sizin isteginize gore tool secebılırler.
- Uygulamanız hakkında iyi bilgi sahibi olun.
- Test senaryolarınızı (test cases) kısa ve bağımsız tutun.
- Test otomasyonlarınızı önem derecesine göre sıralayın.
- Otomasyondan önce test verilerini hazırlayın(id,url,environment)
- Gerekirse, test caselerinizi yönetin(manage) ve bakımını yapın(maintain).Yeni test caselerinizin eski otomasyon test scriptlerini bozmadiginden emin olun.
- Testlerinizde hata olabilecegini goz onunde bulundurun. Bir BUG'i rapor etmeden once mutlaka testinizi gozden gecirin
- Her zaman test ekibinizle, özellikle team liderleri ile iyi iletisim kurun.





SELENIUM BILESENLERİ

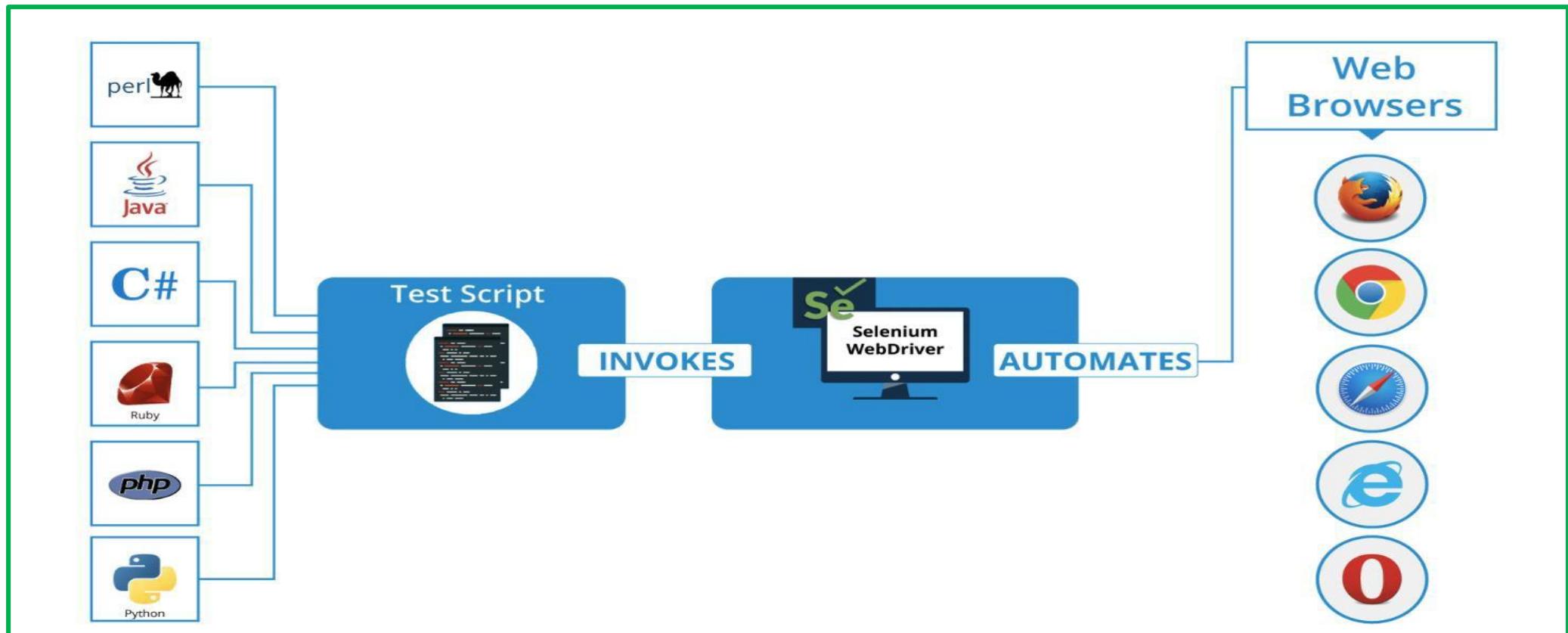


Selenium'un dört bileşeni vardır;

- Selenium Integrated Development Environment (IDE) (Selenyum Entegre Geliştirme Ortamı (IDE))
- Selenium Remote Control (RC)(Selenyum Uzaktan Kumanda (RC))
- WebDriver (Biz Selenium WebDriver kullanacağız)
- Selenium Grid (paralel test için kullanılıyor)



SELENIUM NASIL CALISIR?





SELENIUM'UN AVANTAJLARI NELERDIR?

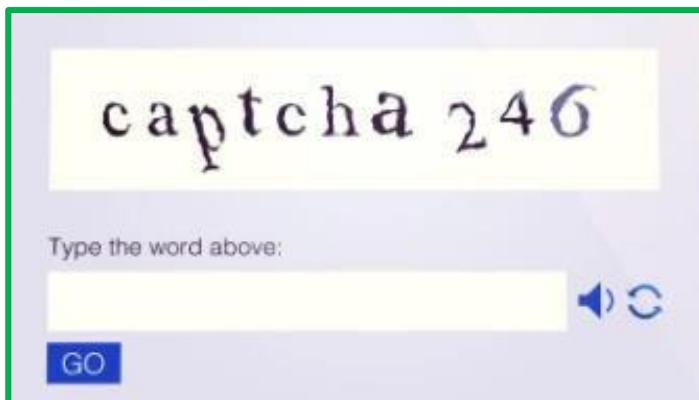


- 1) Ücretsiz ve acik kaynaklidir. (Open source)
- 2) Bir çok programlama dilini destekler (Java, Python, PHP, C#, Ruby vs.)
- 3) Çoklu işletim sistemleriyle çalışır.
Multiple operating systems
(Windows, MacOS, Linux)
- 4) Birden çok tarayıcı ile çalışır.
Multiple browsers
(IE, Safari, Chrome, Firefox vs.)



SELENIUM'UN DEZAVANTAJLARI NELERDIR?

- Programlama bilgisi gerektirir
(Biz Java biliyoruz)
- Yalnızca web tabanlı uygulamaları test eder
- Profesyonel desteği sahip değil
(Ama geniş bir kullanıcı kitlesi var)



SELENIUM'UN YAPAMADIKLARI

- performans testi
- handle captcha
(diğer tüm otomasyon araçları gibi)



TEST AUTOMATION FRAMEWORK NEDIR?

- Test Otomasyon Framework (TAF), yazılım projelerindeki Test Case'lerin oluşturulma ve gözden geçirme süresini azaltmak ve QA ekibimizin isini kolaylaştırmak için tasarlanmış yapılardır.

(İşinizi rahat yapabilmeniz için hazırlanmış çalışma ortamınızdır)

- Kullanılan Tool'lar

Java,
Eclipse/IntelliJ,
Selenium,
JUnit/
TestNG,
Cucumber

- Test Yapılan Layer'lar
UI , API , DataBase

```
my.mavenproject [~/ideaProjects/my.mavenproject] - ~/IdeaProjects/my.mavenproject
Project my.mavenproject ~/IdeaProjects/my.mavenproject
  .idea
  src
    main
      java
      resources
    test
      java
        com.techproed
          CheckboxAndRadioBut
          DropDownAmazon
          DropDownHomework
          FirstMavenClass
          FirstTestNgClass
          TestNGAnnotationsExar
          TestNGAnnotations
          WebTables
      target
        classes
        generated-test-sources
        test-classes
      pom.xml
  External Libraries
  Scratches and Consoles
```

TestNGAnnotationsExample



SELENIUM ve CHROME DRIVER KURULUMU

- 1) <https://www.selenium.dev/downloads/> adresine gidelim
- 2) Selenium Client & WebDriver Language Bindings altında Java driver'ini download edin
- 3) Browsers altında Chrome documentation linkini tiklayalim

Chrome'un kendi sayfasina gidip Current stable release'i tiklayip size uygun olani download edin
***** buradaki surum ile bilgisayarinizdaki Chrome surumunun ayni oldugundan emin olun**
- 4) src altında resources director'si olusturun
- 5) Bu klasor altında drivers ve libraries klasorleri olusturun
- 6) Indirdigimiz chromedriver'i drivers klasorune, selenium-java dosyasini ise libraries klasorune cikartalim
- 7) intelliJ 'de yeni project / package / class olusturalim ve class icinde main method olusturalim
- 8) File/Project Structure/Modules/Dependencies kismindan jar dosyalarini yukleyelim



WebDriver Olusturulmasi

1. Yeni bir class oluşturun: class name ⇒ C01_Get
2. main method oluşturun
3. Java'dan System.setProperty("", "") method'unu kullanarak webdriver'in turunu ve path'ini belirleyelim.

```
System.setProperty("webdriver.chrome.driver", "src/driver/chromedriver"); /MAC
```

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

4. Chrome driver oluşturun

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

5. Olusturdugumuz driver objesi ile WebDriver class'indan static method'lari kullanin



driver.get(); Method'lari

driver.get(String Url); String olarak girilen Url'e gider

driver.getTitle(); Icinde olunan sayfanin basligini String olarak getirir

driver.getCurrentUrl(); Icinde olunan sayfanin Url'ini String olarak getirir

driver.getPageSource(); Icinde olunan sayfanin kaynak kodlarini String olarak getirir

driver.getWindowHandle(); Icinde olunan sayfa ve/veya tab'larin
driver.getWindowHandles(); handle degerlerini getirir



driver.get(); Method'lari

Class Work

1. Yeni bir package olusturalim : day01
2. Yeni bir class olusturalim : C03_GetMethods
3. Amazon sayfasina gidelim. <https://www.amazon.com/>
4. Sayfa basligini(title) yazdirin
5. Sayfa basliginin "Amazon" icerdigini test edin
6. Sayfa adresini(url) yazdirin
7. Sayfa url'inin "amazon" icerdigini test edin.
8. Sayfa handle degerini yazdirin
9. Sayfa HTML kodlarinda "Gateway" kelimesi gectigini test edin
10. Sayfayı kapatın.



driver.navigate(); Method'lari

driver.navigate().to(String Url); String olarak girilen Url'e gider

driver.navigate().back(); Icinde olunan sayfadan, geldigi onceki sayfaya dondurur

driver.navigate().forward(); Back ile donulen bir sayfadan tekrar ileri gider

driver.navigate().refresh(); Icinde olunan sayfayı yeniler.



Selenium Navigation Methods

Class Work

1. Yeni bir Class olusturalim.C05_NavigationMethods
2. Youtube ana sayfasina gidelim . <https://www.youtube.com/>
3. Amazon soyfasina gidelim. <https://www.amazon.com/>
4. Tekrar YouTube'sayfasina donelim
5. Yeniden Amazon sayfasina gidelim
6. Sayfayı Refresh(yenile) yapalim
7. Sayfayı kapatalim / Tüm sayfaları kapatalim



driver.manage().window(); Method'lari

driver.manage().window().maximize(); Browser'i maximize yapar

driver.manage().window().minimize(); Browser'i minimize yapar

driver.manage().window().fullscreen(); Browser'i full screen yapar

driver.manage().window().getPosition(); Browser'in koordinatlarini verir

driver.manage().window().getSize(); Browser'in olculerini verir



Selenium Navigation Methods

Class Work

1. Yeni bir Class olusturalim.C06_ManageWindow
2. Amazon soyfasina gidelim. <https://www.amazon.com/>
3. Sayfanin konumunu ve boyutlarini yazdirin
4. Sayfayı simge durumuna getirin
5. simge durumunda 3 saniye bekleyip sayfayı maximize yapın
6. Sayfanin konumunu ve boyutlarini maximize durumunda yazdirin
7. Sayfayı fullscreen yapın
8. Sayfanin konumunu ve boyutlarini fullscreen durumunda yazdirin
9. Sayfayı kapatın



driver.manage().Method'lari

```
driver.manage().window().setPosition(new Point( x:80 , y:0 ) );
```

Browser'i istenen koordinata tasir

```
driver.manage().window().setSize(new Dimension( 800 , 600 ));
```

Browser'i istenen olculere getirir.

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

Istenen sayfa acilincaya kadar maximum bekleme suresini belirtir

```
driver.close(); sadece calisilan browser'i kapatir
```

```
driver.quit(); acik olan tum browser'lari kapatir
```



Selenium Navigation Methods

Class Work

1. Yeni bir Class olusturalim.C07_ManageWindowSet
2. Amazon soyfasina gidelim. <https://www.amazon.com/>
3. Sayfanin konumunu ve boyutlarini yazdirin
4. Sayfanin konumunu ve boyutunu istediginiz sekilde ayarlayın
5. Sayfanin sizin istediginiz konum ve boyuta geldigini test edin
8. Sayfayı kapatın



Homework

- 1.Yeni bir class olusturalim (Homework)
- 2.ChromeDriver kullanarak, facebook sayfasina gidin ve sayfa basliginin (title) "facebook" oldugunu dogrulayin (verify), degilse dogru basligi yazdirin.
- 3.Sayfa URL'inin "facebook" kelimesi icerdigini dogrulayin, icermiyorsa "actual" URL'i yazdirin.
- 4.<https://www.walmart.com/> sayfasina gidin.
5. Sayfa basliginin "Walmart.com" icerdigini dogrulayin.
6. Tekrar "facebook" sayfasina donun
7. Sayfayı yenileyin
- 8.Sayfayı tam sayfa (maximize) yapın
- 9.Browser'i kapatın



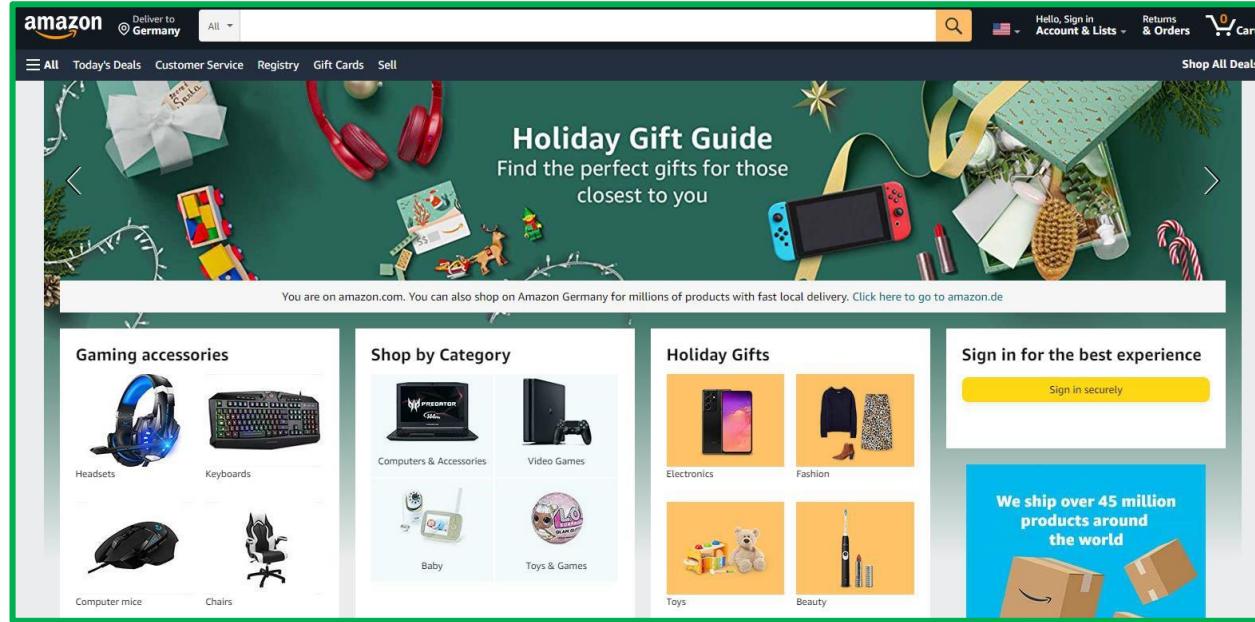
Tekrar Testi

1. Yeni bir class olusturun (TekrarTesti)
2. Youtube web sayfasına gidin ve sayfa başlığının "youtube" olup olmadığını doğrulayın (verify), eğer değilse doğru başlığı(Actual Title) konsolda yazdırın.
3. Sayfa URL'sinin "youtube" içerip içermediğini (contains) doğrulayın, içermiyorsa doğru URL'yi yazdırın.
4. Daha sonra Amazon sayfasına gidin <https://www.amazon.com/>
5. Youtube sayfasına geri donun
6. Sayfayı yenileyin
7. Amazon sayfasına donun
8. Sayfayı tamsayfa yapın
9. Ardından sayfa başlığının "Amazon" içerip içermediğini (contains) doğrulayın, Yoksa doğru başlığı(Actual Title) yazdırın.
10. Sayfa URL'sinin <https://www.amazon.com/> olup olmadığını doğrulayın, degilse doğru URL'yi yazdırın
11. Sayfayı kapatın



WEBELEMENTS

- Web sayfasında kullanılan etkilesimli olan veya olmayan herseye webelement denir



- Button,
- Search box(arama kutusu),
- Text box(metin kutusu),
- Headers(başlıklar),
- Tables(tablolar) vb...

- Farklı türde WebElement tag'ları(etiketleri) vardır.

<html>,<body>,<form>,<label>,<input>,<a> vb.

- Otomasyon için unique(tek) web öğelerini(element) tanımlamak üzere HTML kodunu inceleyeceğiz(inspect).
- Web elementleri birlikte kullanıcı arayüzünde (UI) bir web sayfası oluştururlar.



WEBELEMENTS

```
▼<tbody>
  ►<tr>...</tr>
  ▼<tr> == $0
    ▼<td>
      <input type="email" class="inputtext login_form_input_box" name="email" id="email" data-testid=
        "royal_email">
    </td>
    ▼<td>
      <input type="password" class="inputtext login_form_input_box" name="pass" id="pass" data-testid=
        "royal_pass">
    </td>
    ▼<td>
      ►<label class="login_form_login_button uiButton uiButtonConfirm" id="loginbutton" for="u_0_b">...
      </label>
    </td>
  </tr>
  ►<tr>...</tr>
</tbody>
```



Web Sayfalarını İnceleme (Inspect)

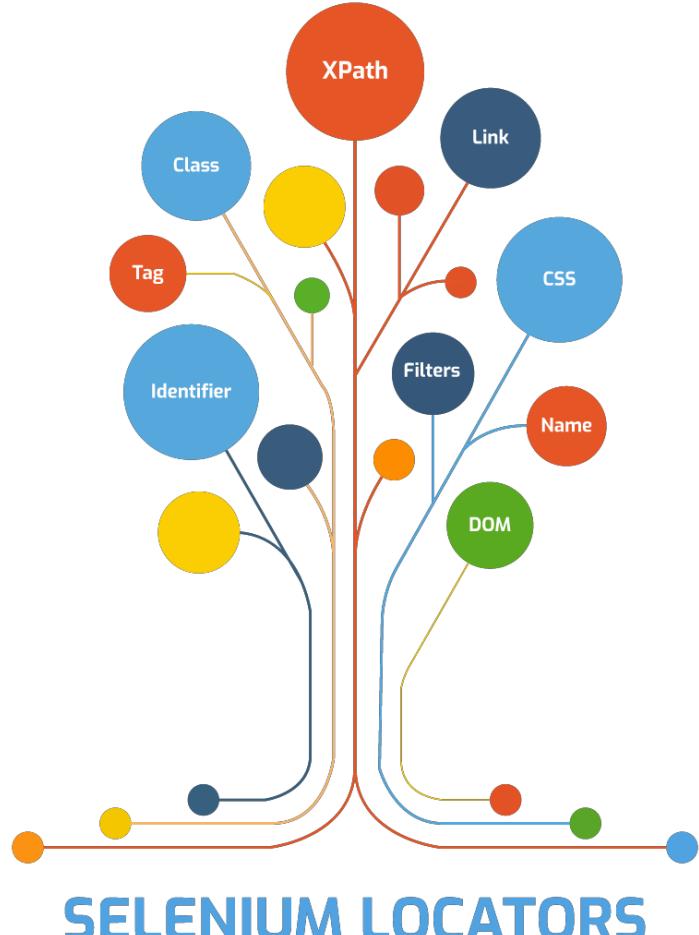
The screenshot shows a web browser displaying the Amazon homepage. A context menu is open over a pair of black earbuds, with the 'Inspect' option highlighted by a red oval. The browser's developer tools are visible at the bottom, showing the HTML structure of the page, including the search bar and its input field.

```
<div class="nav-search">
  <div id="nav-bar-left"></div>
  <form accept-charset="utf-8" action="/s/ref=nb_sb_noss" class="nav-searchbar" method="GET" name="site-search" role="search">
    <div class="nav-left">...</div>
    <div class="nav-right">...</div>
  </div>
<div class="nav-search-field ">
  <label id="nav-search-label" for="twotabsearchtextbox" class="aok-offscreen">
    Search
  </label>
  <input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="Search" class="nav-input" dir="auto" tabindex="19"> == $0
</div>
<div id="nav-iss-attach"></div>
```



LOCATORS (YER BULUCU-KONUM BELIRLEYICI)

- ❖ Selenium LOCATORS, web sayfasındaki web öğelerini tanımlamak için kullanılır.
- ❖ Selenium'da; metin kutuları, onay kutuları, linkler, radyo butonları, liste kutuları ve **diğer tüm web öğeler** üzerinde eylemler gerçekleştirmek için LOCATORS'a ihtiyacımız vardır.
- ❖ Konum belirleyiciler bize nesneleri tanımlamada yardımcı olur.
- ❖ Web Elementlerine ulaşmak için tag veya bazı attribute'ler kullanılır, bunlarla ulaşamayan webelementleri için özel olarak tanımlanan **Xpath** ve css locator'lari kullanılır.





LOCATORS (YER BULUCU-KONUM BELIRLEYICI)

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute" dir="auto" tabindex="0" aria-label="Search">
```

Bir web elementini tanımlamak için 8 tane selenium locator vardır.

1. id
2. name
3. className
4. tagName
5. linkText
6. partialLinkText
7. **xpath** => xpath yazmanın birden fazla yolu vardır
8. **cssSelector** => css yazmanın birden fazla yolu vardır



LOCATORS BULMAK ICIN KULLANILAN METHOD'LAR

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute" dir="auto" tabindex="0" aria-label="Search">
```

- 1) Tanimladigimiz web elementin yerini driver'in bulabilmesi icin findElement (Locator) method'unu kullaniriz.

```
driver.findElement(Locator);
```

- 2) findElement (Locator) method'unun icine parameter olarak yazacagimiz Locator'in 8 locator'dan hangisi oldugunu belirtmek icin de By.LocatorTuru("LocateBilgisi"); kullanilir

```
driver.findElement(By.id("twotabsearchtextbox"));
```

- 3) Locate ettigimiz Web Elementini kullanabilmek icin bir variable'a atama yapariz

```
WebElement aramaKutusu = driver.findElement(By.id("twotabsearchtextbox));
```



LOCATORS

1) By.id();

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute" dir="auto" tabindex="0" aria-label="Search">
```

```
WebElement aramaKutusu = driver.findElement(By.id ("twotabsearchtextbox"));
```

- Web öğesini tanımlamanın en popüler yolu id kullanmaktır.
- id en güvenli ve en hızlı locator seçeneği olarak kabul edilir ve her zaman birden çok locator arasında ilk öncelik olmalıdır.
- Eğer yanlış id locate edilirse; **NoSuchElementException** hatası oluşur.

*** **NoSuchElementException** gordugumuzde hata veren satirdaki locator gozden gecirilmelidir



LOCATORS

2) By.name();

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute" dir="auto" tabindex="0" aria-label="Search">
```

```
WebElement passwordTextBox =driver.findElement(By.name("field-keywords"));
```

- ❖ Name ve value unique ise bu metodu da kullanabilirsiniz.



LOCATORS

3) By.className();

```
<input class="form-control" placeholder="Password" data-test="password"
type="password" name="session[password]" id="session_password">
```

```
WebElement passwordTextBox =driver.findElement(By.className("form-control"));
```

- ❖ Class attribute'u olduğunda kullanılabilir.
- ❖ Class ve value unique ise, bu metodu da kullanabilirsiniz, ancak genelde class attribute aynı islevi yapan bir grup Web Elementi için kullanılır
- ❖ Class attribute'nun değeri boşluk içeriyorsa genelde By.className() ile yapılan locator'lar başarılı çalışmaz



LOCATORS

4) By.linkText();

```
<a class="nav-item nav-link" data-test="addresses" href="/addresses">Addresses</a>
```

```
WebElement passwordTextBox = driver.findElement(By.linkText("Addresses"));
```

- ❖ Bu yalnızca HTML bağlantılarını(link) tanımlamak için kullanılabilir.
- ❖ HTML link elementleri, bir web sayfasında bağlantı etiketi(tag) kısaltması olan <a> etiketi(tag) kullanılarak temsil edilir.
- ❖ Kullanıcı arayüzündeki(UI) hyperlinkleri kolayca tanıyalabilir ve sonra bu yöntemini kullanabilirsiniz
- ❖ Büyük / küçük harfe duyarlıdır (case sensitive) ve bağlantı(link) metniyle eşleşmelidir



LOCATORS

5) By.partialLinkText();

```
<a class="nav-item nav-link" data-test="addresses" href="/addresses">Addresses</a>
```

```
WebElement passwordTextBox = driver.findElement(By.partialLinkText("dresses"));
```

- ❖ linkText () yöntemine benzer.
- ❖ Tek fark, tam metin vermek zorunda kalmamanızdır.
- ❖ Metnin yalnızca belirli bir bölümünü verebilirsiniz.
- ❖ Metnin tamamını verdığınızda de kabul eder.



LOCATORS

6) By.tagName();

```
<input class="form-control" placeholder="Password" data-test="password"
type="password" name="session[password]" id="session_password">
```

```
WebElement passwordTextBox =driver.findElement(By.tagName("input"));
```

- ❖ Bu, diğer konum belirleyicilerden biraz farklıdır.
- ❖ <div>, <a>, <input>, ... gibi belirli bir etiketi ilettiğinizde, birden fazla aynı ad etiketine sahip olabileceğiniz için birden çok öğeyi döndürür.
Çoğunlukla öğelerin bir listesini almak için kullanılır. Bu nedenle findElements() yöntemiyle kullanılması önerilir.
- Örneğin, kullandığımız bir sayfadaki tüm linkleri döndürmek için **By.tagName("a")** kullanılabilir



findElement() Method

```
WebElement elementName=driver.findElement(By.LocatorStrategy("LocatorValue"));
```

- ❖ Driver'in bir elementi bulması için findElement() yöntemini kullanırız.
- ❖ Bu, tek bir web elementini döndürür. Aynı locator ile ulaşılabilen birden fazla web element varsa ilkini dondurmur
- ❖ Driver elementi bulamazsa, runtime exception verir : **NoSuchElementException**.
- ❖ **NoSuchElementException**'ı gördüğünüzde, locatorı tekrar kontrol etmelisiniz.



findElements() Method

```
List<WebElement> elementName=driver.findElements(By.LocatorStrategy("LocatorValue"));
```

- ❖ Locator degerine uygun Web elementlerinin listesini döndürür
- ❖ Locator stratejisiyle eşleşen web elementi yoksa boş bir liste döndürür.
- ❖ **NoSuchElementException** hatası vermez.
- ❖ Listedeki her Web elementi, 0'dan başlayan bir indeks alır.



findElements() ile findElement() arasindaki farklar

	findElement()	findElements()
websayfasinda birden fazla Web Element Locator ile uyuşursa	Ilk elemani dondurur	Tum elemanlari ondurur
websayfasinda hicbir Web Element Locator ile uyusmazsa	NoSuchElementException fırlatir	Exception fırlatmaz, bos bir liste dondurur
Return Type	WebElement	List<WebElement>
Elemana erisim	Direk ulasilabilir	Liste'den index veya iterator ile ulasilabilir



WebElement Method'lari

Bir WebElement üzerinde eylemler gerçekleştirmek otomasyon tester'ları için çok önemlidir.

`webElement.click();` Web Element'e click yapar

`webElement.sendKeys("Metin");` Parametre olarak yazılan metni
Web Elemente gönderir

`webElement.submit();` Web element ile işlem yaparken
Enter tusuna basma görevi yapar

`webElement.sendKeys("Metin" + Keys.ENTER);` İstedigimiz metni yollayıp, sonra ENTER'a basar.



WebElement Method'lari

Class Work: Login Test

1. Bir class oluşturun: LoginTest
2. Main method oluşturun ve aşağıdaki görevi tamamlayın.
 - a. <http://a.testaddressbook.com> adresine gidiniz.
 - b. Sign in butonuna basın
 - c. email textbox,password textbox, and signin button elementlerini locate ediniz..
 - d. Kullanıcı adını ve şifreyi aşağıya girin ve oturum aç (sign in)buttonunu tıklayın:
 - i. Username : testtechproed@gmail.com
 - ii. Password : Test1234!
 - e. Expected user id nin testtechproed@gmail.com olduğunu doğrulayın(verify).
 - f. "Addresses" ve "Sign Out" textlerinin görüntülendiğini(displayed) doğrulayın(verify).
3. Sayfada kaç tane link olduğunu bulun.



WebElement Get Method'lari

`webElement.getText();` Web Element üzerindeki yazıyı getirir

`webElement.getAttribute("Att.ismi");` İsmi girilen attribute'un
değerini getirir

`webElement.getTagName();` Web elementin tag ismini
getirir

Not : Web element ile ilgili bu method'ların dışında size, istenen CSS özelliği, Location, Rect.geometrik özellikler, DOM değeri vb.. Bir çok method vardır ama Automation Test için kullanmiyoruz



WebElement Is Method'lari

webElement.isEnabled(); Web Element erisilebilir ise true
yoksa false doner

webElement.isDisplayed(); Web Element gorunur ise true
yoksa false doner

webElement.isSelected(); Web Element secili ise true
yoksa false doner

Not : Web element ile ilgili bu method'larin disinda size, istenen CSS ozelligi, Location, Rect.geometrik ozellikler, DOM degeri vb.. Bir cok method vardir ama Automation Test icin kullanmiyoruz



WebElement Method'lari

Tekrar Testi

1. Bir class oluşturun : AmazonSearchTest
2. Main method oluşturun ve aşağıdaki görevi tamamlayın.
 - a.web sayfasına gidin. <https://www.amazon.com/>
 - b. Search(ara) "city bike"
 - c. Amazon'da görüntülenen ilgili sonuçların sayısını yazdırın
 - d. Sonra karşınıza çıkan ilk sonucun resmine tıklayın.



HTML Parent-Child-Sibling Terimleri Nedir?

```
<div class="navFooterLine navFooterLinkLine navFooterDescLine" role="navigation" aria-label="More on Amazon.com">
  <table class="navFooterMoreOnAmazon" cellspacing="0">
    <tbody>
      <tr>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">
          <a href="https://advertising.amazon.com/?ref=footer_advtsing_amzn_com" class="nav_a">
            "Amazon Advertising"
            <br>
            <span class="navFooterDescText">...</span>
          </a>
        </td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
```

Tags: < > şeklinde görülen komutlara etiket (tag) adı verilir.

parent-child-sibling relationship(Ebeveyn-çocuk-kardeş ilişkisi) hakkında konuştuğumuzda, yalnızca tag adları önemlidir

<table>, <div> in çocuğu (child) ve <tbody>, <tr> in ebeveyndir (parent)

<td> tagları ise siblings (kardeş)'dır



LOCATORS XPath

7) By.xpath();

Bir WebElement'i locate etmek icin kullanabilecegimiz en etkin method'dur.

```
WebElement passwordTextBox =driver.findElement(By.xpath("xpath"));
```

Method'un yazimi acisindan diger 6 yontem ile ayni olmakla beraber xpath'i digerlerinden ayiran cok onemli bir fark vardir



Diger 6 method HTML kod'una baglidir. Web element'in kodunda id yoksa By.id() method'unu, web element link degilse By.linkText() method'unu kullanamazsiniz.

Xpath ise dinamiktir. Her turlu web element icin mutlaka bir xpath yazilabilir

2 cesit Xpath yazilabilir

- 1.**Absolute** xpath (mutlak)
- 2.**Relative** xpath (bagil)



LOCATORS XPath

1. Absolute Xpath()

```
<div class="navFooterLine navFooterLinkLine navFooterDescLine" role="navigation" aria-label="More on Amazon.com">
  <table class="navFooterMoreOnAmazon" cellspacing="0">
    <tbody> // div/ table/ tbody
      <tr>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">
          <a href="https://advertising.amazon.com/?ref=footer_advtsing_amzn_com" class="nav_a">
            "Amazon Advertising"
            <br>
            <span class="navFooterDescText">...</span> // tbody / tr / td[3] // span
          </a>
        </td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
        <td class="navFooterDescItem">...</td>
        <td class="navFooterDescSpacer" style="width: 4%"></td>
```

Absolute xpath yazmak için en basa // sonraki her adımda / yazarak hedef web element'e kadar tüm tag'lar yazılır.

Eğer aynı path'e sahip birden fazla element varsa index kullanılabilir. [2] gibi

Eğer bir parent'in grand child'lari içinde unique bir tag varsa parent // grand child yazılabilir



LOCATORS XPath

2. Relative Xpath()

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute" dir="auto" tabindex="0" aria-label="Search">
```

Bir web element'te temel olarak 3 bileşen vardır.

- 1) tag name input
- 2) attribute type, id , value, name, autocomplete, placeholder, class
- 3) attribute value type → 'text', id→ 'twotabsearchtextbox' , value → "

Bu 3 bileşeni birlikte kullanarak her bir webelement için unique sonuc veren bir çok xpath yazılabilir

Relative xpath yazmak için bu 3 bileşen aşağıdaki gibi bir araya getirilir, unique sonuc veren her relative xpath kullanılabilir.

//tagName[@attributelsmi='attributeValue']



LOCATORS XPath

2. Relative Xpath()

```
<input type="text" id="twotabsearchtextbox" value="" name="field-keywords" autocomplete="off" placeholder="" class="nav-input nav-progressive-attribute" dir="auto" tabindex="0" aria-label="Search">
```

- ❖ Genelde 3 bilesen de kullanılır, Ancak bazen daha az bilesen yazmak da yeterli olabilir.

Sadece tag name kullanarak xpath yazmak için

```
driver.findElement(By.xpath("//input"));
```

Tag name farketmeksiz attribute ismi ve attribute value kullanarak xpath yazmak için

```
driver.findElement(By.xpath("// * [@type='text']"));
```

Attribute name farketmeksiz tag name ve attribute value kullanarak xpath yazmak için

```
driver.findElement(By.xpath("//input[@ *= 'text']"));
```

Attribute value farketmeksiz tag name ve attribute ismi kullanarak xpath yazmak için

```
driver.findElement(By.xpath("//input[@type]"));
```



LOCATORS XPath

Class Work: Add Remove Element

- 1- https://the-internet.herokuapp.com/add_remove_elements/ adresine gidin
- 2- Add Element butonuna basin
- 3 Delete butonu'nun gorunur oldugunu test edin
- 4 Delete tusuna basin
- 5 "Add/Remove Elements" yazisinin gorunur oldugunu test edin



LOCATORS XPath

2. Relative Xpath()

- ❖ Bazen de attribute'e bagli olmadan sadece web element icinde bulunan text kullanilabilir.

Exact Text(Belirli bir text) ile element bulma:

```
driver.findElement(By.xpath("//tagname[.='text name']"));  
driver.findElement(By.xpath("//*[.='text name'] "));  
driver.findElement(By.xpath("//*[text( )='exact text with extra space and all'] "));
```

Belirli bir metni içeren bir öğeyi bulmak için şunları kullanabiliriz:

```
driver.findElement(By.xpath("//*[contains(text(),'piece of text')] "));
```

- ❖ Tek attribute ile unique bir sonuca ulasamazsak birden fazla attribute yazabiliriz

```
driver.findElement(By.xpath("//div[@id='logo' or class='flex-col logo'] "));  
driver.findElement(By.xpath("//div[@id='logo' and class='flex-col logo'] "));
```



LOCATORS XPath

Class Work: Add Remove Element

Asagidaki testi text'leri kullanarak locate edin

- 1 https://the-internet.herokuapp.com/add_remove_elements/ adresine gidin
- 2 Add Element butonuna basin
- 3 Delete butonu'nun gorunur oldugunu test edin
- 4 Delete tusuna basin
- 5 "Add/Remove Elements" yazisinin gorunur oldugunu test edin



LOCATORS CssSelector

8- By.cssSelector() Method

```
<input class="form-control" placeholder="Password" data-test="password"
type="password" name="session[password]" id="session_password">
```

Css selector xpath'e benzer. Üç ana tip kullanılır

1) css = tagName[attribute name= 'value'];

```
driver.findElement(By.cssSelector("input[name='session[password]']"));
```

2) css="tagName#idValue" veya sadece css="#idValue" =>yalnızca id value ile çalışır

```
driver.findElement(By.cssSelector("input#session_password"));
```

3) css="tagName.className" veya sadece css=".className"=>yalnızca class value ile çalışır

```
driver.findElement(By.cssSelector(".form-control"));
```



LOCATORS CssSelector

Home Work: Log in Test Using Css

- 1) Bir class oluşturun : Locators_css
- 2) Main method oluşturun ve aşağıdaki görevi tamamlayın.
 - a. Verilen web sayfasına gidin. http://a.testaddressbook.com/sign_in
 - b. Locate email textbox
 - c. Locate password textbox ve
 - d. Locate signin button
 - e. Asagidaki kullanıcı adını ve şifreyi girin ve sign in düğmesini tıklayın
 - i. Username : testtechproed@gmail.com
 - ii. Password : Test1234!

NOT: cssSelector kullanarak elementleri locate ediniz.



LOCATORS XPath

Tekrar Testi

- 1-C01_TekrarTesti isimli bir class olusturun
- 2- <https://www.amazon.com/> adresine gidin
- 3- Browseri tam sayfa yapin
- 4- Sayfayı "refresh" yapin
- 5 Sayfa basliginin "Spend less" ifadesi icerdigini test edin
- 6 Gift Cards sekmesine basin
- 7 Birthday butonuna basin
- 8 Best Seller bolumunden ilk urunu tiklayin
- 9- Gift card details'den 25 \$'i secin
- 10-Urun ucretinin 25\$ oldugunu test edin
- 10-Sayfayı kapatın



LOCATORS Relative Locators

Relative Locators nedir ?

- ❖ Selenium 4 ile gelen yeniliklerden biri de bagil locator'lardir.
- ❖ Bir web elementi direk locate edemedigimiz durumlarda gunluk hayatimizda kullandigimiz sekilde o web elementi etrafindaki web elementlerin referansi ile tarif edebiliriz.
- ❖ Ornegin yandaki resimde Berlin icin bir cok relative locator tanimlayabiliriz.
 - Boston'in saginda , Sailor'in ustunde
 - NYC'nin altinda, Bay Area'nin solunda
 - Boston yakinlarinda Bay Areanin solunda ve Toronto'nun saginda vb..



<https://www.diemol.com/selenium-4-demo/relative-locators-demo.html>



LOCATORS Relative Locators

Class Work: Relative Locators

```
driver.get("https://www.diemol.com/selenium-4-demo/relative-locators-demo.html#");

WebElement boston=driver.findElement(By.id("boston"));
WebElement sailor = driver.findElement(By.id("sailor"));

WebElement berlin = driver.findElement(with(By.tagName("li")).above(sailor).toRightOf(boston));

WebElement mountie=driver.findElement(with(By.className("ui-li-has-thumb")).below(boston));
```





LOCATORS Relative Locators

Class Work: Relative Locators

- 1) <https://www.diemol.com/selenium-4-demo/relative-locators-demo.html> adresine gidin
- 2) Berlin'i 3 farkli relative locator ile locate edin 3
-) Relative locator'larin dogru calistigini test edin





Maven™

Bir projeyi geliştirmek kadar onu anlasılır kılmak, tekrar eden kodlardan kaçınmak, update'leri tek merkezden kolayca yapabilmek ve projenin surdurulebilirliğini sağlamak (maintenance) da önemlidir.

- ❖ Apache Maven bir yazılım proje yönetimi ve anlama aracıdır.
- ❖ Maven, en iyi uygulamaların geliştirilmesi için güncel ilkeleri bir araya getirmeyi ve bir projeye bu yönde rehberlik etmeyi kolaylaştırmayı amaçlamaktadır.
- ❖ Maven geliştiricileri birçok ayrıntıdan korur.
- ❖ Proje nesne modeli (POM Project Object Model) konseptine dayalıdır.

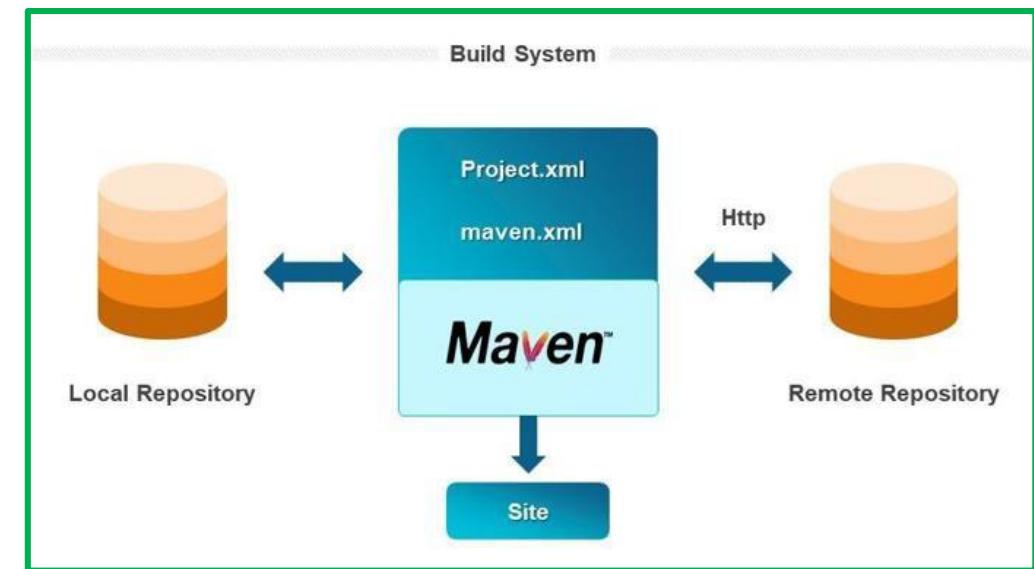
Apache
Maven™

<https://maven.apache.org/>



Maven™

- ❖ Maven bir Java derleme aracıdır (build tool). Maven proje otomasyon ve yönetim aracıdır (automation and management tool).
- ❖ Maven, konfigürasyon için pom.xml dosyasını kullanır. Bu dosya projenin insası, raporlaması ve dokümantasyonu için gerekli bütün bilgileri içerir (dependencies, plugins)
- ❖ Bir Maven projesine aşina olduğunuzda, tüm Maven projelerinin nasıl inşa edildiğini bilirsiniz. Bu, birçok projede gezinirken zaman kazandırır.





Maven™

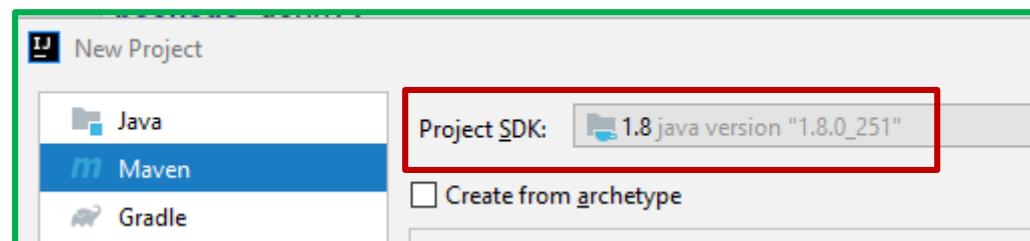
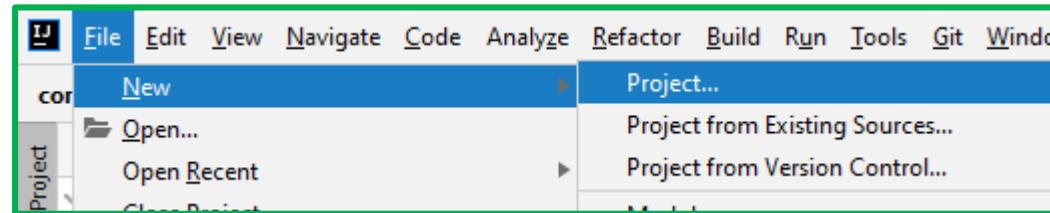
Neden Maven?

- ❖ Tekrarlanabilir derlemeler / yürütütmeleri yapmak kolaylaştırır.(Repeatable builds/executions.)
- ❖ Proje yönetimini kolaylaştırır.
- ❖ Maven, dependencylerle ilgili mevcut jar dosyalarını otomatik olarak indirir.
- ❖ Birden fazla IDE (intelliJ, eclipse, vb.) ve araçlarla(tools) çalışır.
- ❖ Open source
- ❖ Geniş kullanıcı tabanı



Maven™

Maven PROJESINI OLUSTURMA



1. Create Project: File -> New -> Project

2. Select Maven -> click next

Java versiyonunun bilgisayarinizdaki version ile
ayni oldugunu control edin

3. Name: com.Batch81Maven -> finish -> click on
(EnableAutoImport)

4. Package olusturun, name : day04 -> Right click on java create the package

5. Class olusturun, name : FirstMavenClass



POM XML

✓ Pom.xml dosyası nedir?

Maven projesindeki en onemli dosyadır. Dependencies'i POM dosyasına ekler ve POM dosyasından yönetiriz.

✓ Maven ve pom arasındaki fark nedir?

Maven bir araç (tool), pom bir xml dosyasıdır. Pom.xml maven'in bir parçasıdır ve pom dependencies'i yönetir.

✓ Projenizde pom'ı nasıl kullanırız? Neden kullanırız?

Dependencies'i yönetmek için . Tüm projeyi yönetmemize yardımcı olur. Pom ayrıca artifact id, group id, ve version gibi proje bilgilerine de sahiptir.

✓ Dependency nedir? Nasıl kullanıyorsunuz? Neden kullanıyorsunuz?

Sürücüler (drivers) kurmak, sürücüler oluşturmak için dependencies gereklidir. Gerekli araçları içe aktarır(import).

✓ Pom dosyanızı ve dependencies'i nasıl güncellersiniz ?

Mvnrepository.com adresinden gerekli dosyalari bulur ve pom dosyamiza ekleriz



Maven™

POM XML DOSYASINA DEPENDENCIES EKLEME

- 1) Mvnrepository.com adresine gidelim
- 2) WebDriverManager aratalim (En guncel ve en cok kullanılan optimum versiyonu alalım)
- 3) Asagida Maven yazan bolumdeki kodlari kopyalayalim
- 4) pom XML dosyasina gidelim
- 5) <properties> satirinin altina <dependencies> </dependencies> tag'i olusturalim
- 6) Kopyaladigimiz kodlari <dependencies> bolumune yapistiralim
- 7) Ayni islemleri Selenium Java dependency icin de yapalim
- 8) Pom XML dosyasinin sag tarafinda **Maven** yazan bolumu tiklayalim
- 9) Yenile butonuna tiklayip asagida Dependencies bolumunun olustugunu ve altında ekledigimiz kutuphanelerin gorundugunden emin olun



Maven™

POM XML DOSYASINA DEPENDENCIES EKLEME

The screenshot shows the Maven tool window on the right and the code editor on the left. The code editor displays the `pom.xml` file for the project `com.mavenSe4`. A red box highlights the `<dependencies>` section, which contains two dependency declarations for `webdrivermanager` and `selenium-java`. The Maven tool window shows the project structure with the `Dependencies` node expanded, displaying the same two dependencies.

```
<groupId>org.example</groupId>
<artifactId>com.mavenSe4</artifactId>
<version>1.0-SNAPSHOT</version>

<properties>
    <maven.compiler.source>11</maven.compiler.source>
    <maven.compiler.target>11</maven.compiler.target>
</properties>

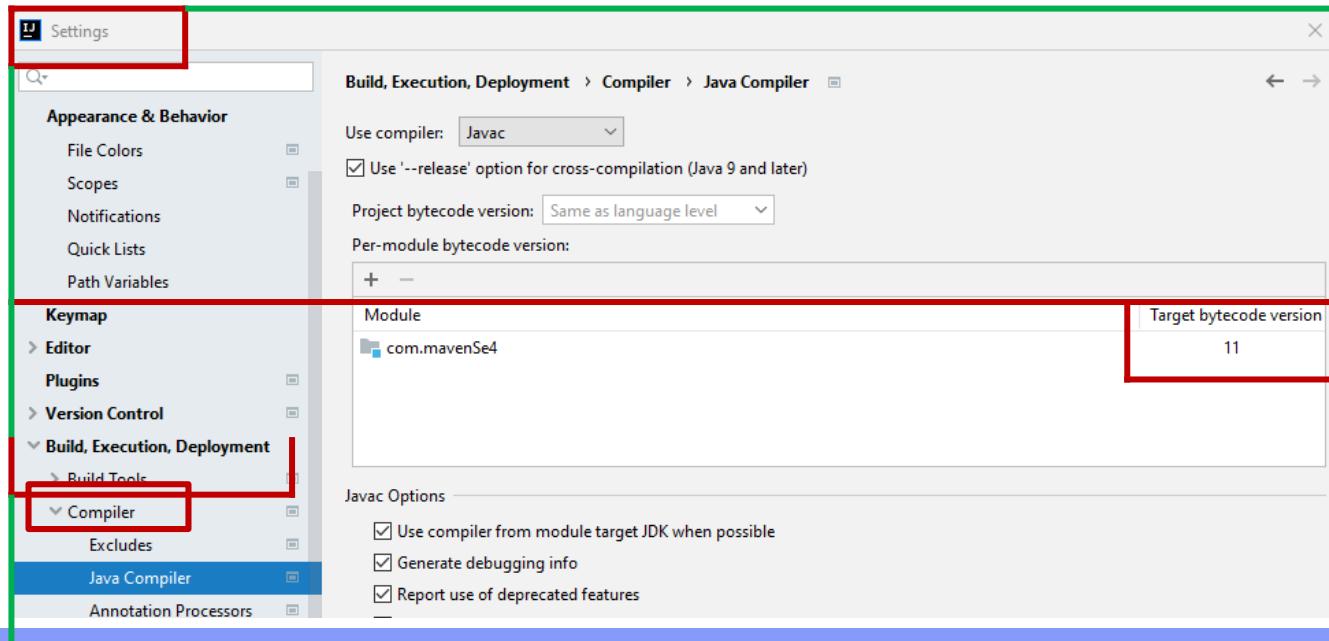
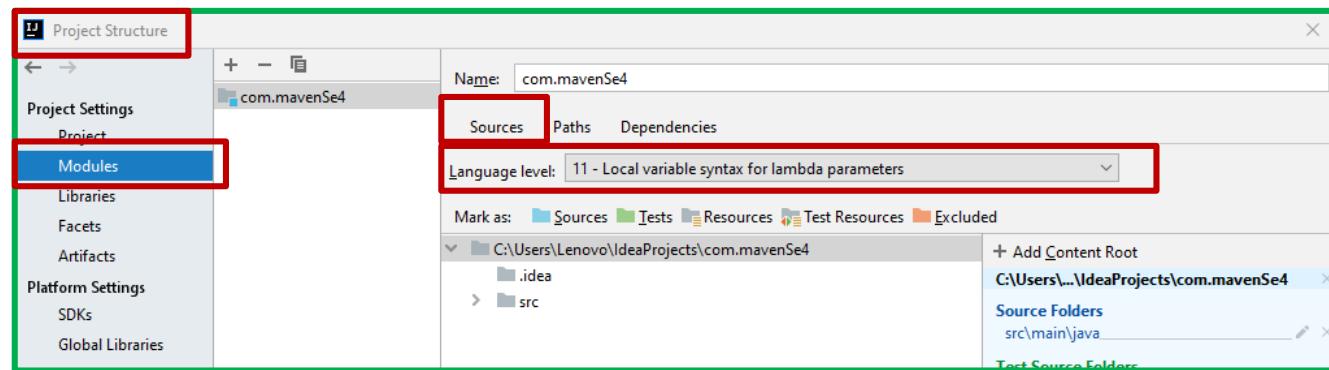
<dependencies>
    <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
    <dependency>
        <groupId>io.github.bonigarcia</groupId>
        <artifactId>webdrivermanager</artifactId>
        <version>5.0.3</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>4.1.0</version>
    </dependency>
</dependencies>
</project>
```



Maven™

CLASS AYARLARI



File
Project Structure
Modules
Sources
Language level : min **8** yapalim,
bilgisayarinizda kurulu java 8 veya üstü ise
java versiyonu aynı olmalı

File
Settings
Build,Execution,Deployment
Compiler
Java Compiler
Target bytecode version : min **8** yapalim,
bilgisayarinizda kurulu java 8 veya üstü ise
java versiyonu aynı olmalı



CLASS WebDriver AYARLARI

```
WebDriverManager.chromedriver().setup();
WebDriver driver = new ChromeDriver();
```

Class Work Amazon Search Test

- 1 <https://www.amazon.com/> sayfasina gidelim
- 2 arama kutusunu locate edelim
- 3 “Samsung headphones” ile arama yapalim
- 4 Bulunan sonuc sayisini yazdiralim
- 5 Ilk urunu tiklayalim
- 6 Sayfadaki tum basliklari yazdiralim



Soru 1

1. <http://zero.webappsecurity.com> sayfasina gidin
2. Signin buttonuna tiklayin
3. Login alanine "username" yazdirin
4. Password alanine "password" yazdirin
5. Sign in buttonuna tiklayin
6. Pay Bills sayfasina gidin
7. amount kismina yatirmak istediginiz herhangi bir miktari yazin
8. tarih kismina "2020-09-10" yazdirin
9. Pay buttonuna tiklayin
10. "The payment was successfully submitted." mesajinin ciktigini control edin



Tekrar Testi

- 1-C01_TekrarTesti isimli bir class olusturun
- 2- <https://www.google.com/> adresine gidin
- 3- cookies uyarisini kabul ederek kapatın
- 4 Sayfa basliginin “Google” ifadesi icerdigini test edin
- 5 Arama cubuguna “Nutella” yazip aratin
- 6 Bulunan sonuc sayisini yazdirin
- 7 sonuc sayisinin 10 milyon’dan fazla oldugunu test edin
- 8 Sayfayı kapatın



Soru 3

1. "https://www.saucedemo.com" Adresine gidin
2. Username kutusuna "standard_user" yazdirin
3. Password kutusuna "secret_sauce" yazdirin
4. Login tusuna basin
5. Ilk urunun ismini kaydedin ve bu urunun sayfasina gidin
6. Add to Cart butonuna basin
7. Alisveris sepetine tiklayin
8. Sectiginiz urunun basarili olarak sepete eklendigini control edin
9. Sayfayı kapatın



JUnit

- ❖ Java ile en temel framework JUnit ile oluşturulabilir.
- ❖ Developerlar da unit testleri calistirmak icin kullanirlar.
- ❖ Biz testlerimizi yapmak icin JUnit'in ileri sürümü olduğundan TestNG framework oluşturup kullanacagiz.
- ❖ Junit maven projesi üzerinde calistigindan <https://mvnrepository.com/> sitesinden dependency'leri projemize ekleriz.
- ❖ Test'lerimizi yaparken main method, if-else gibi java kod bloklarini kullanmak yerine Junit annotationlarını ve method'larine kullanabiliriz.
- ❖ Son framework olarak kuracagımız Cucumber framework'da da Junit kullanacagiz





Annotations

- ❖ Selenium'da kodlarımızı yazarken "@" işaretini ile başlayan notasyonlar kullanırız.
- ❖ Java Annotationlar ile derleyiciye (Compiler) talimatlar verebiliriz.
- ❖ Annotation, bir veri hakkında bilgi barındıran veriyi sağlayan basit bir yapıdır. Bu sağladığı bilgiye de "metadata" denir.
- ❖ Notasyonlar(Annotation) genellikle Java'da konfigürasyon amacıyla kullanılır. Kullanıldığı bileşene ek özellikle katar. Bu bileşenler sınıf, metod, değişkenler, paket ya da parametreler olabilir. Bunların hepsinde notasyonları kullanabiliriz.

En çok kullanılan Junit annotation'ları

@Test

@BeforeClass @AfterClass

@Before , @After

@Ignore



@Test ve @Ignore

- ❖ Junit ile Main Method kullanma donemini bitiriyoruz.
- ❖ Junit Framework kullandigimizda yazdigimiz test metodunun calismasi icin basina @Test notasyonu eklememiz yeterlidir.
- ❖ @Test notasyonu eklemedigimiz metot test sirasinda calistirilmaz. Ancak cagrilrsa calisir.
- ❖ Yazdigimiz bazi test metotları henüz tamamlanmamış veya değişiklikleri ugrayabileceinden dolayı test sirasinda calistirilmasini istemiyorsak @Ignore notasyonu eklememiz yeterlidir.
- ❖ @Ignore notasyonun tanimli oldugu metotlar test sirasinda calistirilmayacaktir. Ayrca istenilirse @Ignore("aciklama") seklinde yazilarak metodun neden test edilmesini istemedigimizide yazabiliriz.



Annotations

@Before ve @After

- ❖ Before notasyonu, her test method'undan önce çalışır. Örneğin bir sayfa ile test yapıyorsak ve her testten önce o sayfaya gitmemiz gerekiyorsa @before kullanabiliriz.
@before notasyanunun kullanıldığı method'a genelde setup() ismi verilir.

- ❖ After notasyonu, her test method'undan sonra çalışmaktadır. Örneğin test sırasında kullandığımız sayfanın kapatılması gibi.
@after notasyanunun kullanıldığı metoda genelde teardown() ismi verilir

Not : Test method'u ve Test farklı yapılardır.

Test dedigimizde tek bir method veya içinde bircok class ve package barindiran bir yapı olabilir. Regression test, smoke test vb..

Test method'u ise @Test notasyonu kullanılarak olusturulan ve bagimsiz olarak calistirabilecegimiz en kucuk test yapısıdır



Annotations

@BeforeClass ve @AfterClass

- ❖ BeforeClass notasyonu, bir class'daki tüm testlerden önce yapılması gereken bir işlem işlem varsa kullanılır (precondition).
Örneğin test metodlarımız çalışmadan driver olusturup tüm methodlarda kullanabilirim.

- ❖ AfterClass notasyonu da, bir class'daki tüm testler tamamlandıktan sonra yapılması gereken işlemlerde kullanılır.
Örneğin actigimiz sayfayı kapatmak veya elde ettigimiz test sonuclarını raporlamak gibi.

- ❖ @BeforeClass ve @AfterClass notasyonları test sürecinde bir kere çalışırken, @Before ve @After notasyonları her test method'unun başında ve sonunda çalışmaktadır.



JUnit

Annotations

@BeforeClass - @AfterClass ve **@Before - @After**
notasyonlarının karşılaştırılması

1) Tüm Class çalıştırılırsa

```
Before class method'u calisti → @BeforeClass
Before method'u calisti
Test 1 calisti } @Before - @After
After method'u calisti
Before method'u calisti } @Before - @After
Test 2 calisti } @Before - @After
After method'u calisti
Before method'u calisti } @Before - @After
Test 3 calisti } @Before - @After
After method'u calisti
After class method'u calisti → @AfterClass
```

2) Tek bir test method'u çalıştırılırsa

```
Before class method'u calisti → @BeforeClass
Before method'u calisti
Test 3 calisti } @Before - @After
After method'u calisti
After class method'u calisti → @AfterClass
```

```
public class C03 {
    @BeforeClass
    public static void beforeClassM(){
        System.out.println("Before class method'u calisti");
    }
    @Before
    public void beforeM() {
        System.out.println("Before method'u calisti");
    }
    @Test
    public void test1(){
        System.out.println("Test 1 calisti");
    }
    @Test
    public void test2(){
        System.out.println("Test 2 calisti");
    }
    @Test
    public void test3(){
        System.out.println("Test 3 calisti");
    }
    @AfterClass
    public static void afterClassM(){
        System.out.println("After class method'u calisti");
    }
    @After
    public void afterM() {
        System.out.println("After method'u calisti");
    }
}
```



CheckBox

1. Bir class oluşturun : CheckBoxTest
2. Gerekli yapıyi olusturun ve aşağıdaki görevi tamamlayın.
 - a. Verilen web sayfasına gidin.
<https://the-internet.herokuapp.com/checkboxes>
 - b. Checkbox1 ve checkbox2 elementlerini locate edin.
 - c. Checkbox1 seçili değilse onay kutusunu tıklayın
 - d. Checkbox2 seçili değilse onay kutusunu tıklayın



RadioButton

1. Bir class oluşturun : RadioButtonTest

2. Gerekli yapıyı olusturun ve aşağıdaki görevi tamamlayın.

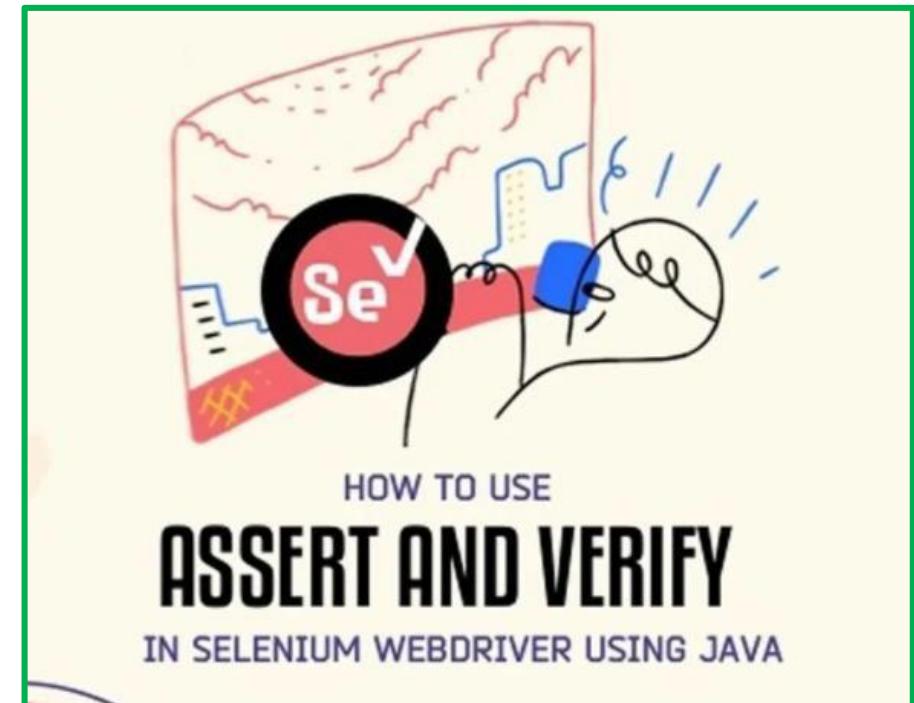
- <https://www.facebook.com> adresine gidin
- Cookies'i kabul edin
- "Create an Account" button'una basin
- "radio buttons" elementlerini locate edin
- Secili degilse cinsiyet butonundan size uygun olani secin



JUnit

Assertions: (Verification)

- ❖ Test için uygun bir şekilde hazırlanmış bir framework'da expected değerlerin actual değerlere eşit olup olmadığını if-else statement ile bakılmaz.
- ❖ JUnit framework'ünde test senaryomuzu çalıştırıldıktan sonra beklediğimiz sonuçların olup olmadığını tespit etmek için assert class'ından method'lar kullanırız.
- ❖ Junit'de, belirli koşulları test etmek için isimleri assert ile başlayan statik method'lari kullanırız.
- ❖ Bu method'lar ile excepted ve actual değerleri karşılaştırarak testimizi sonuçlandırırız.





Assertions

- ❖ Assert method'unu secmeden once kiyaslamada bekledigimiz sonucun true veya false olmasina karar vermeliyiz.

```
String exceptedName= "Ali Can";
String actualName= "ALI CAN";

exceptedName.equals(actualName); → false

exceptedName.equalsIgnoreCase(actualName); → true
```

```
// Emeklilik yasi 65

int yas1=63;

int yas2=68;

emekliOlabilirMi(yas1); → false

emekliOlabilirMi(yas2); → true
```

- ❖ **ONEMLI OLAN** kiyaslama sonucunun true veya false olması değil, bekledigimiz sonucun olup olmamasıdır. Assertion başarısız olursa AssertionException ile hata mesajı verilir.

Assert.assertEquals(actualName,expectedName) → FAILED

Assert.assertTrue(yas2>65) → PASSED

Assert.assertFalse(yas1>65) → PASSED

Assert.assertTrue(yas2<65) → FAILED

Assert.assertFalse(yas1<65) → FAILED



Assertions

A=20 B=30 C=40 D=20

Assert.assertEquals(A,B) → FAILED

false

Assert.assertEquals(A,D) → PASSED

true

Assert.assertTrue(C>D) → PASSED

true

Assert.assertTrue(C>B) → PASSED

true

Assert.assertFalse(B>D) → FAILED

true

Assert.assertFalse(B>A) → FAILED

true



Assertions

- 1) Bir class oluşturun: **YoutubeAssertions**
- 2) <https://www.youtube.com> adresine gidin
- 3) Aşağıdaki adları kullanarak 3 test metodu oluşturun ve gerekli testleri yapın
 - **titleTest** => Sayfa başlığının “YouTube” olduğunu test edin
 - **imageTest** => YouTube resminin görüntüülendiğini (`isDisplayed()`) test edin
 - **Search Box** 'in erişilebilir olduğunu test edin (`isEnabled()`)
 - **wrongTitleTest** => Sayfa basliginin “youtube” olmadigini doğrulayin



Assertions

1. Bir Class olusturalim YanlisEmailTesti
2. <http://automationpractice.com/index.php> sayfasina gidelim
3. Sign in butonuna basalim
4. Email kutusuna @isareti olmayan bir mail yazip enter'a bastigimizda "Invalid email address" uyarisi ciktigini test edelim



ASSERT OZET

Assert, bir test senaryosunun PASS veya FAILED durumunu belirlemeye kullanılan yararlı bir yöntemdir. Seçilen metod ve yazılan **boolean koşul'a** göre test sonucu belirlenir.

En çok kullandığımız 3 Assert metodu;

1) Assert.assertTrue(**koşul**)

Yazılan koşul'un sonucu True ise test PASS, yoksa test FAILED olur

Assert.assertTrue(**20 > 15**) → Test PASSED

True

Assert.assertTrue(**10 > 30**) → Test FAILED

False



ASSERT ÖZET

2) Assert.assertFalse(**koşul**)

Yazılan koşul'un sonucu False ise test PASS, yoksa test FAILED olur

Assert.assertFalse(40 > 50) → Test PASSED
 False

Assert. assertFalse(30 > 20) → Test FAILED
 True

3) Assert.assertEquals(**expected, actual**)

Yazılan expected ile actual eşit ise test PASS, yoksa test FAILED olur

Assert.assertEquals("Ali" , "Ali") → Test PASSED
 True

Assert. assertEquals(30 , 20) → Test FAILED
 False



JUnit

Handle Dropdown

➤ **Adım1:** Dropdown menüyü herhangi bir locator ile locate edin.

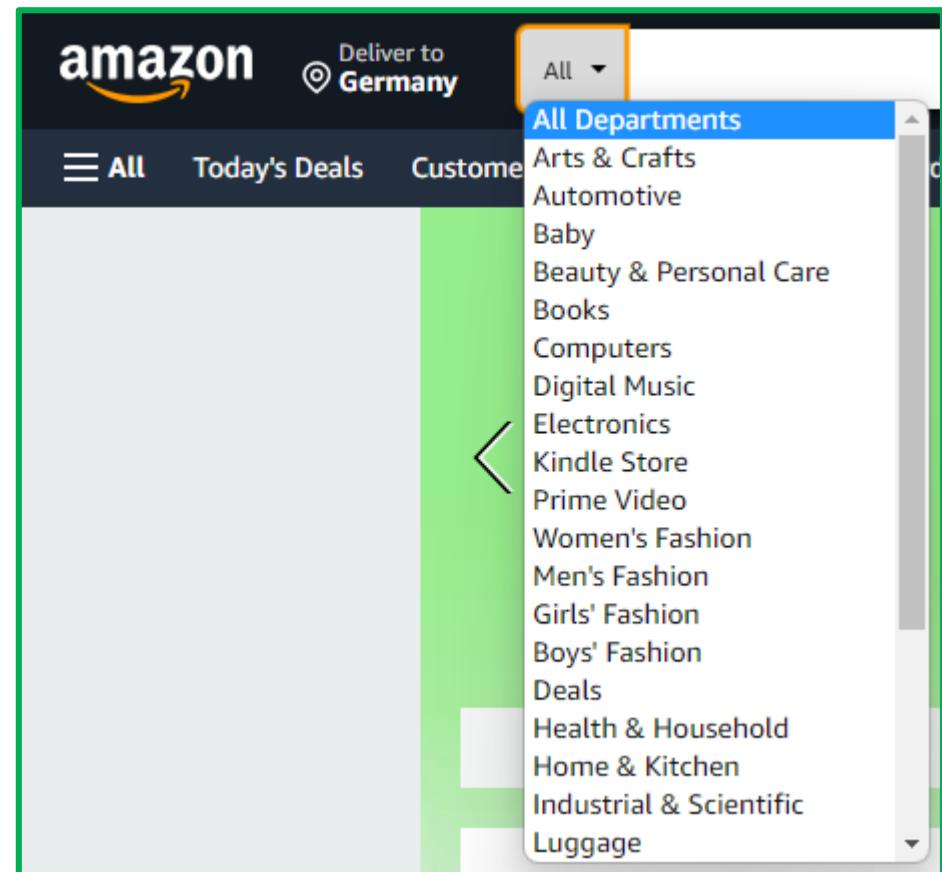
```
WebElement ddm=driver.findElement(By.id("value of id"));
```

➤ **Adım 2:** Yeni bir **Select** objesi oluşturun ve daha önce locate ettigimiz WebElement'i **parametre** olarak yeni objeye ekleyin

```
Select options=new Select(ddm);
```

➤ **Adım 3:** Select class'ından kullanabileceginiz 3 yontemden biriyle dropdown menusundeki elemanlardan istediginizi secin

```
options.selectByIndex(1);
```



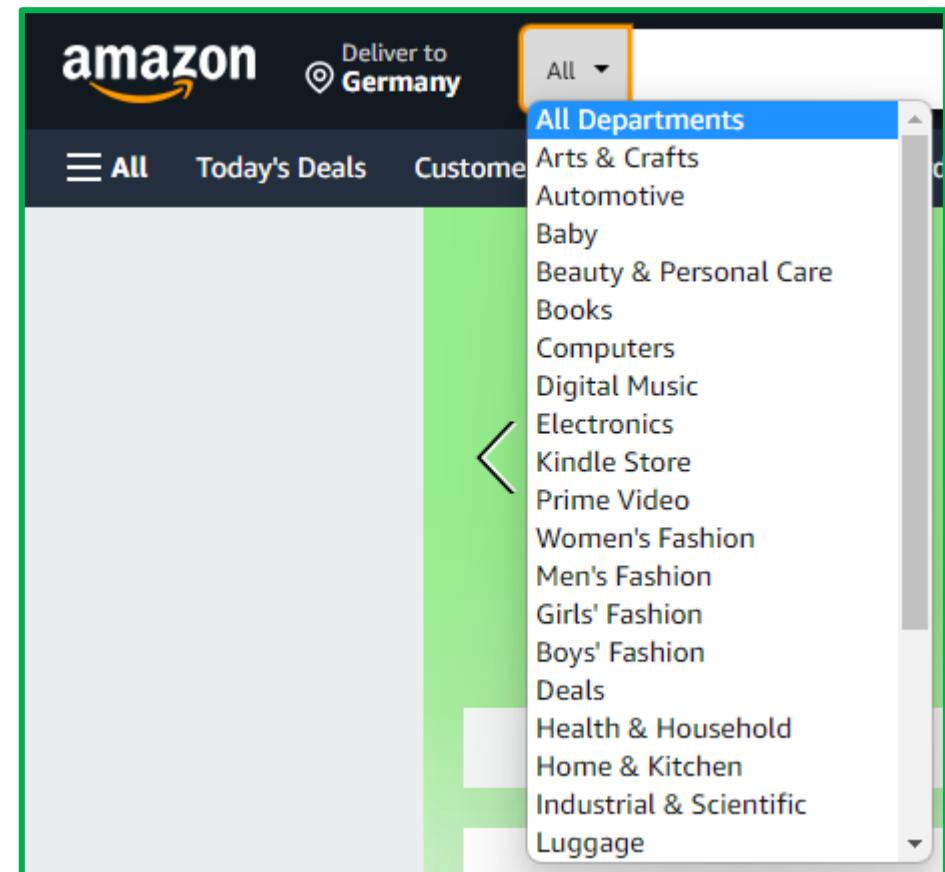


Handle Dropdown

Dropdown menusundeki elementleri Select Class'indan kullanacagimiz yontemlerle 3 sekilde secebiliriz

1. Index kullanarak **selectByIndex()**;
2. Deger kullanarak **selectByValue()**;
3. Gorunen degerini kullanarak **selectByVisibleText()**;

Istenirse **getOptions()**; methodu kullanilarak DropDown'daki tum elementler webelementlerden olusan bir listeye konabilir. **List<WebElement>**





Handle Dropdown

- Bir class oluşturun: DropDown
- <https://the-internet.herokuapp.com/dropdown> adresine gidin.
 - 1.Index kullanarak Seçenek 1'i (Option 1) seçin ve yazdırın
 - 2.Value kullanarak Seçenek 2'yi (Option 2) seçin ve yazdırın
 - 3.Visible Text(Görünen metin) kullanarak Seçenek 1'i (Option 1) seçin ve yazdırın
 - 4.Tüm dropdown değerleri(value) yazdırın
 5. Dropdown'un boyutunu bulun, Dropdown'da 4 öğe varsa konsolda True , degilse False yazdırın.



JUnit

Handle Dropdown

- Bir class oluşturun: C3_DropDownAmazon
- <https://www.amazon.com/> adresine gidin.
 - Test 1

Arama kutusunun yanindaki kategori menusundeki kategori sayisinin 45 oldugunu test edin

-Test 2

1. Kategori menusunden Books secenegini secin
2. Arama kutusuna Java yazin ve aratin
3. Bulunan sonuc sayisini yazdirin
4. Sonucun Java kelimesini icerdigini test edin



Handle Dropdown

1. <http://zero.webappsecurity.com/> Adresine gidin
2. Sign in butonuna basin
3. Login kutusuna “username” yazın
4. Password kutusuna “password.” yazın
5. Sign in tusuna basin
6. Pay Bills sayfasına gidin
7. “Purchase Foreign Currency” tusuna basin
8. “Currency” drop down menusünden Eurozone’u secin
9. “amount” kutusuna bir sayı girin
10. “US Dollars” in secilmədigini test edin
11. “Selected currency” butonunu secin
12. “Calculate Costs” butonuna basin sonra “purchase” butonuna basin
13. “Foreign currency cash was successfully purchased.” yazısının çıktığını kontrol edin.



Handle Alerts

Alert Nedir?

Alert kullanıcıya bir tür bilgi vermek veya belirli bir işlemi gerçekleştirmek izni istemek için ekran bildirimini görüntüleyen küçük bir mesaj kutusudur. Uyarı amacıyla da kullanılabilir.

HTML Alerts

Bir alert çıktığında sağ click ile inspect yapabiliyorsak html alert'dir ve extra bir işleme gerek yoktur.

Js Alerts

Js alerts inspect yapılamaz, ekstra işleme ihtiyaç vardır.

Click for JS Alert

1.Simple Alert : Bu basit alert ekranda bazı bilgiler veya uyarılar görüntüler. Ok denilerek kapatılır

Click for JS Confirm

2.Confirmation Alert : Bu onay uyarısı bir tür işlem yapma izni ister. Alert onaylanıyorsa OK, onaylanmıyorsa Cancel butonuna basılır.

Click for JS Prompt

3.Prompt Alert : Bu Prompt Uyarısı kullanıcıdan bazı girdilerin girilmesini ister ve selenium webdriver metni sendkeys ("input....") kullanarak girebilir.



Handle Alert Methods

- `accept()` => Bir uyarıda(alert) OK'ı tıklamakla aynı.

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

- `dismiss()` => Bir uyarıda(alert) Cancel'ı tıklamakla aynı.

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

- `getText()` => Uyarıdaki(alert) mesajı almak için.

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

- `sendKeys("Text")` => Uyarı(alert) text kutusuna text göndermek için

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



Handle Alert Class Work

- Bir class olusturun: **Alerts**
- https://the-internet.herokuapp.com/javascript_alerts adresine gidin.
- Bir metod olusturun: **acceptAlert**
 - 1. butona tıklayın, uyarıdaki OK butonuna tıklayın ve result mesajının "You successfully clicked an alert" olduğunu test edin.
- Bir metod olusturun: **dismissAlert**
 - 2. butona tıklayın, uyarıdaki Cancel butonuna tıklayın ve result mesajının "successfully" icermedigini test edin.
- Bir metod olusturun: **sendKeysAlert**
 - 3. butona tıklayın, uyarıdaki metin kutusuna isminizi yazın, OK butonuna tıklayın ve result mesajında isminizin görüntülediğini doğrulayın.



Basic Authentication

the-internet.herokuapp.com/basic_auth

Authentication Nedir?

Kısaca, herhangi bir internet kullanıcısının, uygulamanın ya da programın, söz konusu sisteme dahil olup olamayacağını belirleyen formu ifade eder.

Uygulama ana sayfalarındaki kullanıcı adı ve password istemek de bir authentication'dır.

End user'lar için tasarılanmayan uygulamalarda(Ornegin API sorularında) bu authentication HTML komutları ile de yapılabilir.

Bu authentication'i yapabilmek için uygulamanın kullanıcılarına authentication'i nasıl yapacağına dair bilgilendirme yapmış olması gereklidir.

Oturum açın
<https://the-internet.herokuapp.com>

Kullanıcı adı:

Şifre:

Oturum açın **İptal**

Ornegin yandaki uygulama için authentication aşağıdaki gibi yapılabilir.

`https://username:password@URL`



JUnit

Basic Authentication Class Work

- 1 Bir class olusturun : BasicAuthentication
- 2 https://the-internet.herokuapp.com/basic_auth sayfasina gidin
- 3 asagidaki yontem ve test datalarini kullanarak authentication'i yapin

Html komutu : https://username:password@URL

Username : admin

password : admin

- 4 Basarili sekilde sayfaya girildigini dogrulayin

```
driver.get("https://admin:admin@the-internet.herokuapp.com/basic_auth");
```

the-internet.herokuapp.com/basic_auth

Oturum açın

<https://the-internet.herokuapp.com>

Kullanıcı adı

Şifre

Oturum açın İptal

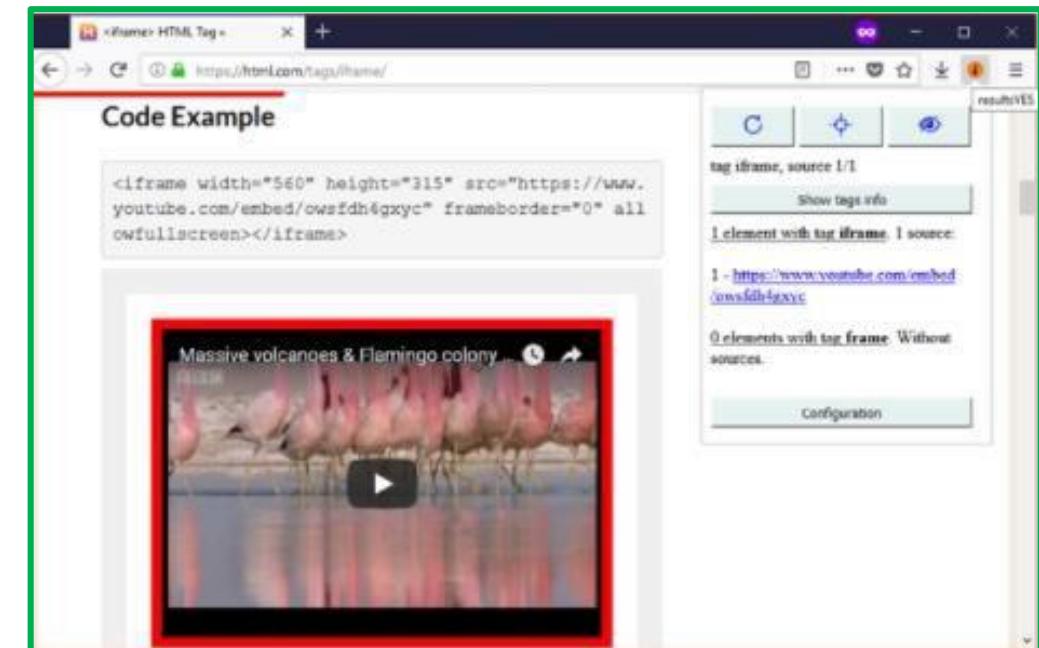


Handle Iframe

Iframe nedir?

- IFrame, bir web sayfasına içine yerleştirilmiş başka bir web sayfasıdır veya bir HTML dokumanının içine yerleştirilmiş başka bir HTML dokumanıdır.
- IFrame genellikle bir Web sayfasına dokuman, video veya interaktif media gibi başka bir kaynaktan içerik eklemek için kullanılır. <iframe> tag'ı bir inline frame belirtir.

<https://html.com/tags/iframe/>





- Bir sayfada iframe varsa, Selenium bir iframe içindeki elementleri doğrudan göremez
- switchTo() metod'u ile iframe'e geçmenin 3 yolu vardır;

1) index ile :

```
driver.switchTo().frame(index of the iframe); //index 0'dan baslar
```

2) id veya name value ile

```
driver.switchTo().frame("id of the iframe");
```

3) WebElement ile

```
driver.switchTo().frame(WebElement of the iframe);
```



Iframe'den cikmak icin 2 komut vardir

`driver.switchTo().parentFrame();` 1 ust seviyedeki frame'e cikartir

`driver.switchTo().defaultContent();` En ustteki frame'e cikmak icin kullanilir

Birden fazla iframe varsa gecislerde dikkatli olmak lazim.

Gecisler her zaman basit olamayabilir

<https://html.com/tags/iframe/>



Handle Iframe Class Work

- Bir class olusturun: IframeTest
- <https://the-internet.herokuapp.com/iframe> adresine gidin.
- Bir metod olusturun: iframeTest
 - “An IFrame containing....” textinin erisilebilir oldugunu test edin ve konsolda yazdirin.
 - Text Box'a “Merhaba Dunya!” yazin.
 - TextBox'in altinda bulunan “Elemental Selenium” linkini textinin gorunur oldugunu dogrulayin ve konsolda yazdirin.



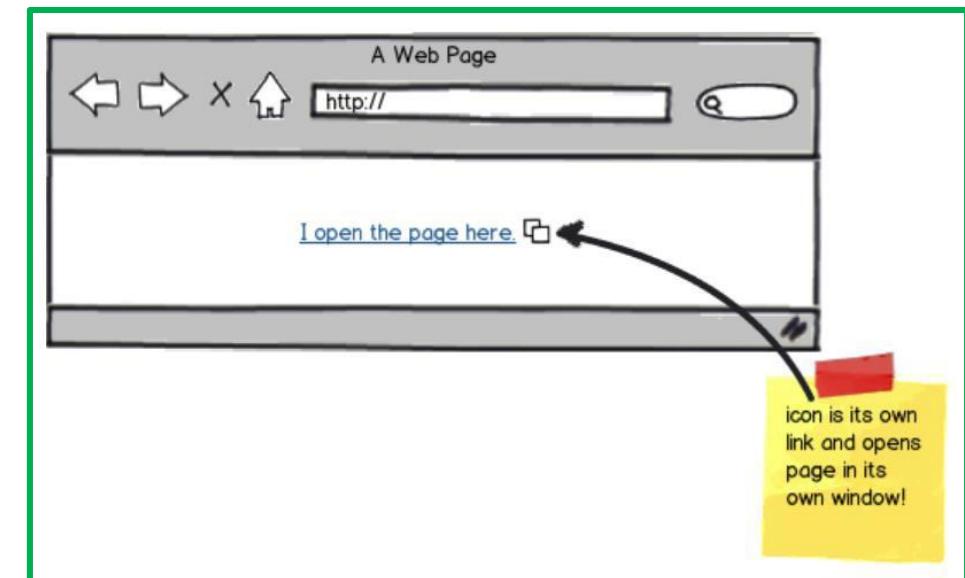
Handle Iframe Class Work

- Bir class olusturun: IframeTest02
- 1) <http://demo.guru99.com/test/guru99home/> sitesine gidiniz
 - 2) sayfadaki iframe sayısını bulunuz.
 - 3) ilk iframe'deki (Youtube) play butonuna tıklayınız.
 - 4) ilk iframe'den çıkışp ana sayfaya dönünüz
 - 5) ikinci iframe'deki (Jmeter Made Easy) linke (<https://www.guru99.com/live-selenium-project.html>) tıklayınız



Handle Windows

- Bazen bir butona tıkladığımızda, başka bir sekmede(tab) yeni bir pencere açılır.
- Birden fazla pencereyle çalışırken driver'a pencereler arasında geçis yaptırmamız gereklidir.
- Pencereler arasında geçis yapmak için **window handle** değerini kullanırız.
- **window handle** : Selenium WebDriver'in, WebDriver objesi başlatıldığından her pencereye verdiği unique alfanumerik kimlik değeridir.





JUnit

Handle Windows Method'lari

1) Icinde oldugumuz sayfanin window handle degerini alma

```
driver.getWindowHandle();
```

2) Pencereler arasında geçiş yapma(switch)

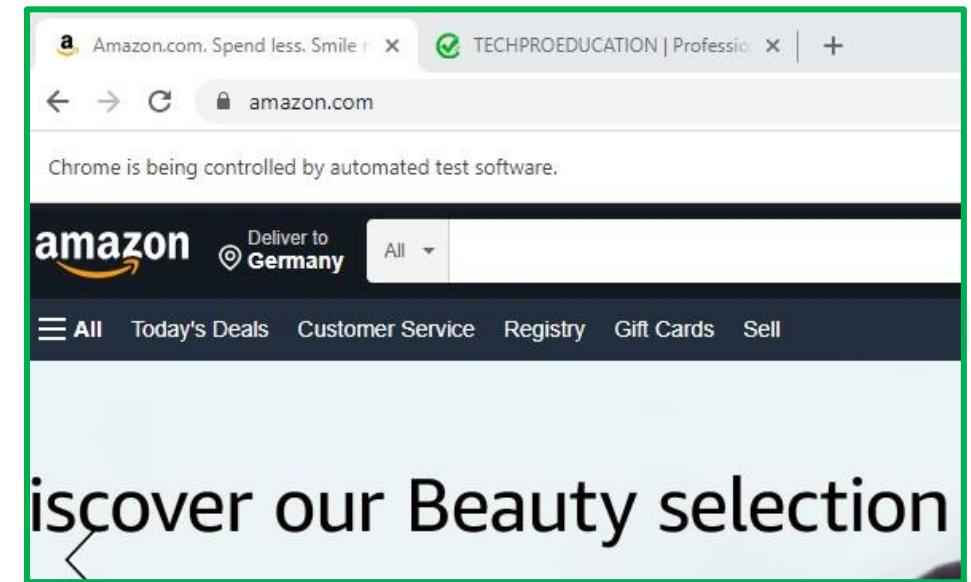
```
driver.switchTo().window(sayfa1HandleDegeri);
```

3) Yeni tab olusturup geçiş yapma(switch)

```
driver.switchTo().newWindow(WindowType.TAB);
```

4) Yeni window olusturup geçiş yapma(switch)

```
driver.switchTo().newWindow(WindowType.WINDOW);
```





Handle Windows ClassWork

- Yeni bir class olusturun: WindowHandle
- Amazon anasayfa adresine gidin.
- Sayfa'nin window handle degerini String bir degiskene atayin
- Sayfa title'nin “Amazon” icerdigini test edin
- Yeni bir tab olusturup, acilan tab'da techproeducation.com adresine gidin
- Sayfa title'nin “TECHPROEDUCATION” icerdigini test edin
- Yeni bir window olusturup, acilan sayfada walmart.com adresine gidin
- Sayfa title'nin “Walmart” icerdigini test edin
- Ilk acilan sayfaya donun ve amazon sayfasina dondugunuzu test edin



Handle Windows Method'lari

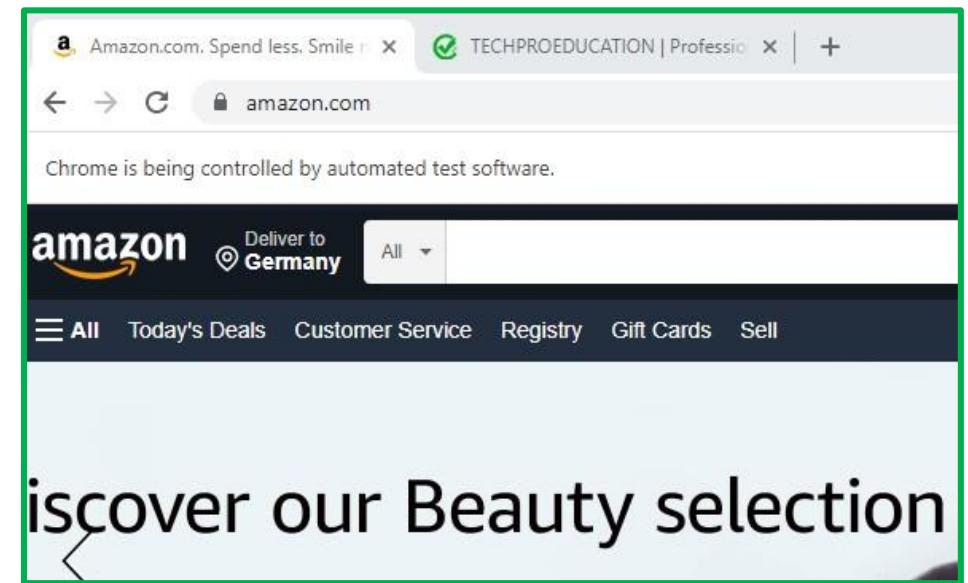
5) Acik olan pencerelerin window handle degerlerini alma

```
driver.getWindowHandles();
```

Method'un sonundaki "s" den anlasilacagi uzere birden fazla window handle return eder

Birden fazla window handle degeri oldugu icin collection return etmelidir, method Set return edecek sekilde yazilmistir.

Set'de tum window handle degerleri olacaktir. Biz icinde oldugumuz ilk sayfanin handle degerini kaydedersek, set'te olan 2 window handle degerinden, ilk sayfa handle degerine esit olmayan ikinci handle degeridir.





Handle Windows Class Work

- Tests package'ında yeni bir class olusturun: WindowHandle2
- <https://the-internet.herokuapp.com/windows> adresine gidin.
- Sayfadaki textin “Opening a new window” olduğunu doğrulayın.
- Sayfa başlığının(title) “The Internet” olduğunu doğrulayın.
- Click Here butonuna basın.
- Açılan yeni pencerenin sayfa başlığının (title) “New Window” olduğunu doğrulayın.
- Sayfadaki textin “New Window” olduğunu doğrulayın.
- Bir önceki pencereye geri döndükten sonra sayfa başlığının “The Internet” olduğunu doğrulayın.



JUnit

TestBase Class

- TestBase, testlerden önce ve sonra tekrar tekrar kullandığımız kodları içermektedir.
- İçerisindeki metodları kullanabilmemiz için extends yapıyoruz. Bu sayede test class'ımızda sadece test case'ler bulunmaktadır.
- Utilities package'da TestBase'i oluşturuyoruz.
 - setUp method
 - reports (Raporlar)
 - tearDown method
- TestBase class'i abstract yapabiliriz (Olmasada olabilir), extends yaparak kullanabiliriz. Ve bu class'da object create edemeyiz.
- olusturdugumuz driver'in farklı package'lardan kullanılmak için PROTECTED yaparız

```
import java.util.concurrent.TimeUnit;

public abstract class TestBase {
    protected WebDriver driver;
    @BeforeMethod
    public void setUp() {
        WebDriverManager.chromedriver().setup();
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().pageLoadTimeout(10, TimeUnit.SECONDS);
        driver.manage().implicitlyWait(10, TimeUnit.SECONDS);
    }

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



Actions Class

- TestNg'de Actions class'ini kullanarak mouse ve klavye ile yapabilecegimiz tum islevleri gerceklestirebiliriz.
- Actions Class birçok kullanışlı mouse ve klavye işlevine sahiptir.
- Çift tıklama (double click), sürükleme ve bırakma(drag and drop), mouse'u hareket ettirme (mouse actions) gibi karmaşık mouse eylemleri icin veya Keyboard ile yapabilecegimiz pageUp, pageDown, shift, arrowDown gibi islemleri Actions classindan object ureterek driver ile yapabiliriz.





JUnit

Actions Class

➤ 1.Adım: Actions class'ta bir object oluşturulur.

Actions actions= new Actions(driver);

➤ 2. Adım: Üzerinde çalışmak istediğiniz WebElement öğesini bulunur.

WebElement element = driver.findElement(By.id("ID"));

➤ 3.Adım : Ardından bu webelement üzerinde action gerçekleştirilir. Örneğin tıklamak için.

actions.click(element).perform();

```
driver.get("https://www.amazon.com");
Actions actions = new Actions(driver);

WebElement gununFirsatiElementi=driver.findElement(By.linkText("Today's Deals"));
actions.
    click(gununFirsatiElementi).
    perform();

Thread.sleep( millis: 3000);
WebElement amazonDevicesElement= driver.findElement(By.xpath("//span[text()='Amazon Devices'][2]"));

actions.
    sendKeys(Keys.PAGE_DOWN).
    perform();

Thread.sleep( millis: 3000);
actions.
    click(amazonDevicesElement).
    perform();
```

NOT : Action Class'ını her kullanmak istedigimizde yeniden obje olusturmamız gerekmekz.

action objesi'ni bir kere olusturunca, istediginiz kadar action. ile baslayan komut yazar ve calismasi icin sonuna **perform()** yazariz.

action objesi kullanilarak baslayan her komut, calismak icin **perform()** bekler.



Mouse Base Actions Mouse Aksiyonları

- **doubleClick ()**: WebElement'e çift tıklama yapar
- **clickAndHold ()**: WebElement üzerinde click yapılmış olarak bizden komut bekler.
- **dragAndDrop ()**: WebElement'i bir noktadan diğerine sürüklər ve bırakır
- **moveToElement ()**: Mouse'u istedigimiz WebElement'in üzerinde tutar
- **contextClick ()**: Mouse ile istedigimiz WebElement'e sağ tıklama yapar.





JUnit

Mouse Base Actions Class Work

1 Yeni bir class olusturalim: MouseActions1

2 https://the-internet.herokuapp.com/context_menu sitesine gidelim

3- Cizili alan uzerinde sag click yapalim

4 Alert'te cikan yazinin "You selected a context menu" oldugunu
test edelim.

5 Tamam diyerek alert'i kapatalim

6 Elemental Selenium linkine tiklayalim

7 Acilan sayfada h1 taginda "Elemental Selenium" yazdigini test edelim



Mouse Base Actions Class Work

Yeni bir class olusturalim: MouseActions2

- 1 <https://demoqa.com/droppable> adresine gidelim
- 2 “Drag me” butonunu tutup “Drop here” kutusunun ustune birakalim
- 3 “Drop here” yazisi yerine “Dropped!” oldugunu test edin



Mouse Base Actions Class Work

Yeni bir class olusturalim: MouseActions3

- 1 <https://www.amazon.com/> adresine gidelim
- 2 Sag ust bolumde bulunan “Account & Lists” menusunun acilmasi icin mouse'u bu menunun ustune getirelim
- 3 “Create a list” butonuna basalim
- 4 Acilan sayfada “Your Lists” yazisi oldugunu test edelim



Keyboard Base Actions Klavye Aksiyonları

- Action Class'ından kullanacağımız method'lar ile klavyedeki tusları kontrol edebiliriz.
- Bunun için 3 method kullanırız.
 - **sendKeys ()**: Öğeye bir dizi anahtar gönderir
 - **keyDown ()**: Klavyede tuşa basma işlemi gerçekleştirir
 - **keyUp ()**: Klavyede tuşu serbest bırakma işlemi gerçekleştirir





Keyboard Base Actions Class Work

- 1 Bir Class olusturalim KeyboardActions1
- 2 <https://www.amazon.com> sayfasina gidelim
- 3 Arama kutusuna actions method'larine
kullanarak samsung A71 yazdirin ve Enter'a
basarak arama yaptirin
- 4 aramanin gerceklestigini test edin





Mouse Base Actions Class Work

Yeni bir class olusturalim: D15_MouseActions4

- 1- <https://www.facebook.com> adresine gidelim
- 2- Yeni hesap olustur butonuna basalim
- 3- Ad, soyad, mail ve sifre kutularina deger yazalim ve kaydol tusuna basalim
- 4- Kaydol tusuna basalim



Faker Kutuphanesi

Faker class'i testlerimizi yaparken ihtiyac duyduğumuz isim, soyisim, adres vb bilgiler için fake değerler üretmemize imkan tanır.

Faker değerler üretmek için Faker class'ından bir obje üretir ve var olan method'lari kullanırız.

1. "https://facebook.com" Adresine gidin
2. "create new account" butonuna basin
3. "firstName" giriş kutusuna bir isim yazın
4. "surname" giriş kutusuna bir soyisim yazın
5. "email" giriş kutusuna bir email yazın
6. "email" onay kutusuna emaili tekrar yazın
7. Bir şifre girin
8. Tarih için gün seçin
9. Tarih için ay seçin
10. Tarih için yıl seçin
11. Cinsiyeti seçin
12. İsaretlediğiniz cinsiyetin seçili, diğer cinsiyet kutusunun seçili olmadığını test edin.
13. Sayfayı kapatın



Keyboard Base Actions Homework

Yeni Class olusturun ActionsClassHomeWork

- 1- "<http://webdriveruniversity.com/Actions>" sayfasina gidin
- 2- Hover over Me First" kutusunun ustune gelin
- 3 Link 1" e tiklayin
- 4 Popup mesajini yazdirin
- 5 Popup'i tamam diyerek kapatin
- 6 "Click and hold" kutusuna basili tutun
- 7-"Click and hold" kutusunda cikan yaziyi yazdirin
- 8- "Double click me" butonunu cift tiklayin



Keyboard Base Actions Class Work

1 Bir Class olusturalim KeyboardActions2

2 <https://html.com/tags/iframe/> sayfasina gidelim

3- video'yu gorecek kadar asagi inin

4 videoyu izlemek icin Play tusuna basin

5 videoyu calistirdiginizi test edin



Test01:

- 1- amazon gidin
- 2 Arama kutusunun solundaki dropdown menuyu handle edip listesini ekrana yazdırın
- 3 dropdown menude 40 eleman olduğunu doğrulayın

Test02

- 1 dropdown menuden elektronik bölümü seçin
- 2 arama kutusuna iphone yazıp aratin ve bulunan sonuç sayısını yazdırın
- 3 sonuc sayısı bildiren yazının iphone içerdigini test edin
- 4 ikinci ürüne relative locater kullanarak tıklayın
- 5 ürünün title'ni ve fiyatını variable'a assign edip ürünü sepete ekleyelim

Test03

- 1 yeni bir sekme açarak amazon anasayfaya gidin
- 2 dropdown'dan bebek bölümüne secin
- 3 bebek puset aratıp bulundan sonuç sayısını yazdırın
- 4 sonuç yazsının puset içerdigini test edin
- 5-üçüncü ürüne relative locater kullanarak tıklayın
- 6-title ve fiyat bilgilerini assign edelim ve ürünü sepete ekleyin

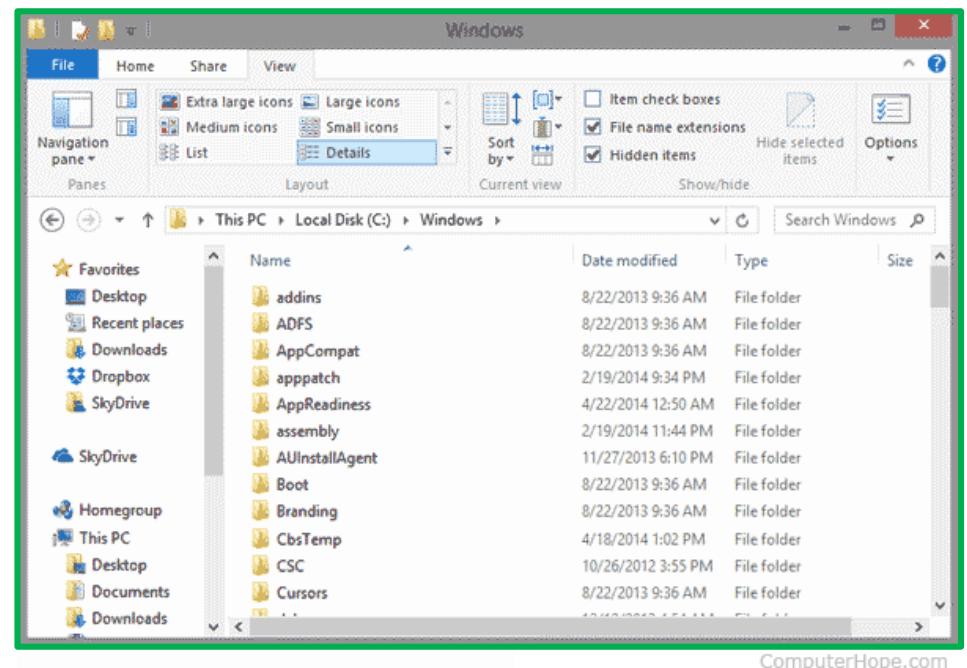
Test 4

- 1-sepetteki ürünlerle eklediğimiz ürünlerin aynı olduğunu isim ve fiyat olarak doğrulayın



File Exist

- Selenium ile windows uygulamalarını test edemiyoruz. Ama test etmek için JAVA kullanabiliriz.
- Bilgisayarımızda bir dosya olup olmadığını(exist) kontrol etmek için Java'yı kullanabiliriz
 - `System.getProperty ("user.dir");` içinde bulunulan klasörün yolunu (Path) verir
 - `System.getProperty ("user.home");` bilgisayarımızda bulunan user klasörünü verir
 - `Files.exists (Paths.get (filePath));` Bilgisayarınızda dosyanın olup olmadığını kontrol eder
- İndirilen bir dosyanın indirme klasörümüzde olup olmadığını kontrol etmek için bu Java konseptini kullanabiliriz





File Download/Exist Class Work

1. Tests packagenin altına bir class oluşturalım : C04_FileDownload
2. <https://the-internet.herokuapp.com/download> adresine gidelim.
3. code.txt dosyasını indirelim
4. Dosyanın başarıyla indirilip indirilmediğini test edelim



File Upload (Dosya Yukleme)

Dosya yükleme işlemini anlamak için önce manuel olarak test yapılmalı.

Daha sonra dosya, dosyanın bulunduğu yer (path) kullanılarak yüklenebilir.

<https://the-internet.herokuapp.com/upload>

File Uploader

Choose a file on your system and then click upload.

Dosya seçilmedi



File Upload Class Work

1. Tests packagenin altina bir class oluşturun : C05_UploadFile
2. <https://the-internet.herokuapp.com/upload> adresine gidelim
3. chooseFile butonuna basalim
4. Yuklemek istediginiz dosyayı secelim.
5. Upload butonuna basalim.
6. “File Uploaded!” textinin goruntulendigini test edelim.



Synchronization - Selenium Waits

- Synchronization(Senkronizasyon), UI (kullanıcı arayüzü) otomasyon testi için çok çok önemlidir.
- Bir sayfanın uygulama sunucusu veya web sunucusu çok yavaşsa veya internet ağı çok yavaşsa, web sayfasındaki öğe (webelement) çok yavaş yüklenir.
- Bu durumda, komut dosyanız (test script) öğeyi bulmaya çalıştığında, öğeler yüklenmez.
- Bu yüzden test komut dosyası(test script) öğeyi bulamaz ve başarısız olur ve **NoSuchElementException** alırız.





Synchronization - Selenium Waits

- » Driver ile cihaz veya internet arasında yasanan senkronizasyon sorunlarını çözmek için sürücümüz(driver) belli şartlar ile bekletmemiz(wait) gereklidir.

1) Java tabanlı wait

Thread.sleep : Javadan gelir ve kodları yazılan sure kadar bekletir. Sure dolduktan sonra alt satırdan işleme devam eder

2) Selenium tabanlı wait'ler

Implicitly Wait: Sayfadaki tüm öğeler için global bir zaman aşımıdır(timeout).

Explicitly Wait: Çoğunlukla belirli öğeler için belirli bir koşul(expected condition) için kullanılır.





IMPLICITLY WAIT

Bir sayfanın yüklenmesi veya sayfadaki her bir öğenin locate edilebilmesi için driver'i bekletir.

Selenium tabanlı wait'lerde verilen sure max. bekleme süresidir, işlem daha önce biterse surenin bitmesi beklenmez, kod çalışmaya devam eder.

Genellikle otomasyon frameworklerinde olası senkronizasyon problemleri için default olarak implicitly wait ile kullanılır.

Implicitly wait'i TestBase class'ımızda kullanıyoruz.

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

Bu kod, driver'in sayfadaki herhangi bir weblement için maximum 10 saniye beklemesini istediğimiz anlamına gelir.



Weblement 10 saniyeden kısa sürede yüklenirse driver bulur ve devam eder. Örneğin, Weblement 3 saniye içinde yüklenirse, driver sadece 3 saniye bekleyecek ve bir sonraki satırı geçecektir.

Weblement 10 saniye içinde yüklenmezse, test case başarısız olur ve **NoSuchElementException** uyarısı verir.



EXPLICIT WAIT

Beklenen bir durum(expected condition) olduğunda explicit wait kullanılır.

Implicitly wait ile cozulebilecek durumlar için explicitly wait kullanımına ihtiyaç yoktur.

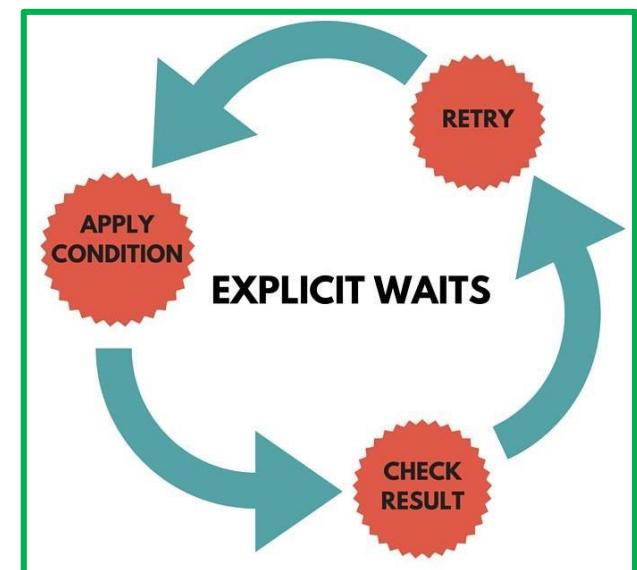
Nadiren karsımıza çıkan ve daha fazla bekleme süresi gerektiren belirli öğeler(web element) için explicitly wait kullanılır.

İlk olarak belirli miktarda bekleme süresi ile wait object create edilir.

```
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(20));
```

Oluşturduğumuz wait objesi sayesinde WebDriverWait class'ından until() method'u ile birlikte ExpectedConditions Class'ını kullanabiliriz. Örneğin WebElement'i locate etmek için wait object'i kullanma

```
WebElement element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("...")));
```





EXPLICIT WAIT Expected Conditions

- 1.alertIsPresent()
- 2.elementSelectionStateToBe()
- 3.elementToBeClickable()
- 4.elementToBeSelected()
- 5.frameToBeAvailableAndSwitchToIt()
- 6.invisibilityOfTheElementLocated()
- 7.invisibilityOfElementWithText()
- 8.presenceOfAllElementsLocatedBy()
- 9.presenceOfElementLocated()
- 10.textToBePresentInElement()
- 11.textToBePresentInElementLocated()
- 12.textToBePresentInElementValue()
- 13.titleIs()
- 14.titleContains()
- 15.visibilityOf()
- 16.visibilityOfAllElements()
- 17.visibilityOfAllElementsLocatedBy()
- 18.visibilityOfElementLocated()



Explicit Wait Class Work

1. Bir class olusturun : WaitTest
2. Iki tane metod olusturun : implicitWait() , explicitWait()

Iki metod icin de asagidaki adimlari test edin.

3. https://the-internet.herokuapp.com/dynamic_controls adresine gidin.
4. Remove butonuna basin.
5. “It’s gone!” mesajinin goruntulendigini dogrulayin.
6. Add buttonuna basin
7. It’s back mesajinin gorundugunu test edin



Explicit Wait Class Work

1. Bir class olusturun : EnableTest
2. Bir metod olusturun : isEnabled()
3. https://the-internet.herokuapp.com/dynamic_controls adresine gidin.
4. Textbox'in etkin olmadigini(enabled) doğrulayın
5. Enable butonuna tıklayın ve textbox etkin oluncaya kadar bekleyin
6. "It's enabled!" mesajinin goruntulendigini doğrulayın.
7. Textbox'in etkin oldugunu(enabled) doğrulayın.



Actions Class Home Work

1. ["http://webdriveruniversity.com/Actions"](http://webdriveruniversity.com/Actions) sayfasina gidin
2. "Hover over Me First" kutusunun ustune gelin
3. "Link 1" e tiklayin
4. Popup mesajini yazdirin
5. Popup'i tamam diyerek kapatin
6. "Click and hold" kutusuna basili tutun
7. "Click and hold" kutusunda cikan yaziyi yazdirin
8. "Double click me" butonunu cift tiklayin



Iframe Home Work

1. "<http://webdriveruniversity.com/IFrame/index.html>" sayfasina gidin
2. "Our Products" butonuna basin
3. "Cameras product"i tiklayın
4. Popup mesajini yazdirin
5. "close" butonuna basin
6. "WebdriverUniversity.com (IFrame)" linkini tiklayın
7. "<http://webdriveruniversity.com/index.html>" adresine gittigini test edin



Window Handle Home Work

- 1."<http://webdriveruniversity.com/>" adresine gidin
- 2."Login Portal" a kadar asagi inin
- 3."Login Portal" a tiklayin
- 4.Diger window'a gecin
- 5."username" ve "password" kutularina deger yazdirin
- 6."login" butonuna basin
- 7.Popup'ta cikan yazinin "validation failed" oldugunu test edin
- 8.Ok diyerek Popup'i kapatın
- 9.Ilk sayfaya geri donun
- 10.Ilk sayfaya donuldugunu test edin



Cookies

Çerezler, belirli kullanıcıları tanımlamak ve bu kullanıcıların göz atma deneyimini iyileştirmek için kullanıcının bilgisayarı ile web sunucusu arasında takas edilen, kullanıcı adı ve parola gibi küçük veri parçalarını içeren dosyalarıdır.

İnternette gezinirken ziyaret ettiğiniz web sayfaları, bilgisayarınıza ve telefonunuza küçük bilgi dosyaları kaydeder. Bu dosyalar telefon veya bilgisayarınızın hafızasında saklanır. Daha sonra aynı siteleri ziyaret ettiğinizde bu kayıtlı bilgi dosyaları sayesinde siteler sizi tanıyabilir.

Bilgileriniz bu dosyalara yazıldığından dolayı tekrar aynı web sayfalarını ziyaret ettiğinizde bilgilerinizi yeniden girmeye gerek duymazsınız.



Cookies, kişisel bilgiler de dahil olmak üzere birçok bilgiyi içerebilir. Web siteleri, ancak sizin izin verdığınız bilgilere erişebilir. Bu web sayfaları, sizin vermediğiniz bilgilere erişemez ya da bilgisayarlarınızdaki diğer dosyaları görüntüleyemez.



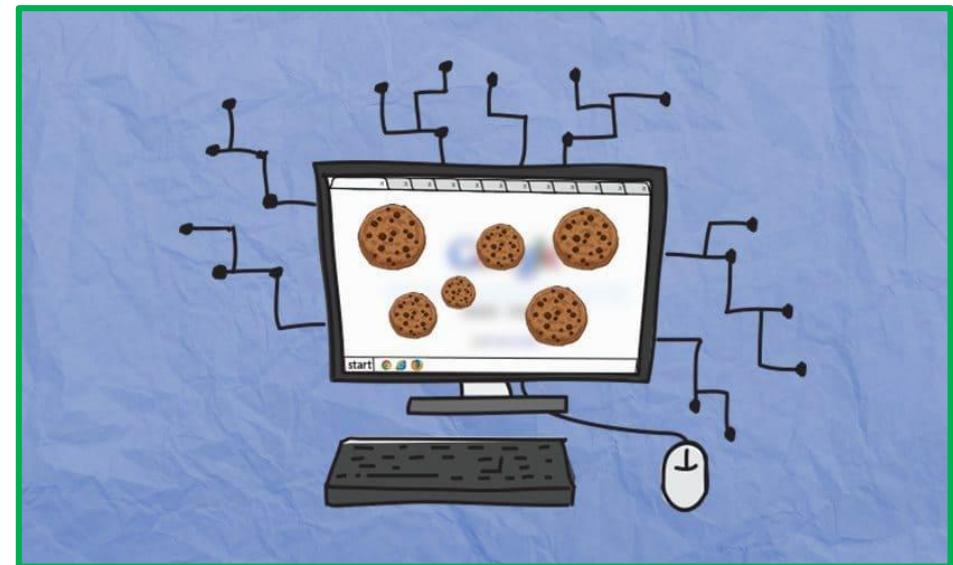
Cookies

Birkaç farklılıkla, siber dünyadaki çerezlerin **oturum çerezleri** ve **kalıcı çerez** olmak üzere iki çeşidi vardır.

Oturum çerezleri yalnızca bir web sitesinde gezinirken kullanılır. Bunlar rastgele erişimli bellekte saklanır ve hiçbir zaman sabit sürücüye yazılmaz. Oturum sona erdiğinde oturum çerezleri otomatik olarak silinir.

Kalıcı çerezler bir bilgisayarda sonsuza kadar kalır ancak birçoğunun bir son kullanma tarihi olup bu tarihe gelindiğinde otomatik olarak kaldırılırlar.

Üçüncü taraf çerezler daha sıkıntılıdır. Bunlar, genellikle kullanıcıların halihazırda gezindiği web sayfalarındaki reklamlarla bağlantılı olduklarından bu sayfalardan farklı web siteleri tarafından oluşturulur.





Cookies

Selenium ile cookies otomasyonu yapabiliriz.

Driver.manage(). method'u ile cookie'leri

- listeleyebilir
- Isim ile cagirabilir
- Yeni cookie ekleyebilir
- Var olanlari ismi silebilir
- Var olan tum cookie'leri silebiliriz

```
driver.manage().cook
  m addCookie(Cookie cookie) void
  m getCookies() Set<Cookie>
  m deleteAllCookies() void
  m deleteCookie(Cookie cookie) void
  m getCookieNamed(String name) Cookie
  m deleteCookieNamed(String name) void
  ...
  Ctrl+Down and Ctrl+Up will move caret down and up in the editor Next Tip
```



Cookies Class Work

Yeni bir class olusturun : cookiesAutomation

1- Amazon anasayfaya gidin

2 tum cookie'leri listeleyin

3 Sayfadaki cookies sayisinin 5'den buyuk oldugunu test edin

4 ismi i18n-prefs olan cookie degerinin USD oldugunu test edin

5- ismi "en sevdigim cookie" ve degeri "cikolatali" olan bir cookie olusturun ve sayfaya ekleyin

6 eklediginiz cookie'nin sayfaya eklendagini test edin

7 ismi skin olan cookie'yi silin ve silindigini test edin

8 tum cookie'leri silin ve silindigini test edin



Web Tables

Bir Web sayfasında webelementleri düzenlemek için HTML tablo oluşturabiliriz

HTML tablo farklı sekillerde oluşturulabilir ama genellikle `<table>`, `<thead>`, `<tbody>`, `<tr>`, `<th>` ve `<td>` attribute'leri kullanılır.

Amazon Music Stream millions of songs	Amazon Advertising Find, attract, and engage customers	Amazon Drive Cloud storage from Amazon	6pm Score deals on fashion brands
Sell on Amazon Start a Selling Account	Amazon Business Everything For Your Business	AmazonGlobal Ship Orders Internationally	Home Services Experienced Pros Happiness Guarantee
Audible Listen to Books & Original Audio Performances	Book Depository Books With Free Delivery Worldwide	Box Office Mojo Find Movie Box Office Data	ComiXology Thousands of Digital Comics
Goodreads Book reviews & recommendations	IMDb Movies, TV & Celebrities	IMDbPro Get Info Entertainment Professionals Need	Kindle Direct Publishing Indie Digital & Print Publishing Made Easy
Zappos Shoes & Clothing	Ring Smart Home Security Systems	eero WiFi Stream 4K Video in Every Room	Neighbors App Real-Time Crime & Safety Alerts

```
<table> ==>>>table
  <thead> ==>>>header
    <tr> ==>>>table rows
      <th> ==>>>table header dat
      </tr>
    </thead>
    <tbody> ==>>>table body
      <tr> ==>>>table row
        <td> </td> ==>>>table data-cell
        </tr>
      </tbody>
</table>
```



JUnit

Web Tables

<https://www.hotelmycamp.com>

ID	UserName	Email	Name Surname	BirthDate
1				<input type="button" value=""/>
2	manager	manager@st.com	manager	22.02.2020
3	customerservice	customerservice@st.com	customerservice	14.01.2020
154	123456	kristalhotel@gmail.com	cevdet oz	12.12.2020

table

thead

tbody

data

row

row

row

row

row

```
<table class="table table-striped table-bordered table">
  <thead>...</thead>
  <tbody>
    <tr role="row" class="odd"> ← row
      <td>2</td>
      <td class="sorting_1">manager</td> ← red box
      <td>manager@st.com</td>
      <td>manager</td>
      <td>22.02.2020</td>
      <td>(545) 345-3453</td>
      <td>Manager</td>
    </tr>
    <tr role="row" class="even"> ← row
      <td>3</td>
      <td class="sorting_1">customerservice</td>
      <td>customerservice@st.com</td>
      <td>customerservice</td>
      <td>14.01.2020</td>
      <td>(543) 545-4354</td>
      <td>CustomerService</td>
    </tr>
    <tr role="row" class="odd">...</tr> ← row
    <tr role="row" class="even">...</tr> ← row
    <tr role="row" class="odd">...</tr> ← row
```



Web Tables Class Work

- Bir class oluşturun : C02_WebTables
- login() metodun oluşturun ve oturum açın.
- <https://www.hotelmycamp.com> admin/HotelRoomAdmin adresine gidin
 - Username : manager
 - Password : Manager1!
- table() metodunu oluşturun
 - Tüm table body'sinin boyutunu(sutun sayısı) bulun. /tbody
 - Table'daki tüm body'l ve başlıklarını(headers) konsolda yazdırın.
- printRows() metodunu oluşturun //tr
 - table body'sinde bulunan toplam satır(row) sayısını bulun.
 - Table body'sinde bulunan satırları(rows) konsolda yazdırın.
 - 4.satırındaki(row) elementleri konsolda yazdırın.



Web Tables Class Work

- Bir class oluşturun : D18_WebTables
- login() metodun oluşturun ve oturum açın.
- <https://qa-environment.concorthotel.com/admin/HotelRoomAdmin> adresine gidin
 - Username : manager ○ Password : Manager2!
- table() metodunu oluşturun
 - Tüm table body'sinin boyutunu(sutun sayisi) bulun. /tbody
 - Table'daki tüm body'i ve başlıklarını(headers) konsolda yazdırın.
- printRows() metodunu oluşturun //tr
 - table body'sinde bulunan toplam satır(row) sayısını bulun.
 - Table body'sinde bulunan satırları(rows) konsolda yazdırın.
 - 4.satırda(row) elementleri konsolda yazdırın.
- printCells() metodunu oluşturun //td
 - table body'sinde bulunan toplam hücre(cell) sayısını bulun.
 - Table body'sinde bulunan hücreleri(cells) konsolda yazdırın.
- printColumns() metodunu oluşturun
 - table body'sinde bulunan toplam sutun(column) sayısını bulun.
 - Table body'sinde bulunan sutunları(column) konsolda yazdırın.
 - 5.column daki elementleri konsolda yazdırın.



Web Tables Class Work

WebTables class'ini kullanın.

1. Bir metod oluşturun : `printData(int row, int column);`
 - a. Satır(row) ve sütun(column) numarasını girdiğinizde, `printData` metodu bu hücredeki(cell) veriyi yazdırmalıdır.
2. Baska bir Test metodu oluşturun: `printDataTest();`
 - a. Ve bu metodу `printData()` methodunu cagirmak icin kullanın.
 - b. Örnek: `printData (3,5); => 3. satır, 5. Sütundaki veriyi yazdırmalıdır`
 - c. yazdirilan datanin olmasi gereken dataya esit oldugunu test edin



Web Tables Homework

Bir Class olusturun D19_WebtablesHomework

1. “<https://demoqa.com/webtables>” sayfasina gidin
2. Headers da bulunan department isimlerini yazdirin
3. sutunun basligini yazdirin
4. Tablodaki tum datalari yazdirin
5. Tabloda kac cell (data) oldugunu yazdirin
6. Tablodaki satir sayisini yazdirin
7. Tablodaki sutun sayisini yazdirin
8. Tablodaki 3.kolonu yazdirin
9. Tabloda "First Name" i Kierra olan kisinin Salary'sini yazdirin
10. Page sayfasinda bir method olusturun, Test sayfasindan satir ve sutun sayisini girdigimde bana datayi yazdirlasın



Excel'in Yapısı

- Excel için daha önce inceledigimiz Web Table yapısına benzer bir yapı vardır.
- Excel'de bir hücredeki bilgiye ulaşmak için dosya/sayfa/satır/sütun sırasıyla ilerlemeliyiz
- Excel ile ilgili otomasyonda web table'da olduğu gibi sütun yapısı yoktur, ihtiyaç duyarsak kodla sütunu elde edebiliriz ancak bir dataya ulaşmak için sütun kullanamayız

Workbook excel dosyamız

Sheet Her açık excel sekmesi (Sheet1, Sheet2, etc)

Row(satır) Java, yalnızca içerisinde veri varsa satırları sayar.
Default olarak, Java perspektifinden satır sayısı 0'dır

Cells (hücre) Java her satıra bakar ve yalnızca hücrede veri varsa hücre sayısını sayar.

	A	B	C	D
1	Ülke (İngilizce)	Başkent (İngilizce)	Ülke (Türkçe)	Başkent (Türkçe)
2	Afghanistan	Kabul	Afganistan	Kabil
3	Albania	Tirana	Arnavutluk	Tiran
4	Algeria	Aljiers	Cezayir	Cezayir
5	Andorra	Andorra la Vella	Andorra	Andorra la Vella
6	Angola	Luanda	Angola	Luanda
7	Antigua & Barbuda	Saint John's	Antigua ve Barbuda	Saint John's
8	Argentina	Buenos Aires	Arjantin	Buenos Aires
9	Armenia	Yerevan	Ermenistan	Erivan
10	Australia	Canberra	Australya	Canberra
11	Austria	Vienna	Avusturya	Viyana
12	Azerbaijan	Baku	Azerbaycan	Bakü
13	The Bahamas	Nassau	Bahamalar	Nassau
14	Bahrain	Manama	Bahreyn	Manama
15	Bangladesh	Dhaka	Bangladeş	Dakka
16	Barbados	Bridgetown	Barbados	Bridgetown
17	Belarus	Minsk	Belarus	Minsk
18	Belgium	Brussels	Belçika	Brüksel
19	Belize	Belmopan	Belize	Belmopan
20	Benin	Porto Novo	Benin	Porto Novo
21	Bhutan	Thimphu	Butan	Thimphu
22	Bolivia	La Paz	Bolivya	La Paz
23	Bosnia & Herzegovina	Sarajevo	Bosna-Hersek	Saraybosna
24	Botswana	Gaborone	Botswana	Gaboron
25	Brazil	Brasilia	Brezilya	Brasilia
26	Brunei	Bandar Seri Begawan	Brunei	Bandar Seri Begawan
27	Bulgaria	Sofia	Bulgaristan	Sofya



- Apache POI, microsoft ofis dokumanlarına erişmek için kullanılan Java API'sidir.
- Poi.apache.com sayfasından official dokumanlar incelenebilir.
- Excel kullanmak için;

```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>4.1.2</version>
</dependency>
```



Read Excel Class Work

1. apache poi dependency'i pom file'a ekleyelim
2. Java klasoru altinda **resources** klasoru olusturalim
3. Excel dosyamizi resources klasorune ekleyelim
4. **excelAutomation** isminde bir package olusturalim
5. ReadExcel isminde bir class olusturalim
6. **readExcel()** method olusturalim
7. Dosya yolunu bir String degiskene atayalim
8. FileInputStream objesi olusturup,parametre olarak dosya yolunu girelim
9. Workbook objesi olusturalim,parameter olarak inputStream objesini girelim
10. **WorkbookFactory.create(inputStream)**
11. Worksheet objesi olusturun **workbook.getSheetAt(index)**
12. Row objesi olusturun **sheet.getRow(index)**
13. Cell objesi olusturun **row.getCell(index)**



Read Excel Class Work

Yeni bir test method olusturalim readExcel2()

- 1.satirdaki 2.hucreye gidelim ve yazdiralim
- 1.satirdaki 2.hucreyi bir string degiskene atayalim ve yazdiralim
- 2.satir 4.cell'in afganistan'in baskenti oldugunu test edelim
- Satir sayisini bulalim
- Fiziki olarak kullanilan satir sayisini bulun
- Ingilizce Ulke isimleri ve baskentleri bir map olarak kaydedelim



Write Excel Class Work

- 1) Yeni bir Class olusturalim WriteExcel
- 2) Yeni bir test method olusturalim writeExcelTest()
- 3) Adimlari takip ederek 1.satira kadar gidelim
- 4) 4.hucreye yeni bir cell olusturalim
- 5) Olusturdugumuz hucreye "Nufus" yazdiralim
- 6) 2.satir nufus kolonuna 1500000 yazdiralim
- 7) 10.satir nufus kolonuna 250000 yazdiralim
- 8) 15.satir nufus kolonuna 54000 yazdiralim
- 9) Dosyayı kaydedelim
- 10) Dosyayı kapatelim



JUnit

Get Screenshot / Tüm Sayfa

Selenium 3 ile tüm sayfanın fotoğrafını alabiliyorduk, Selenium 4 ile gelen güzel özelliklerden bir tanesi de istedigimiz WebElement'in fotoğrafını almamıza imkan tanımıştır.

1.Adım : Bir TakeScreenshot objesi oluşturup driver'imizi TakeScreenshot'a cast yapalım

```
TakesScreenshot ts=(TakesScreenshot) driver;
```

2.Adım : kaydettigimiz ekran görüntüsünü projede istedigimiz yere kaydedebilmek için path ile yeni bir File olusturalım

```
File tumResim= new File( pathname: "target/ekranResmi/tumSayfa");
```



3.Adım : Takescreenshot objesini kullanarak getScreenshotAs() methodunu çalıştırıyalım ve gelen resmi geçici bir file'a assign edelim

```
File resimGecici=tss.getScreenshotAs(OutputType.FILE);
```

4.Adım : Kaydettigimiz görüntuyu, saklamak istedigimiz dosyaya kopyalayalım

```
FileUtils.copyFile(resimGecici,tumResim);
```



JUnit

Get Screenshot / Spesific Webelement

1.Adım : Webelementi locate edelim

```
WebElement logoElementi=driver.findElement(By.id("nav-logo-sprites"));
```

2.Adım : kaydettigimiz ekran goruntusunu projede istedigimiz yere kaydedebilmek icin path ile yeni bir File olusturalim

```
File logoResmi=new File( pathname: "target/ekranResmi/element");
```



3.Adım : Webelement uzerinden getScreenshotAs() methodunu calistiralim ve gelen resmi olusturdugumuz gecici file'a assign edelim

```
File logoGecici=logoElementi.getScreenshotAs(OutputType.FILE);
```

4.Adım : Gecici resmi, saklamak istedigimiz dosyaya kopyalayalim

```
FileUtils.copyFile(logoGecici,logoResmi);
```



Get Screenshot Class Work

Yeni bir class olusturun : amazonNutellaSearch

1amazon anasayfaya gidin

2amazon anasayfaya gittiginizi test edin ve tum sayfanin goruntusunu kaydedin

3- Nutella icin arama yapin

4- sonucun Nutella icerdigini test edin ve ilk urunun goruntusunu alin

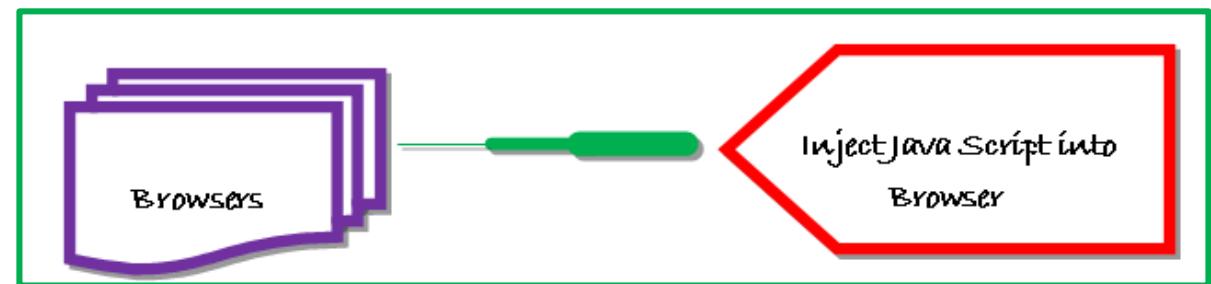


JUnit

Js Executors

Selenium ile web elementlerinin kontrollerini sağlarken selenium komutları yetersiz kalabilir veya sorunlarla karşılaşabiliriz.

Bu durumlarda alternatif olarak üstesinden gelmek için JavascriptExecutor class'ını dahil edebiliriz.



JavaScript HTML kodlara direk erişip yönetebilen bir script dili olduğundan bize çok fazla kolaylık sağlayabilir.



JUnit

Js Executors

1.Adım : JavascriptExecutor kullanmak için ilk adım olarak driver'imizi JavascriptExecutor'a cast edelim

```
JavascriptExecutor jse=(JavascriptExecutor)driver;
```

2.Adım : Kullanicagimiz WebElement'i locate edelim

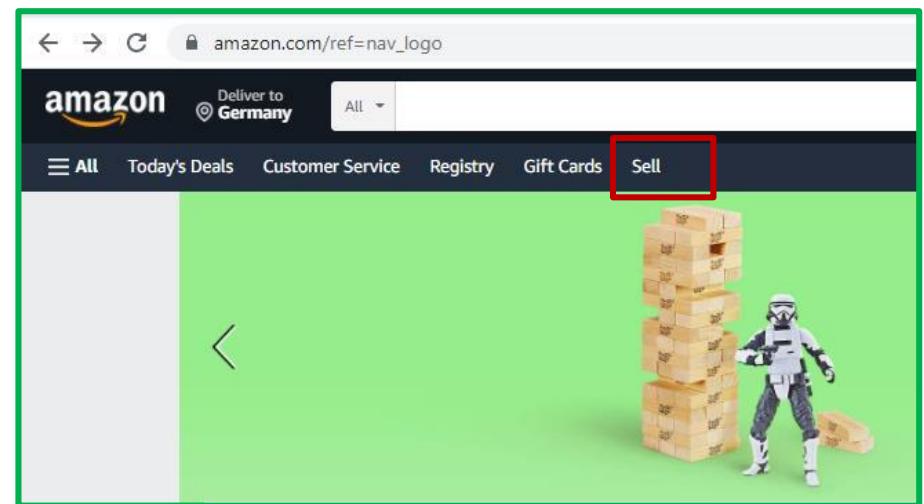
```
WebElement sellElementi= driver.findElement(By.xpath("//a[text()='Sell']"));
```

3.Adım : Js driver ile executeScript method'unu calistiralim,
icine parameter olarak ilgili script ve webelement'i yazalim

```
jse.executeScript( script: "ilgili script", ...args: "web element");
```

Ornegin Sell elementine click yapmak icin

```
jse.executeScript( script: "arguments[0].click();", sellElementi);
```





JUnit

Js Executors ScroollIntoView

```
@Test
public void test02() throws InterruptedException {

    JavascriptExecutor je= (JavascriptExecutor) driver;

    driver.get("https://www.koalaresorthotels.com");

    Thread.sleep( millis: 3000);

    WebElement ourRooms = driver.findElement(By.xpath("//*[.= 'Our Rooms']"));

    je.executeScript( script: "arguments[0].scrollIntoView(true);",ourRooms);

    Thread.sleep( millis: 8000);

}
```



25/7 Front Desk



Restaurant Bar



Transfer Services



Spa Suites

Our Rooms





Js Executors ScroolIntoView Class Work

- 1 Yeni bir class olusturun : Scroollnto
 - 2 hotelmycamp anasayfasina gidin
 - 3 2 farkli test method'u olusturarak actions clası ve Js Executor kullanarak asagidaki oda turlerinden ust sira ortadaki odayi tiklayın
 - 4 istediginiz oda inceleme sayfasi acildigini test edin

Double		Double		Double	
\$700.00	per night	\$700.00	per night	\$700.00	per night
View Room Details	→	View Room Details	→	View Room Details	→
Double		Double		Double	
\$700.00	per night	\$700.00	per night	\$700.00	per night
View Room Details	→	View Room Details	→	View Room Details	→