



# Selenium giris

SELENIUM GIRIS

OTOMASYON TEST NEDIR?

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.

## About Selenium

Selenium is a suite of tools for automating web browsers.

**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.

- Selenyum, 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 NEDİR

- IntelliJ IDEA 2000 Yılında Kurulmuş Olan JetBrains İsimli Yazılım Firmasına Ait Olan, Popüler Bir Java Editörüdür.. (ide)
- IntelliJ IDEA'nın her yönü, geliştirici üretkenliğini en üst düzeye çıkarmak için tasarlanmıştır.
- Akıllı kodlama yardımı ve ergonomik tasarım ile, kod yazimini yalnızca verimli değil, aynı zamanda keyifli hale getirir.
- Bircok Framework ve plugin ile calisma imkani verir
- Akilli tamamlama ozelligi ile kod yazimini oldukca kolaylastirir
- IDE ihtiyaclarinizi 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 işlemidir.

- 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 Strory icin Positive ve Negative Test(ler) yapılmalıdır
- Test, musteri/isletme ihtiyaclarini karsilamak icin yapilir.
- Bir uygulamayi test etmek icin onceden belirlenmis user storyler (kullanici hikayeleri) ve tanimlanmis acceptance criterias (kabul kriterleri) dikkate alinir.



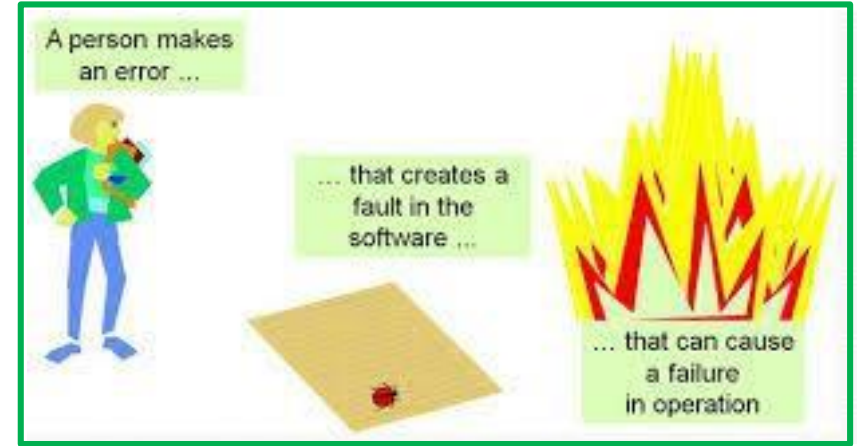


# SOFTWARE TESTING NEDEN ÖNEMLİDİR ?

- 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 amaçlarıyla ürün müşteriye sunulmadan önce test edilmesi gerekmektedir.



**KISACA :** Testing saves money and lives .

Testing para kayıplarını onlar ve hayat kurtarır. (Uçak kazaları 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, çalıştırılan test paketini kaydetmek ve gerektiğinde yeniden çalıştırmak mümkündür. Test paketi otomatik hale getirildikten sonra hiçbir insan müdahalesi gerekmesizin programlandığı zamanda çalışabilir.
- 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 sayfayi kapatir



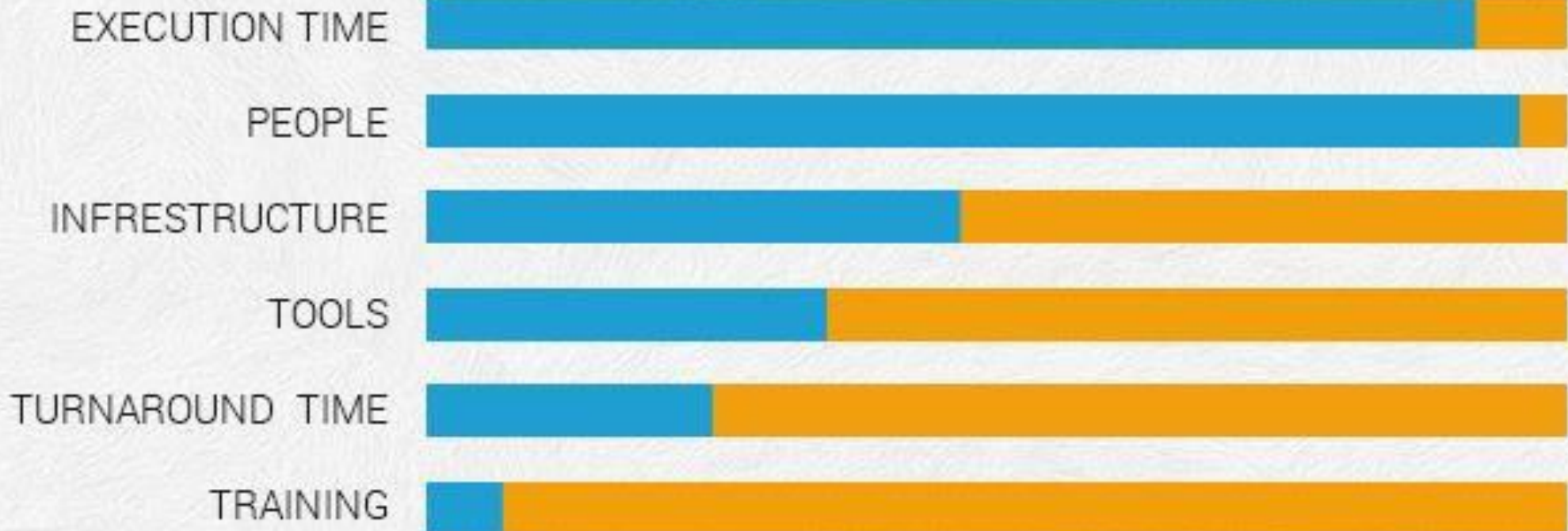


# MANUAL TESTING vs AUTOMATION TESTING

MANUAL TESTING

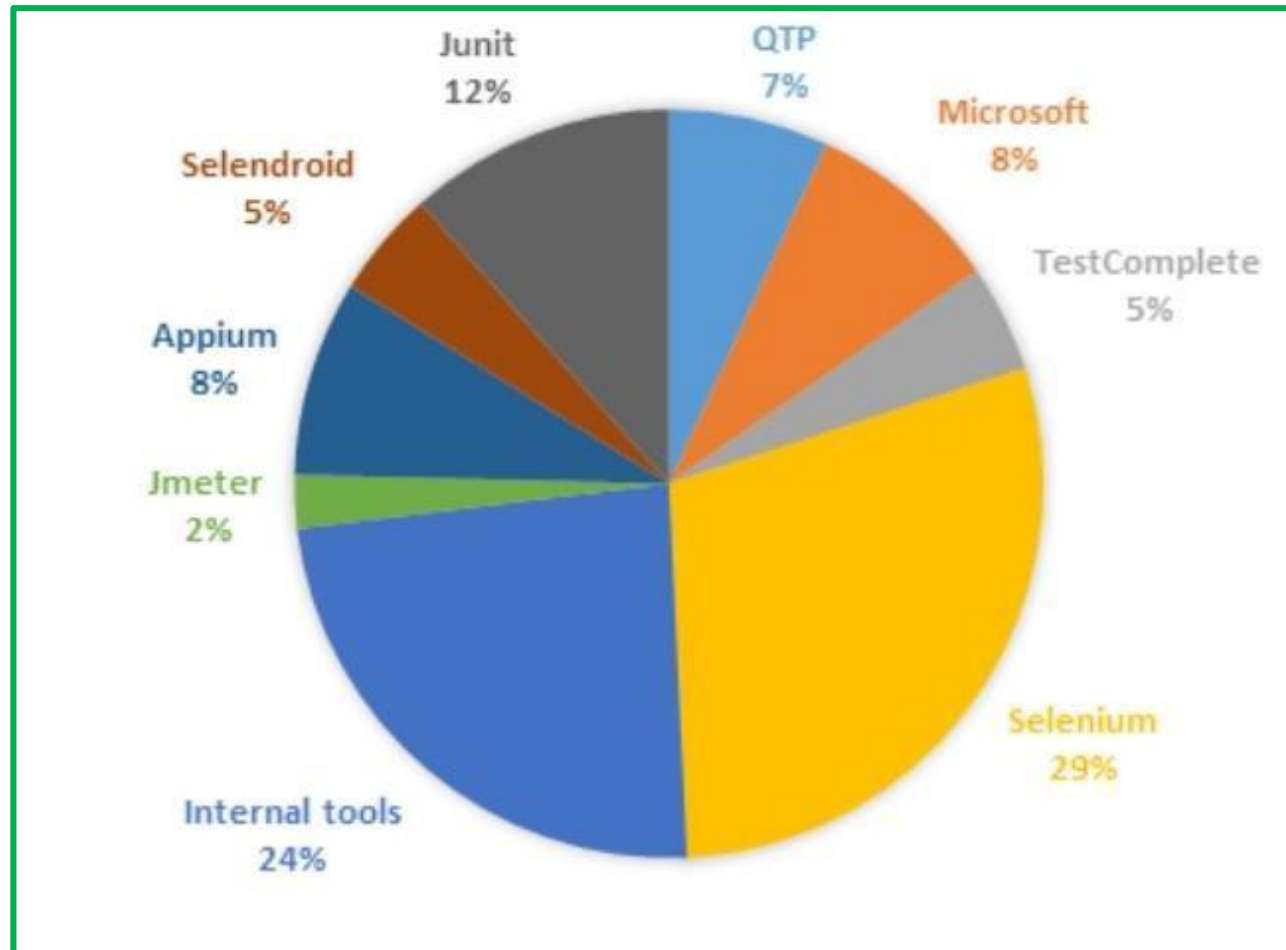
VS

AUTOMATION TESTING





# EN COK KULLANILAN TEST TOOL'LARI





# Basarılı Bir Otomasyon Testi için Yapılması Gerekenler

- Doğru otomasyon araçlarını(tools) seçin. Genellikle şirketlerin kullandıkları tool'ları vardır, bazı şirketler de sizin isteginize göre tool secebilirler.
- 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 bozmadigindan emin olun.
- Testlerinizde hata olabilecegini göz önünde bulundurun. Bir BUG'i rapor etmeden önce mutlaka testinizi gözden geçirin
- Her zaman test ekibinizle, özellikle team liderleri ile iyi iletişim kurun.





# SELENIUM BİLESENLERİ

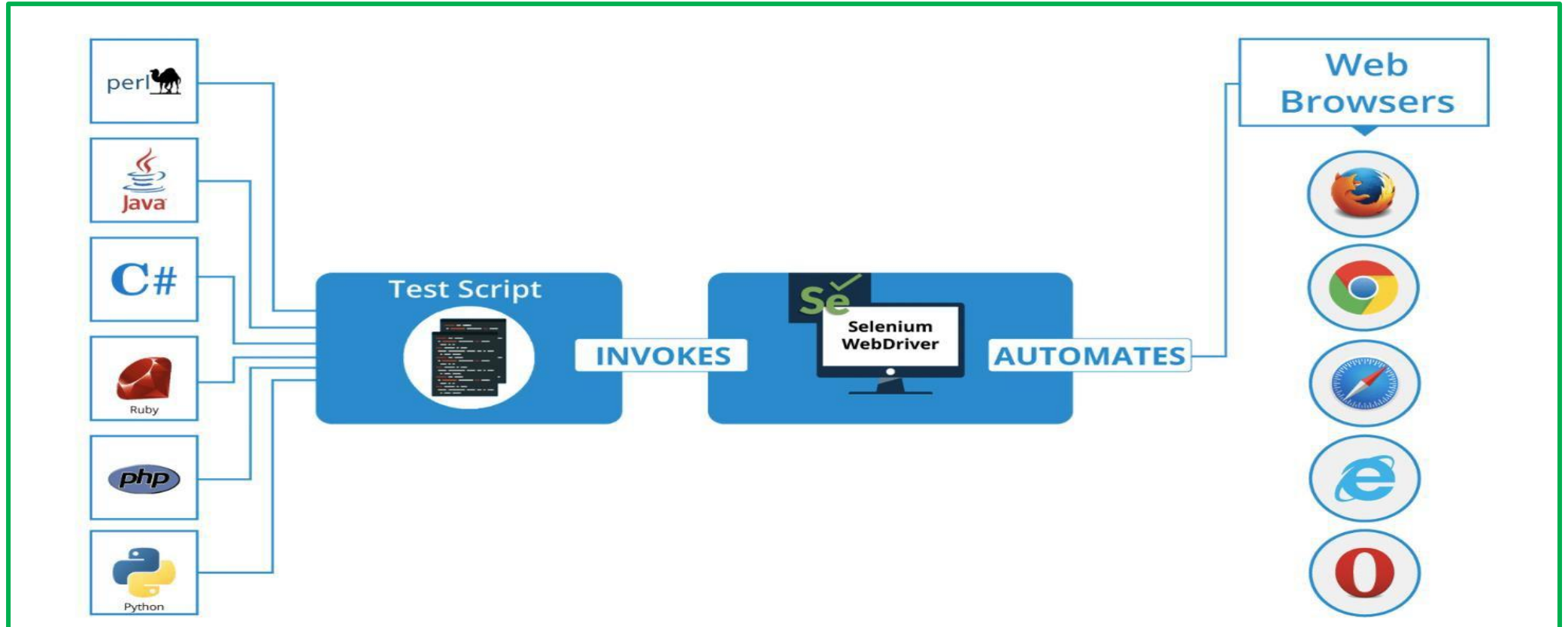
Selenium'un dört bileşeni vardır;



- 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 NELERDİR?



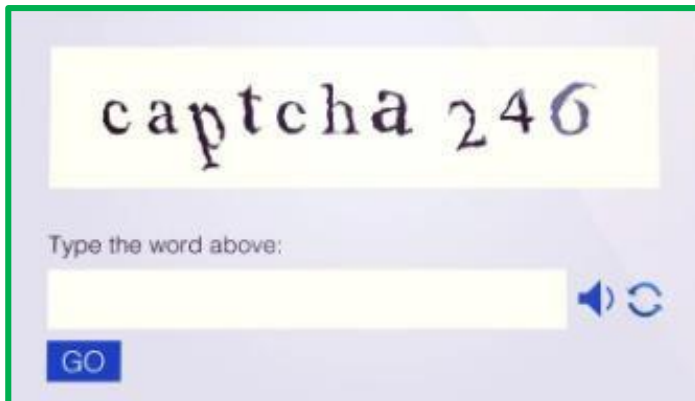
- 1 ) Ücretsiz ve açık kaynaklıdır. ( Open source )
- 2 ) Bir çok programlama dilini destekler (Java, Python, PHP, C#, Ruby vs.)
- 3 ) Çoklu işletim sistemleriyle çalışır.  
Multiple operating systems  
(Windows, MacOS, Linux)
- 4 ) Birden çok tarayıcı ile çalışır.  
Multiple browsers  
(IE, Safari,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





# SELENIUM'UN DEZAVANTAJLARI NELERDIR?

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



## SELENIUM'UN YAPAMADIKLARI

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



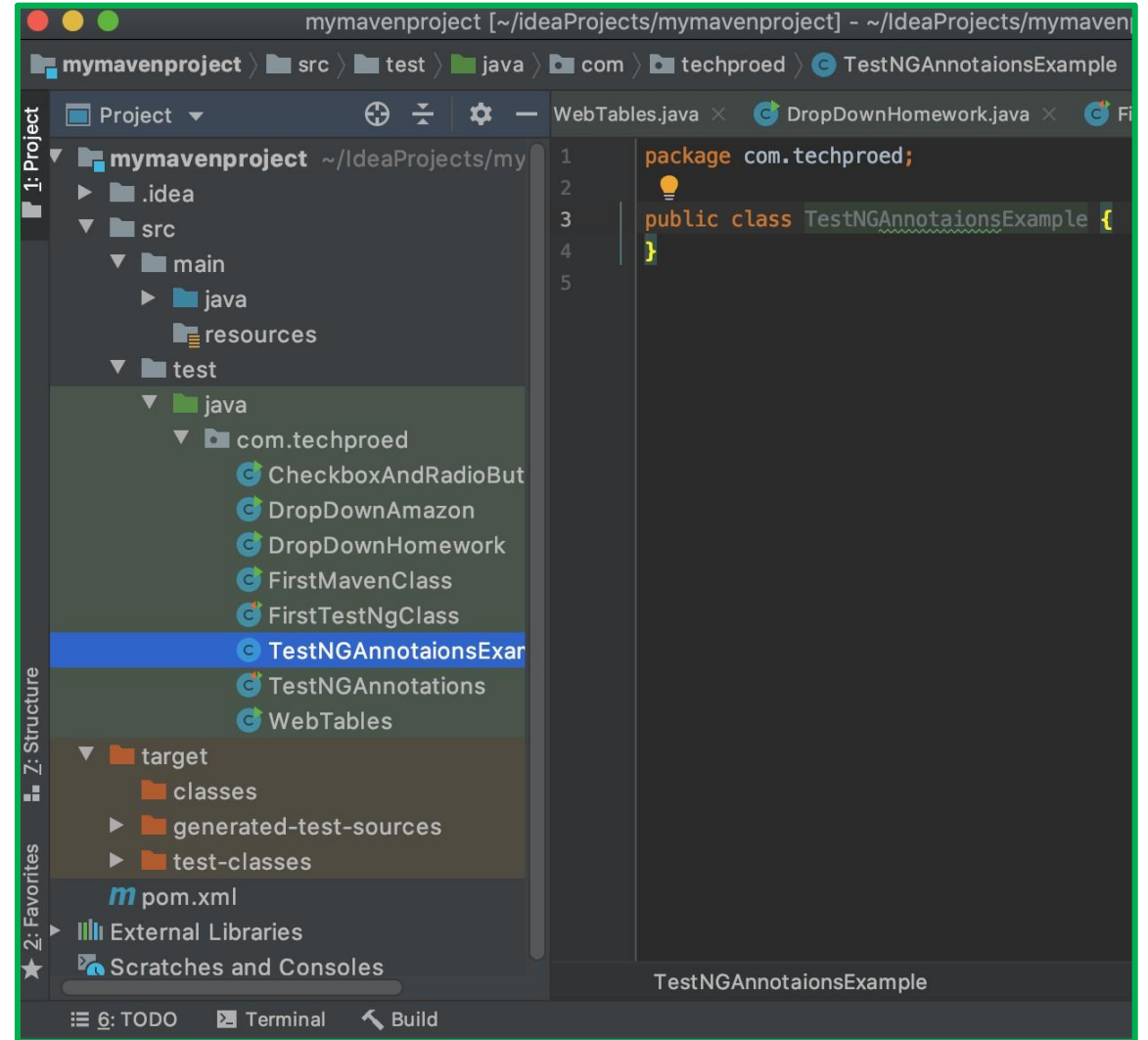


# TEST AUTOMATION FRAMEWORK NEDİR?

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

( Isinizi rahat yapabilmeniz için hazırlanmış çalışma ortamınızdır )

- Kullanılan Tool'lar  
Java,  
Eclipse/IntelliJ,  
Selenium,  
JUnit/  
TestNG,  
Cucumber
- Test Yapılan Layer'lar  
UI , API , DataBase





# 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 tıklayalım  
Chrome'un kendi sayfasına gidip Current stable release'i tıklayıp size uygun olanı download edin  
\*\*\* buradaki surum ile bilgisayarınızdaki Chrome surumunun aynı olduğundan emin olun
- 4) src altında resources director'si oluşturun
- 5) Bu klasör altında drivers ve libraries klasörleri oluşturun
- 6) İndirdiğimiz chromedriver'i drivers klasörüne, selenium-java dosyasını ise libraries klasörüne çıkartalım
- 7) IntelliJ 'de yeni project / package / class oluşturalım ve class içinde main method oluşturalım
- 8) File/Project Structure/Modules/Dependencies kısmından jar dosyalarını yükleyelim



# WebDriver Oluşturulması

1. Yeni bir class oluşturun: class name  $\Rightarrow$  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. Oluşturduğumuz driver objesi ile WebDriver class'ından 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. Sayfayi 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 sayfayi 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. Sayfayi Refresh(yenile) yapalim
7. Sayfayi kapatalim / Tum sayfalari 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. Sayfayi simge durumuna getirin
5. simge durumunda 3 saniye bekleyip sayfayi maximize yapin
6. Sayfanin konumunu ve boyutlarini maximize durumunda yazdirin
7. Sayfayi fullscreen yapin
8. Sayfanin konumunu ve boyutlarini fullscreen durumunda yazdirin
9. Sayfayi 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));
```

Isteneen 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 ayarlayin
5. Sayfanin sizin istediginiz konum ve boyuta geldigini test edin
8. Sayfayi 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. Sayfayi yenileyin
- 8.Sayfayi tam sayfa (maximize) yapin
- 9.Browser'i kapatın



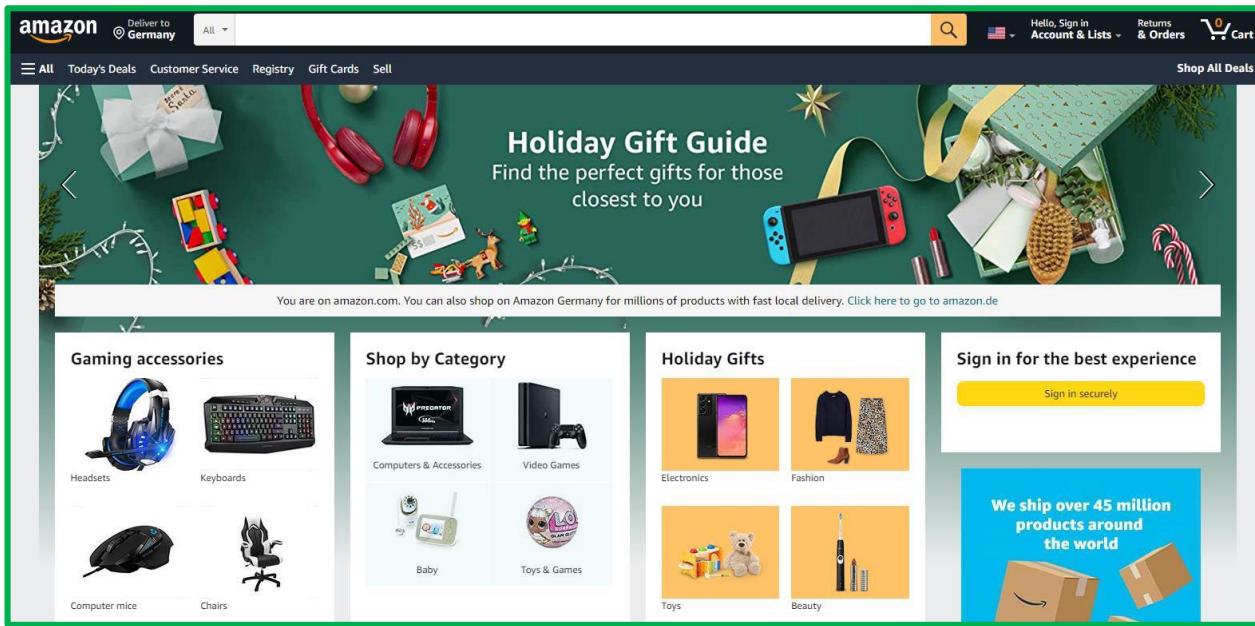
# Tekrar Testi

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, değilse doğru URL'yi yazdırın
11. Sayfayı kapatın



# WEBELEMENTS

- Web sayfasında kullanılan etkileşimli 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 the Amazon website interface. The top navigation bar includes the Amazon logo, a search bar, and links for 'Hello, Sign in', 'Account & Lists', and 'Returns & Orders'. Below this, there are links for 'Select your address', 'Today's Deals', 'Best Sellers', 'Find a Gift', 'Customer Service', and 'New'. The main banner features a red background with a pair of black earbuds and the text 'e Alexa on the go' and 'buds'. The browser's developer tools are open at the bottom, with the 'Elements' tab selected. A context menu is overlaid on the page, showing options like 'Emoji & Symbols', 'Undo', 'Redo', 'Cut', 'Copy', 'Paste', 'Paste and Match Style', 'Select All', 'Language Settings', 'Writing Direction', 'Inspect' (highlighted with a red circle), and 'Speech'. The 'Elements' panel shows the HTML structure of the page, with the search bar element selected.

```
<div class="nav-fill">
  <script type="text/javascript">window.navmet.tmp+=new Date();</script>
  <div id="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 class="nav-fill">
        <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 class="nav-input" dir="auto" tabindex="19"> == $0
        </div>
      </div>
    </div>
  </div>
</div>
```



- 
- SELENIUM LOCATORS**



# LOCATORS (YER BULUCU-KONUM BELİRLEYİCİ)

```
<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 ) Tanımladığımız web elementin yerini driver'in bulabilmesi için findElement (Locator) method'unu kullanırız.

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

- 2 ) findElement (Locator) method'unun içine parameter olarak yazacağımız Locator'in 8 locator'dan hangisi olduğunu belirtmek için de By.LocatorTuru("LocateBilgisi"); kullanılır

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

- 3 ) Locate ettiğimiz Web Elementini kullanabilmek için bir variable'a atama yaparız

```
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 ögesini 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 işlevi yapan bir grup Web Elementi için kullanılır
- ❖ Class attribute'nun değeri boşluk içeriyorsa genelde By.classname( ) ile yapılan locator'lar sağlıklı 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 tanıyabilir ve sonra bu yöntemi 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 verdiğinizde 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 bulmasi için findElement( ) yöntemini kullanırız.
- ❖ Bu, tek bir web elementini döndürür. Ayni locator ile ulasilabilen birden fazla web element varsa ilkini dondiurur
- ❖ Driver elementi bulamazsa, runtime exception verir : **NoSuchElementException**.
- ❖ **NoSuchElementException**'ı gördüğünüzde, locatorı tekrar kontrol etmelisiniz.



# findElements( ) Method

```
List<WebElement> elementName=driver.findElement(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 uyusursa	Ilk elemani dondurur	Tum elemanlari ondurur
websayfasinda hicbir Web Element Locator ile uyusmazsa	NoSuchElementException firlatir	Exception firlatmaz, bos bir liste dondurur
Return Type	WebElement	List<WebElement>
Elemana erisim	Direk ulasilabilir	Liste'den index veya iterator ile ulasilabilir





# WebElement Method'ları

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'ları

## 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](mailto:testtechproed@gmail.com)
    - ii. Password : Test1234!
  - e. Expected user id nin [testtechproed@gmail.com](mailto: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'ları

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

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

`webElement.tagName()`; 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 kullanmıyoruz



# WebElement Is Method'leri

**webElement.isEnabled();** Web Element erişilebilir ise true  
yoksa false döner

**webElement.isDisplayed();** Web Element görünür ise true  
yoksa false döner

**webElement.isSelected();** Web Element seçili ise true  
yoksa false döner

**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 kullanmıyoruz



# WebElement Method'ları

## 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>
      </tr>
    </tbody>
  </table>
</div>
```

**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 çocuğudur(child) ve <tbody>, <tr> in ebeveyndir (parent)

<td> tagları ise siblings (kardes)'dir



# LOCATORS XPath

## 7 ) By.xpath( );

Bir WebElement'i locate etmek için kullanabileceğimiz en etkin method'dur.

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

Method'un yazımı açısından diğer 6 yöntem ile aynı olmakla beraber xpath'i diğerlerinden ayıran çok önemli bir fark vardır



Diğer 6 method HTML kod'una bağlıdır. Web element'in kodunda id yoksa By.id( ) method'unu, web element link değilse By.linkText( ) method'unu kullanamazsınız.

Xpath ise dinamiktir. Her türlü web element için mutlaka bir xpath yazılabilir

2 çeşit Xpath yazılabilir

1.Absolute xpath (mutlak)

2.Relative xpath (bağlı)



# 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>
          // tbody / tr / td[3] / a / span
        </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 basit // 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 web element için unique sonuç veren bir çok xpath yazılabilir

Relative xpath yazmak için bu 3 bileşen aşağıdaki gibi bir araya getirilir, unique sonuç 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 bileşen de kullanılır, Ancak bazen daha az bileşen yazmak da yeterli olabilir.

Sadece tag name kullanarak xpath yazmak için

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

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

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

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

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

Attribute value farketmeksizin 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/](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 bağlı olmadan sadece web element icinde bulunan text kullanılabilir.

**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 ulaşamazsak 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/](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 CcssSelector

## 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.classValue" veya sadece css=".classValue"=>yalnızca class value ile çalışır

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



# LOCATORS CcssSelector

Home Work: Log in Test Using Ccss

- 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](http://a.testaddressbook.com/sign_in)
  - b. Locate email textbox
  - c. Locate password textbox ve
  - d. Locate signin button
  - e. Aşağıdaki kullanıcı adını ve şifreyi girin ve sign in düğmesini tıklayın
    - i. Username : [testtechproed@gmail.com](mailto: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- Sayfayi “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-Sayfayi kapatın





# LOCATORS Relative Locators

## Relative Locators nedir ?

- ❖ Selenium 4 ile gelen yeniliklerden biri de bagli 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 **R**elative **L**ocators

## Class **W**ork: **R**elative **L**ocators

- 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 anlaşılır kılmak, tekrar eden kodlardan kaçınmak, updateleri tek merkezden kolayca yapabilmek ve projenin sürdürülebilirliğini 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.

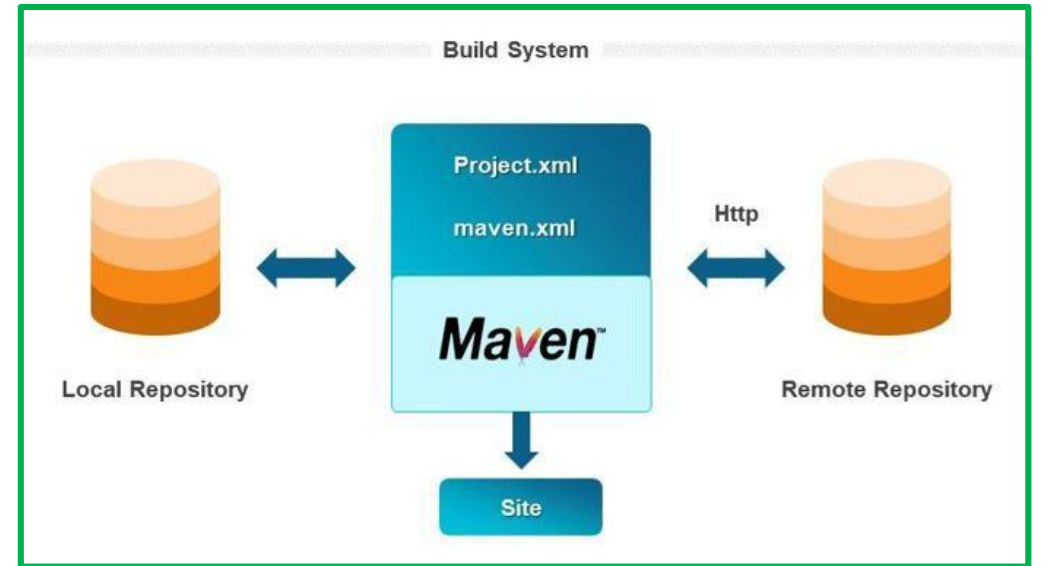
# Apache Maven™

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



# Maven™

- ❖ Maven bir Java derleme aracıdır (build tool). Maven proje otomasyon ve yönetim aracıdır (automation and management tool).
- ❖ Maven, konfigrasyon için pom.xml dosyasını kullanır. Bu dosya projenin inşası, 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™

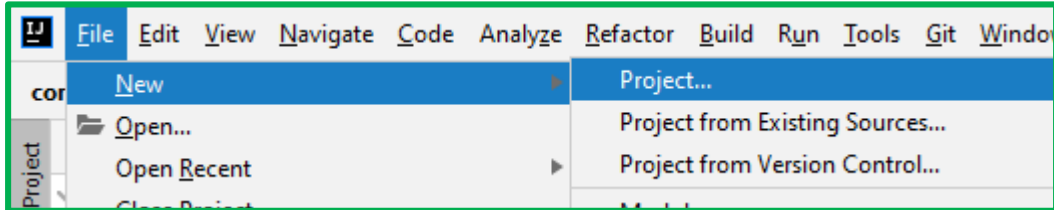
## Neden Maven?

- ❖ Tekrarlanabilir derlemeler / yürütmeleri 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 önemli 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?

Sürücüleri (drivers) kurmak, sürücüleri oluşturmak için dependencies gereklidir. Gerekli araçları içe aktarır(import).

### ✓ Pom dosyanızı ve dependencies'i nasıl güncellersiniz ?

Mvnrepository.com adresinden gerekli dosyaları bulur ve pom dosyamıza 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 kopyalayalım
- 4) pom XML dosyasina gidelim
- 5) <properties> satirinin altina <dependencies> </dependencies> tag'i olusturalım
- 6) Kopyaladigimiz kodlari <dependencies> bolumune yapistiralım
- 7) Ayni islemleri Selenium Java dependency icin de yapalım
- 8) Pom XML dosyasinin sag tarafinda **Maven** yazan bolumu tiklayalım
- 9) Yenile butonuna tiklayip asagida Dependencies bolumunun olustugunu ve altinda ekledigimiz kutuphanelerin gorundugunden emin olun



# Maven™

## POM XML DOSYASINA DEPENDENCIES EKLEME

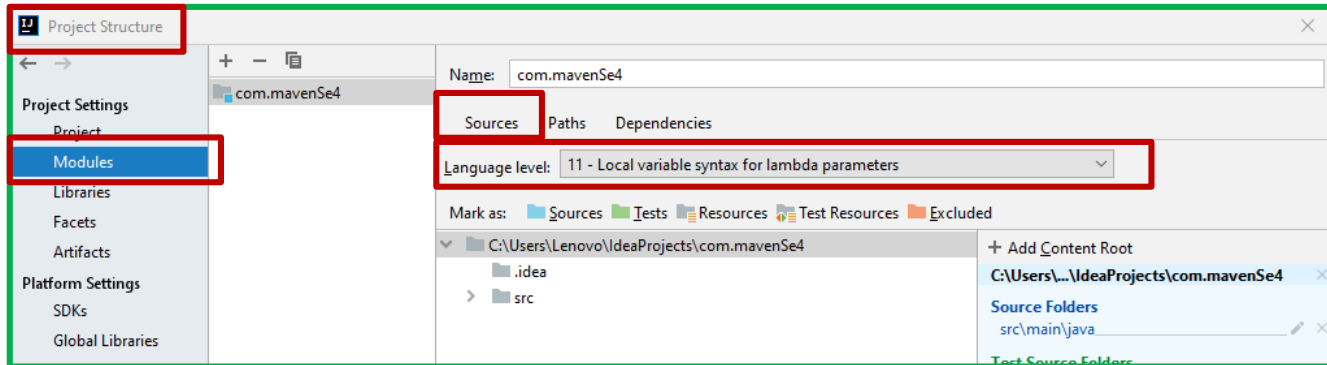
```
7 <groupId>org.example</groupId>
8 <artifactId>com.mavenSe4</artifactId>
9 <version>1.0-SNAPSHOT</version>
10
11 <properties>
12 <maven.compiler.source>11</maven.compiler.source>
13 <maven.compiler.target>11</maven.compiler.target>
14 </properties>
15
16 <dependencies>
17 <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
18 <dependency>
19 <groupId>io.github.bonigarcia</groupId>
20 <artifactId>webdrivermanager</artifactId>
21 <version>5.0.3</version>
22 </dependency>
23
24 <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
25 <dependency>
26 <groupId>org.seleniumhq.selenium</groupId>
27 <artifactId>selenium-java</artifactId>
28 <version>4.1.0</version>
29 </dependency>
30
31 </dependencies>
32
33
34 </project>
```

The screenshot shows an IDE with a Maven project. The main editor displays the `pom.xml` file for `com.mavenSe4`. The `<dependencies>` section is highlighted with a red box, showing two dependencies: `io.github.bonigarcia:webdrivermanager:5.0.3` and `org.seleniumhq.selenium:selenium-java:4.1.0`. The right sidebar shows the Maven project structure, with the `Dependencies` section expanded and highlighted with a red box, listing the same two dependencies.

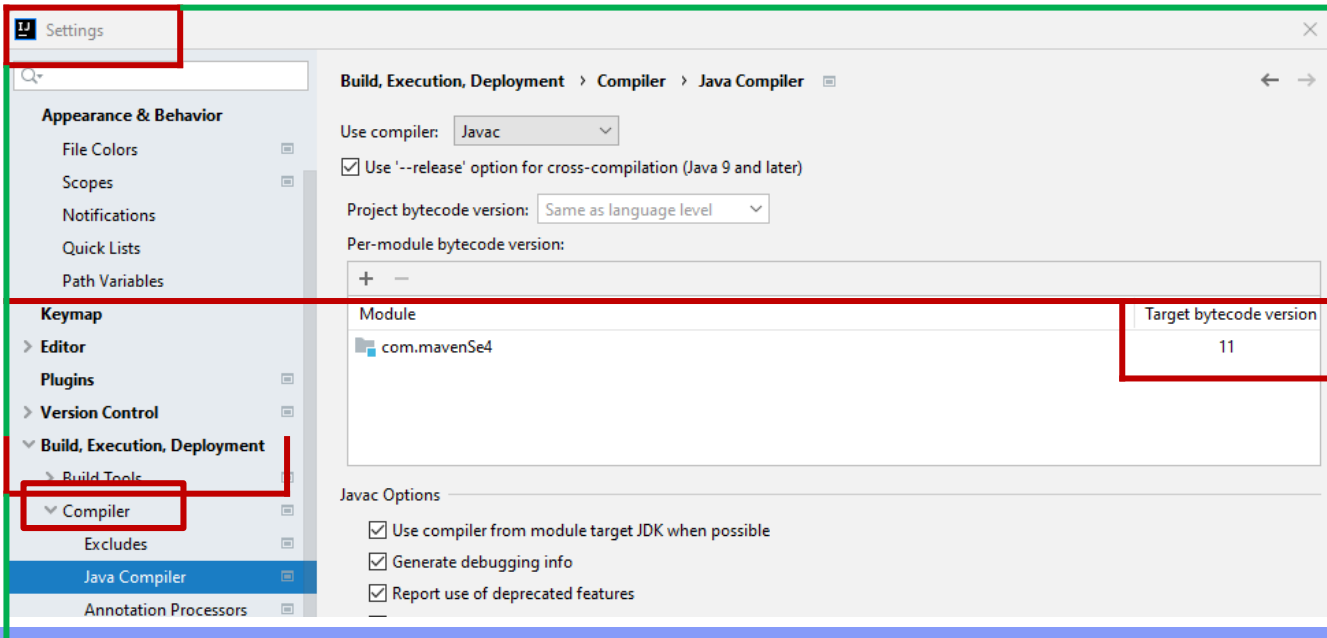


# Maven™

## CLASS AYARLARI



File  
Project Structure  
Modules  
Sources  
Language level : min **8** yapalim,  
bilgisayarınızda kurulu java 8 veya ustu ise  
**java versiyonu ayni** olmalı



File  
Settings  
Build,Execution,Deployment  
Compiler  
Java Compiler  
Target bytecode version : min **8** yapalim,  
bilgisayarınızda kurulu java 8 veya ustu ise  
**java versiyonu ayni** olmalı



# Maven™

## CLASS WebDriver AYARLARI

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

### Class Work Amazon Search Test

- 1 <https://www.amazon.com/> sayfasına gidelim
- 2 arama kutusunu locate edelim
- 3 "Samsung headphones" ile arama yapalım
- 4 Bulunan sonuc sayısını yazdıralım
- 5 İlk ürünü tıklayalım
- 6 Sayfadaki tüm başlıkları yazdıralım



## 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 yatirmek 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 Sayfayi 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. Sayfayi kapatın





# JUnit

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





## Annotations

- ❖ Selenium'da kodlarımızı yazarken “@” işareti 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 özellikler 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



## Annotations

### @Test ve @Ignore

- ❖ Junit ile Main Method kullanma donemini bitiriyoruz.
- ❖ Junit Framework kullandigimizda yazdigimiz test metodunun calismasi icin başına @Test notasyonu eklememiz yeterlidir.
- ❖ @Test notasyonu eklemedigimiz metot test sırasında çalıştırılmaz. Ancak cagrilirsa calisir.
- ❖ Yazdigimiz bazı test metotları henüz tamamlanmamış veya değişiklikleri uğrayabileceğinden dolayı test sırasında çalıştırılmasını istemiyorsak @Ignore notasyonu eklememiz yeterlidir.
- ❖ @Ignore notasyonun tanımlı olduğu metotlar test sırasında çalıştırılmayacaktır. Ayrıca istenilirse @Ignore("açıklama") şeklinde yazılarak metodun neden test edilmesini istemediğimizide 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 notasyonunun kullanıldığı method'a genelde setup() ismi verilir.

- ❖ After notasyonu, her test method'undan sonra çalışmaktadır. Örneğin test sırasında kullandığımız sayfanın kapatılması gibi.

@after notasyonunun kullanıldığı metoda genelde teardown() ismi verilir

**Not :** Test method'u ve Test farklı yapılardır.

**Test** dedigimizde tek bir method veya içinde birçok class ve package barındıran bir yapı olabilir. Regression test, smoke test vb..

**Test method'u** ise @Test notasyonu kullanılarak oluşturulan ve bağımsız olarak çalıştırabileceğimiz en küçük test yapısıdır



## Annotations

### @BeforeClass ve @AfterClass

- ❖ BeforeClass notasyonu, bir class'daki tüm testlerden önce yapılması gereken bir işlem varsa kullanılır (precondition).

Örneğin test metodlarımız çalışmadan driver oluşturup tüm methodlarda kullanabilirim.

- ❖ AfterClass notasyonu da, bir class'daki tüm testler tamamlandıktan sonra yapılması gereken işlemlerde kullanılır.

Örneğin açtığımız sayfayı kapatmak veya elde ettiğimiz test sonuçlarını 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 karsilastirilmesi

### 1) Tum Class calistirilirs

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

eger bir class ta Before ve After methodlar varsa bir method icin üç method calisir

### 2) Tek bir test method'u calistirilirs

```
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
```

```
1 public class C03 {
2     @BeforeClass
3     public static void beforeClassM(){
4         System.out.println("Before class method'u calisti");
5     }
6     @Before
7     public void beforeM() {
8         System.out.println("Before method'u calisti");
9     }
10    @Test
11    public void test1(){
12        System.out.println("Test 1 calisti");
13    }
14    @Test
15    public void test2(){
16        System.out.println("Test 2 calisti");
17    }
18    @Test
19    public void test3(){
20        System.out.println("Test 3 calisti");
21    }
22    @AfterClass
23    public static void afterClassM(){
24        System.out.println("After class method'u calisti");
25    }
26    @After
27    public void afterM() {
28        System.out.println("After method'u calisti");
29    }
30 }
```



## CheckBox

1. Bir class oluşturun : CheckBoxTest
2. Gerekli yapıyı oluşturun 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



## Radio Button

1. Bir class oluşturun : RadioButtonTest
2. Gerekli yapıyı oluşturun ve aşağıdaki görevi tamamlayın.
  - <https://www.facebook.com> adresine gidin
  - Cookies'i kabul edin
  - "Create an Account" button'una basın
  - "radio buttons" elementlerini locate edin
  - Secili değilse cinsiyet butonundan size uygun olanı secin

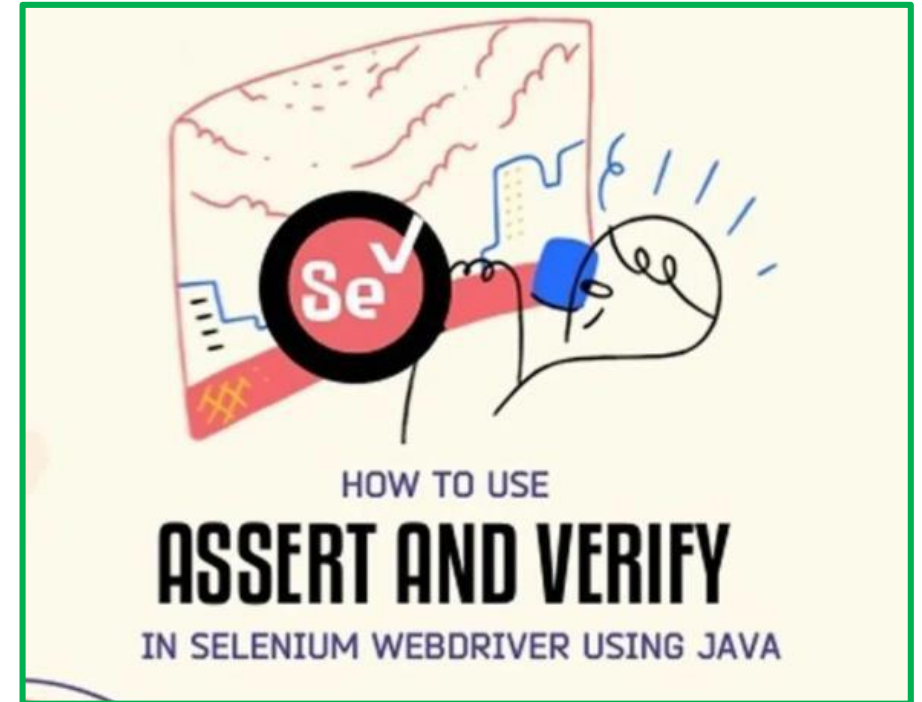




# JUnit

## Assertions: (Verification) Doğrulama

- ❖ Test için uygun bir şekilde hazırlanmış bir framework'da expected değerlerin actual değerlere eşit olup olmadığına if-else statement ile bakılmaz.
- ❖ JUnit framework'ünde test senaryomuzu çalıştırdıktan 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'ları kullanırız.
- ❖ Bu method'lar ile expected ve actual değerleri karşılaştırarak testimizi sonuçlandırırız.





## Assertions (Verifications)

- ❖ Assert method'unu seçmeden önce kıyaslamada beklediğimiz sonucun true veya false olmasına karar vermeliyiz.

```
String expectedName= "Ali Can";  
String actualName= "ALI CAN";  
  
expectedName.equals(actualName); → false  
expectedName.equalsIgnoreCase(actualName); → true
```

```
// Emeklilik yasi 65  
  
int yas1=63;  
int yas2=68;  
  
emekliOlabilirMi(yas1); → false  
emekliOlabilirMi(yas2); → true
```

- ❖ ÖNEMLİ OLAN kıyaslama sonucunun true veya false olması değil, beklediğimiz sonucun olup olmamasıdır. Assertion başarısız olursa AssertionError ile hata mesajı verilir.

Assert.**assertEqual**(actualName,expectedName) → FAILED  
Burasi bekledigimiz sonuc false

Assert.**assertTrue**(yas2>65) → PASSED  
Beklenen true true

Assert.**assertTrue**(yas2<65) → FAILED  
false

Assert.**assertFalse**(yas1>65) → PASSED  
Beklenen false false sonuc

Assert.**assertFalse**(yas1<65) → FAILED  
true



## Assertions

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

expected = actual ---> test PASSED  
expected  $\neq$  actual ---> test FAILED

Assert.**assertEqual**(A,B)  $\longrightarrow$  FAILED  
false

Assert.**assertEqual**(A,D)  $\longrightarrow$  PASSED  
true

Assert.**assertTrue**(C>D)  $\longrightarrow$  PASSED  
true

Assert.**assertTrue**(C>B)  $\longrightarrow$  PASSED  
true

Assert.**assertFalse**(B>D)  $\longrightarrow$  FAILED  
beklenen false  
true  
gelen true

Assert.**assertFalse**(B>A)  $\longrightarrow$  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 görüntülendiğini (isDisplayed()) test edin
  - Search Box 'in erişilebilir olduğunu test edin (isEnabled())
  - wrongTitleTest => Sayfa başlığının “youtube” olmadığını doğrulayın



## 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 ÖZET

Assert, bir test senaryosunun PASS veya FAILED durumunu belirlemede kullanılan yararlı bir yöntemdir. Secilen metot ve yazılan **boolean koşul**'a gore 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 esit ise test PASS, yoksa test FAILED olur

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

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



## Handle Dropdown Acilir Pencere

- **Adim1:** Dropdown menuyu herhangi bir locator ile locate edin.

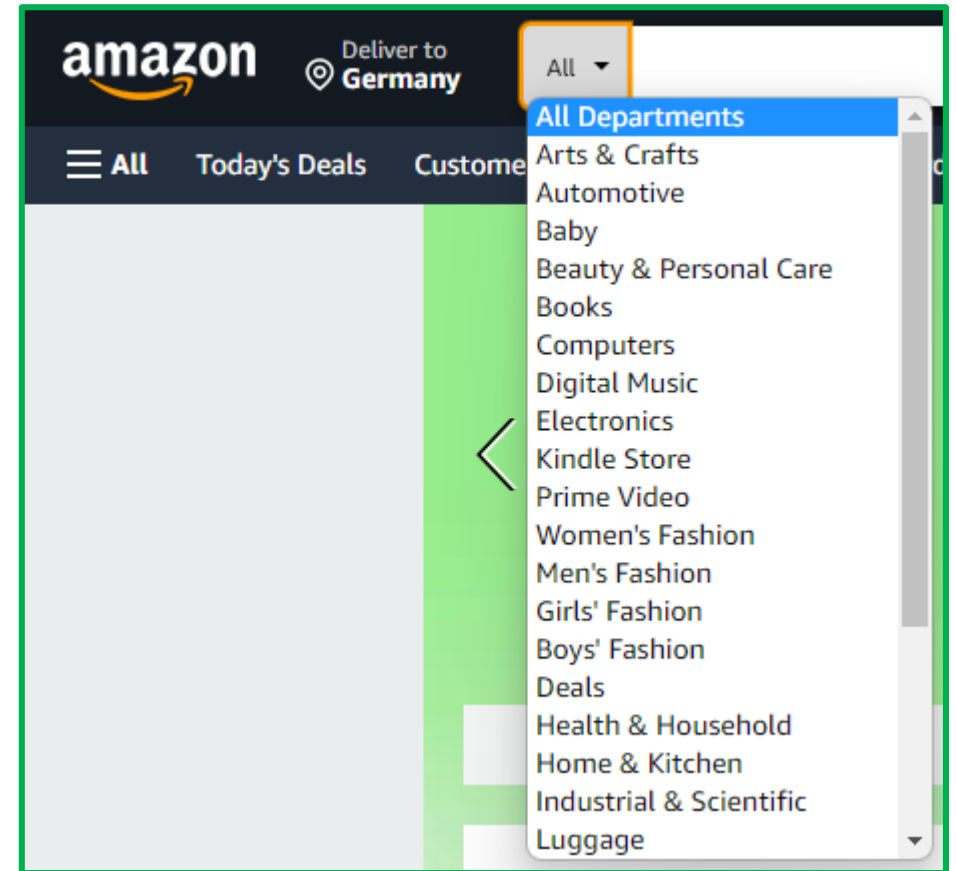
WebElement `ddm=driver.findElement(By.id("value of id"));`

- **Adim 2:** Yeni bir **"Select"** objesi olusturun ve daha once locate ettigimiz WebElement'i **parametre** olarak yeni objeye ekleyin

Select `options=new Select(ddm);`

- **Adim 3:** Select class'ından kullanabileceginiz 3 yontemden biriyle dropdown menusundeki elemanlardan istediginizi secin

`options.selectByIndex(1);`





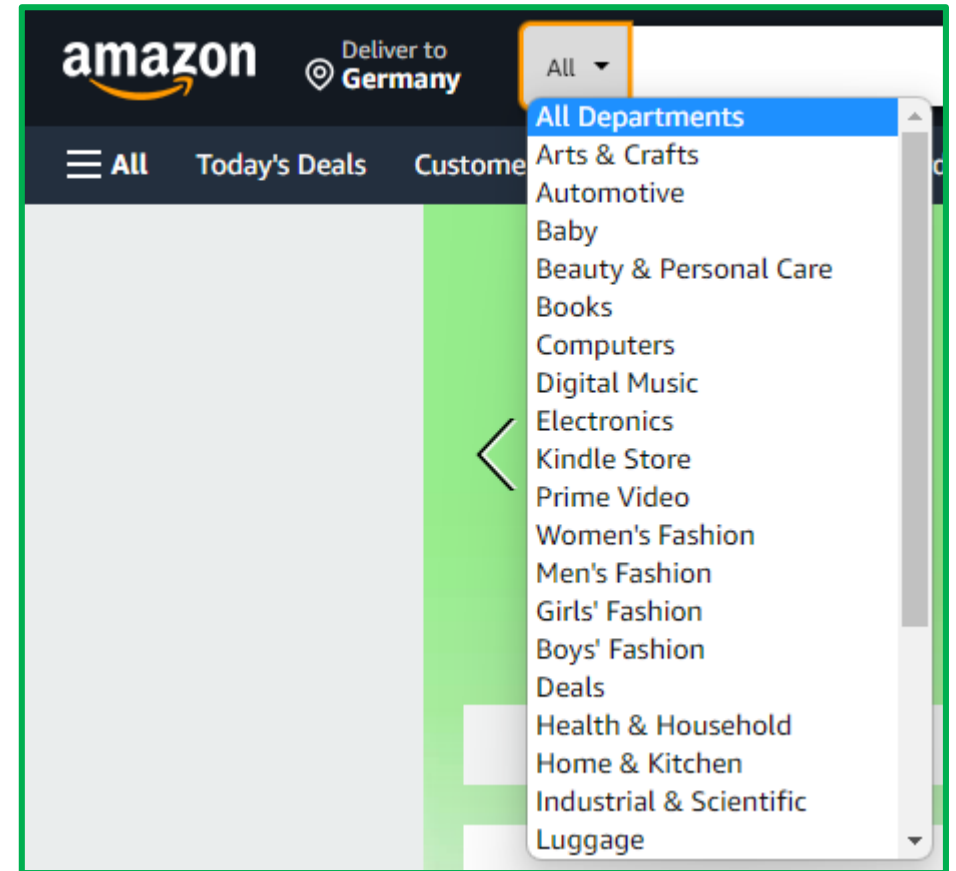


## Handle Dropdown

Dropdown menüsündeki elementleri Select Class'ından kullanacağımız yöntemlerle 3 şekilde seçebiliriz

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

Istenirse **`getOptions();`** methodu kullanılarak 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 , değilse False yazdırın.



## Handle Dropdown

- Bir class oluşturun: C3\_DropDownAmazon
- <https://www.amazon.com/> adresine gidin.

### - Test 1

Arama kutusunun yanındaki kategori menüsündeki kategori sayısının 45 olduğunu test edin

### -Test 2

1. Kategori menüsünden Books secenegini secin
2. Arama kutusuna Java yazın ve aratın
3. Bulunan sonuç sayısını yazdırın
4. Sonucun Java kelimesini içerdigini test edin



## Handle Dropdown

1. <http://zero.webappsecurity.com/> Adresine gidin
2. Sign in butonuna basin
3. Login kutusuna "username" yazin
4. Password kutusuna "password." yazin
5. Sign in tusuna basin
6. Pay Bills sayfasina gidin
7. "Purchase Foreign Currency" tusuna basin
8. "Currency" drop down menusunden Eurozone'u secin
9. "amount" kutusuna bir sayi girin
10. "US Dollars" in secilmedigini test edin
11. "Selected currency" butonunu secin
12. "Calculate Costs" butonuna basin sonra "purchase" butonuna basin
13. "Foreign currency cash was successfully purchased." yazisinin ciktigini kontrol edin.



## Handle Alerts

### Alert Nedir?

Alert kullanıcıya bir tür bilgi vermek veya belirli bir işlemi gerçekleştirme izni istemek için ekran bildirimi 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 ihtiyac 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.

[https://the-internet.herokuapp.com/javascript\\_alerts](https://the-internet.herokuapp.com/javascript_alerts)



## Handle Allert 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 Allert Class Work

- Bir class oluşturun: **Alerts**
- [https://the-internet.herokuapp.com/javascript\\_alerts](https://the-internet.herokuapp.com/javascript_alerts) adresine gidin.
- Bir metod oluşturun: **acceptAlert**
  - 1. butona tıklayın, uyarıdaki OK butonuna tıklayın ve result mesajının “You successfully clicked an alert” olduğunu test edin.
- Bir metod oluşturun: **dismissAlert**
  - 2. butona tıklayın, uyarıdaki Cancel butonuna tıklayın ve result mesajının “successfully” içermediğini test edin.
- Bir metod oluşturun: **sendKeysAlert**
  - 3. butona tıklayın, uyarıdaki metin kutusuna isminizi yazın, OK butonuna tıklayın ve result mesajında isminizin görüntülendiğini 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'dir.

End user'lar için tasarlanmayan uygulamalarda(Örneğin API sorgularında) bu authentication HTML komutları ile de yapılabilir.

Bu authentication'i yapabilmek için uygulamanın kullanıcılara authentication'i nasıl yapacağına dair bilgilendirme yapmış olması gerekir.

Örneğin yandaki uygulama için authentication aşağıdaki gibi yapılabilir.

`https://username:password@URL`

Oturum açın

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

Kullanıcı adı

Şifre

Oturum açın

İptal





## Basic Authentication Class Work

the-internet.herokuapp.com/basic\_auth

- 1 Bir class olusturun : BasicAuthentication
- 2 [https://the-internet.herokuapp.com/basic\\_auth](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

Oturum açın

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

Kullanıcı adı

Şifre

Oturum açın

İptal

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

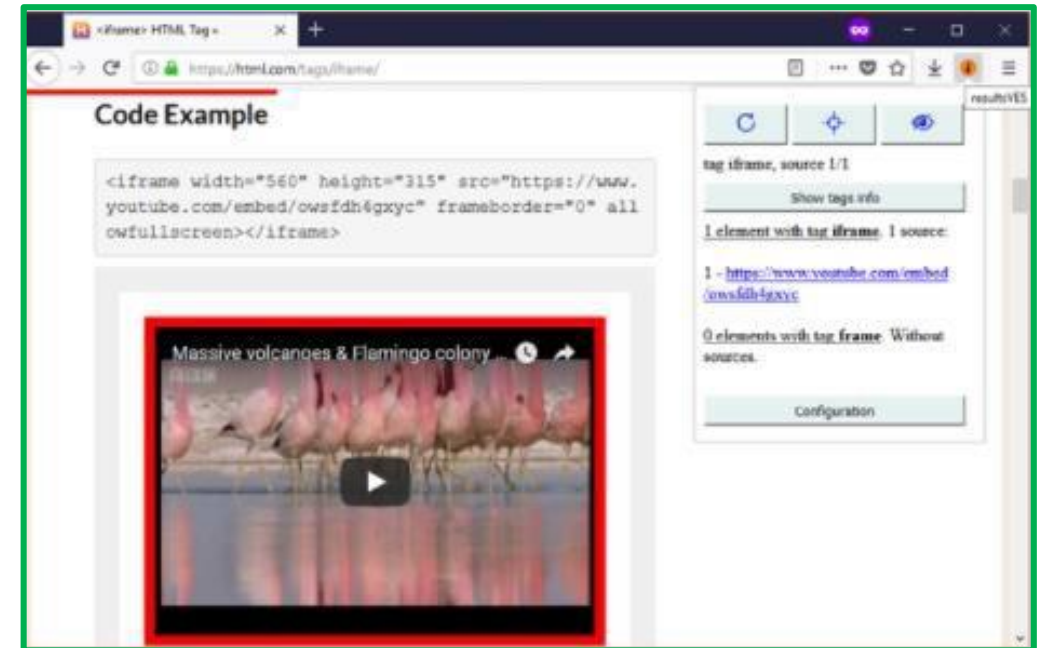


## Handle Iframe

### Iframe nedir?

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

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





## Handle Iframe

- Bir sayfada iframe varsa, Selenium bir iframe içindeki elementleri doğrudan göremez
- switchTo() metodu 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);
```



## Handle Iframe

Iframe'den cikmak icin 2 komut vardır

`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 çıkıp ana sayfaya dönünüz
  - 5) ikinci iframe'deki (Jmeter Made Easy) linke (<https://www.guru99.com/live-selenium-project.html>) tıklayınız