



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.

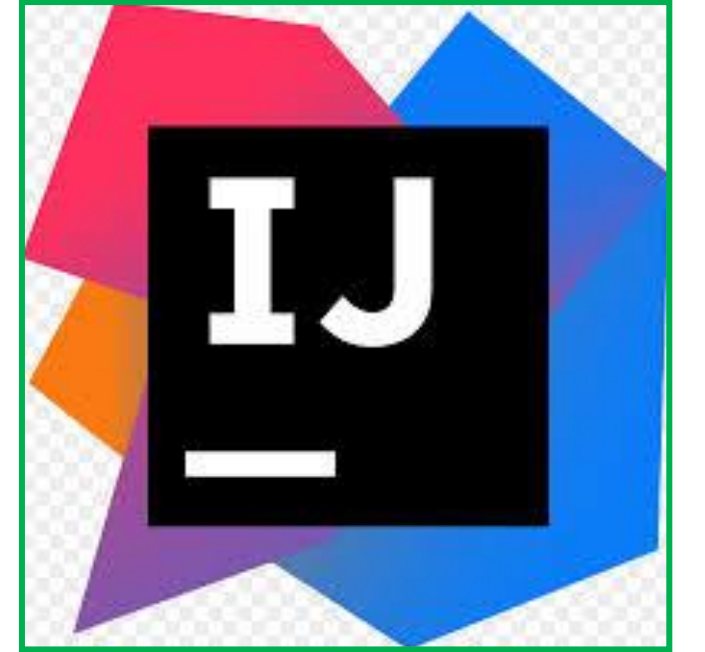
- Selenium, otomasyon yapmak için kendi sınıflarına ve yöntemlerine sahip bir suite'dir..

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



IntelliJ IDEA 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.



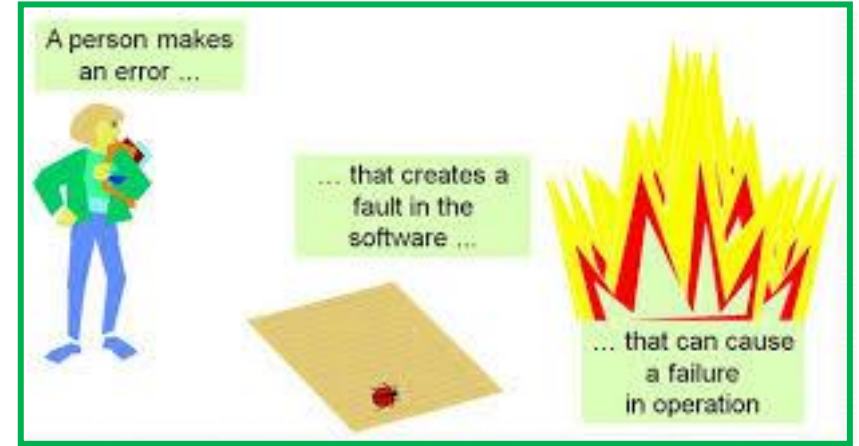


SOFTWARE TESTING NEDEN ÖNEMLİDİR ?

- 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
```

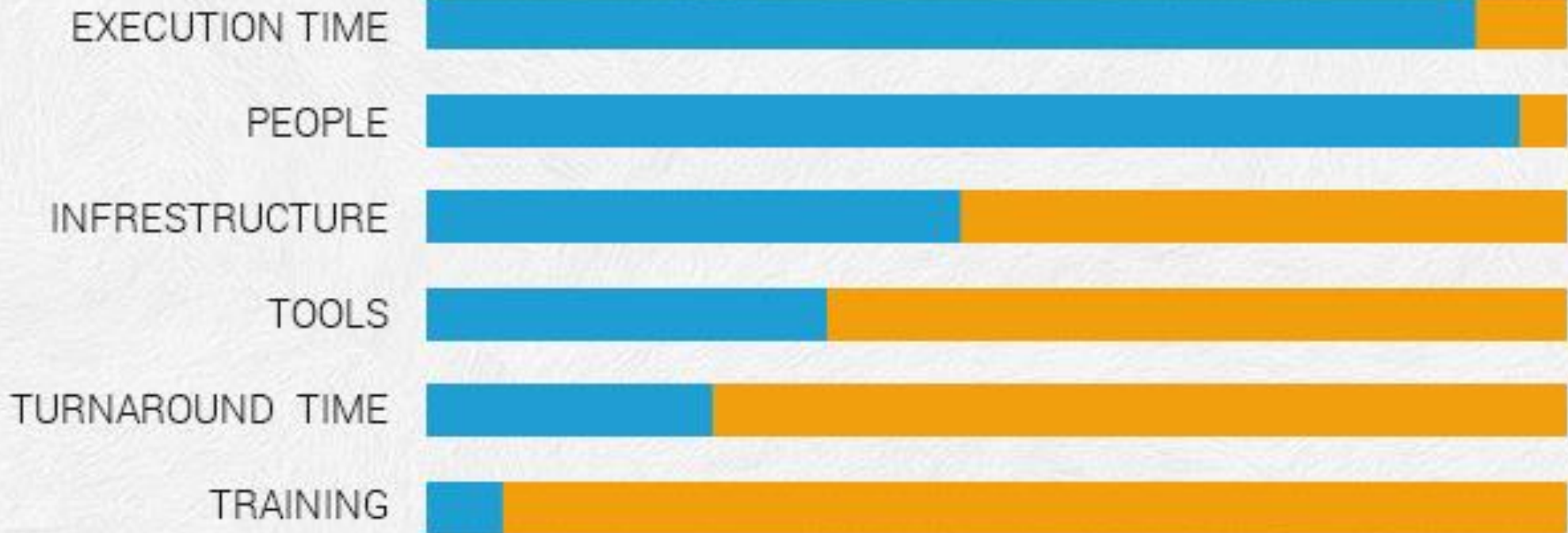



MANUAL TESTING vs AUTOMATION TESTING

MANUAL TESTING

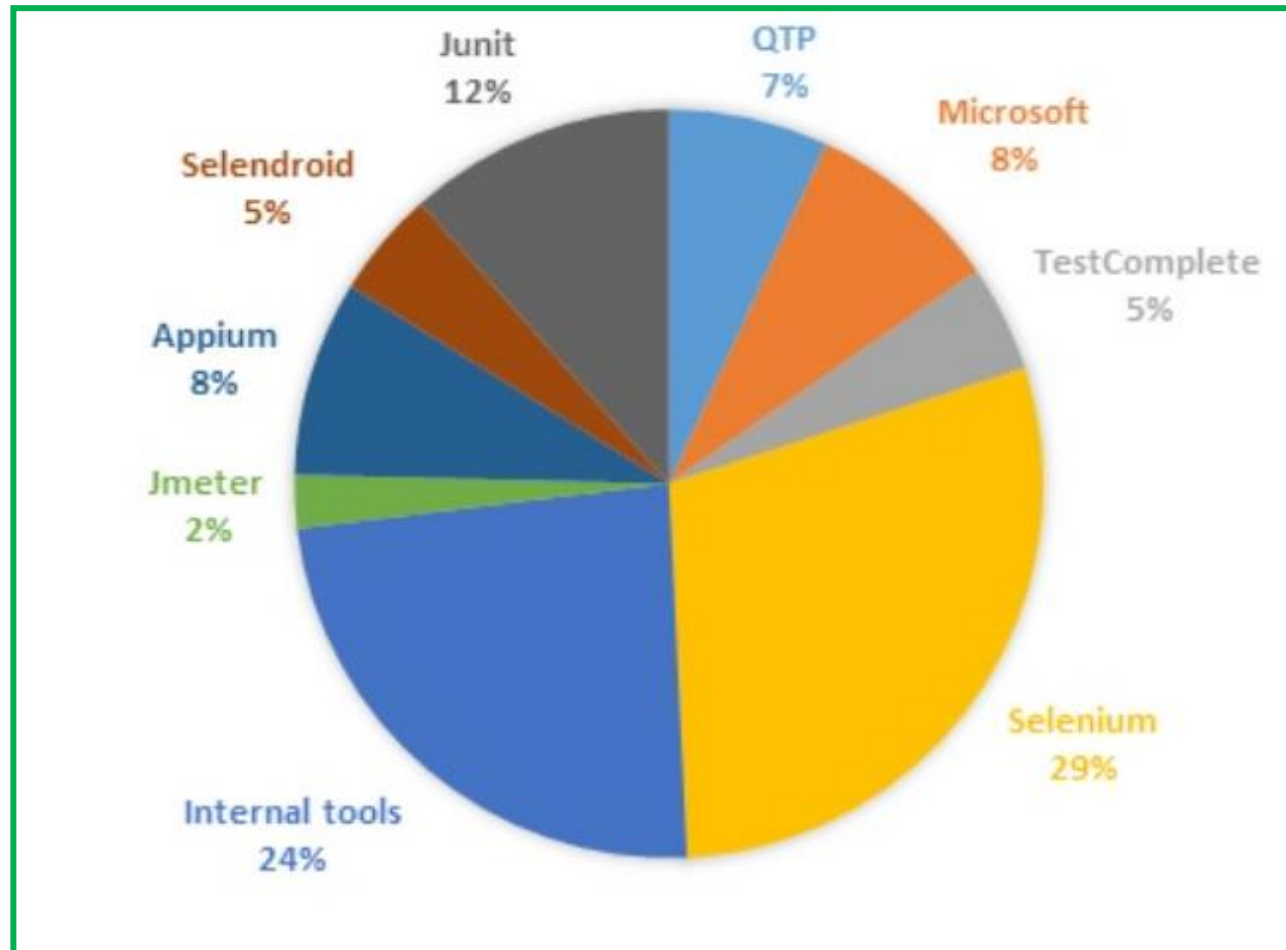
VS

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

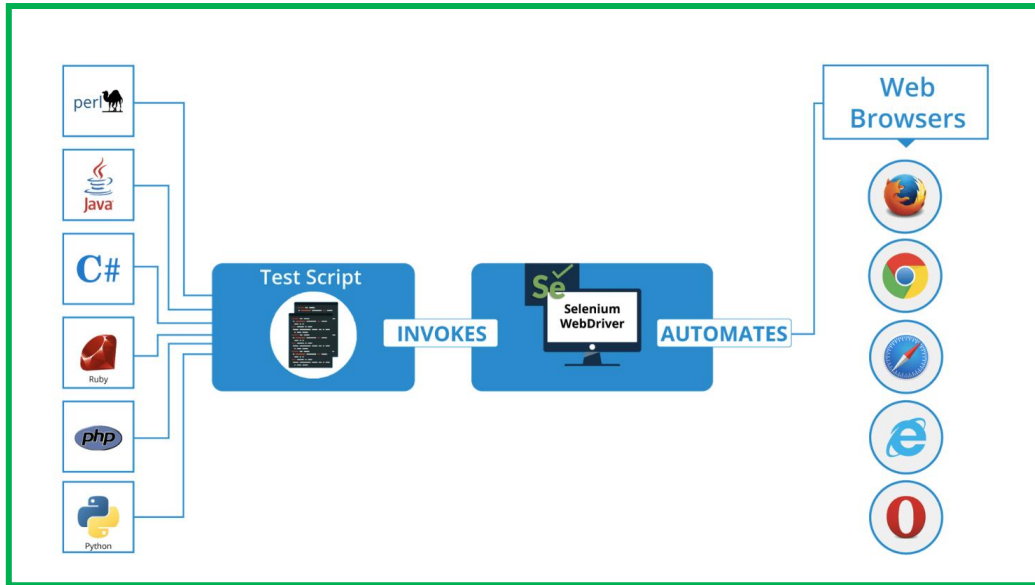
Selenium'un dört bileşeni vardır;



- 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?



- 1) Driver selenium'dan komutları alır.
- 2) Bu komutları tarayıcının API'ında dönüştürür
- 3) Browser'lardan donen sonuclari alır ve Selenium'a geri gönderir



SELENIUM'UN AVANTAJLARI NELERDİR?

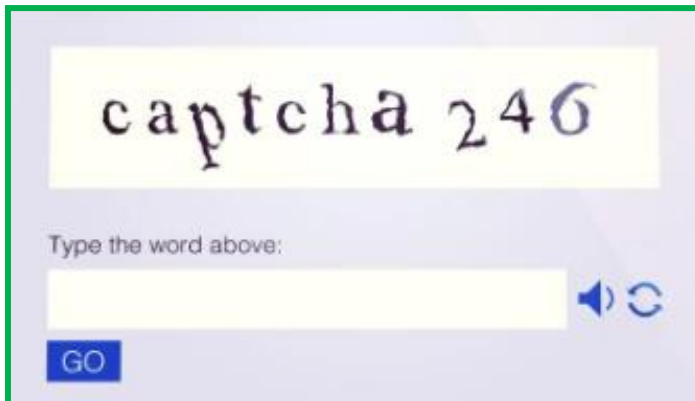


- 1) Ücretsiz ve açık kaynaklıdır.
(Open source)
- 2) Bir çok programlama dilini destekler
(Java, Python, PHP, C#, Ruby vs.)
- 3) Çoklu işletim sistemleriyle çalışır.
Multiple operating systems
(Windows, MacOS, Linux)
- 4) Birden çok tarayıcı ile çalışır.
Multiple browsers
(IE, Safari, Chrome, Firefox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

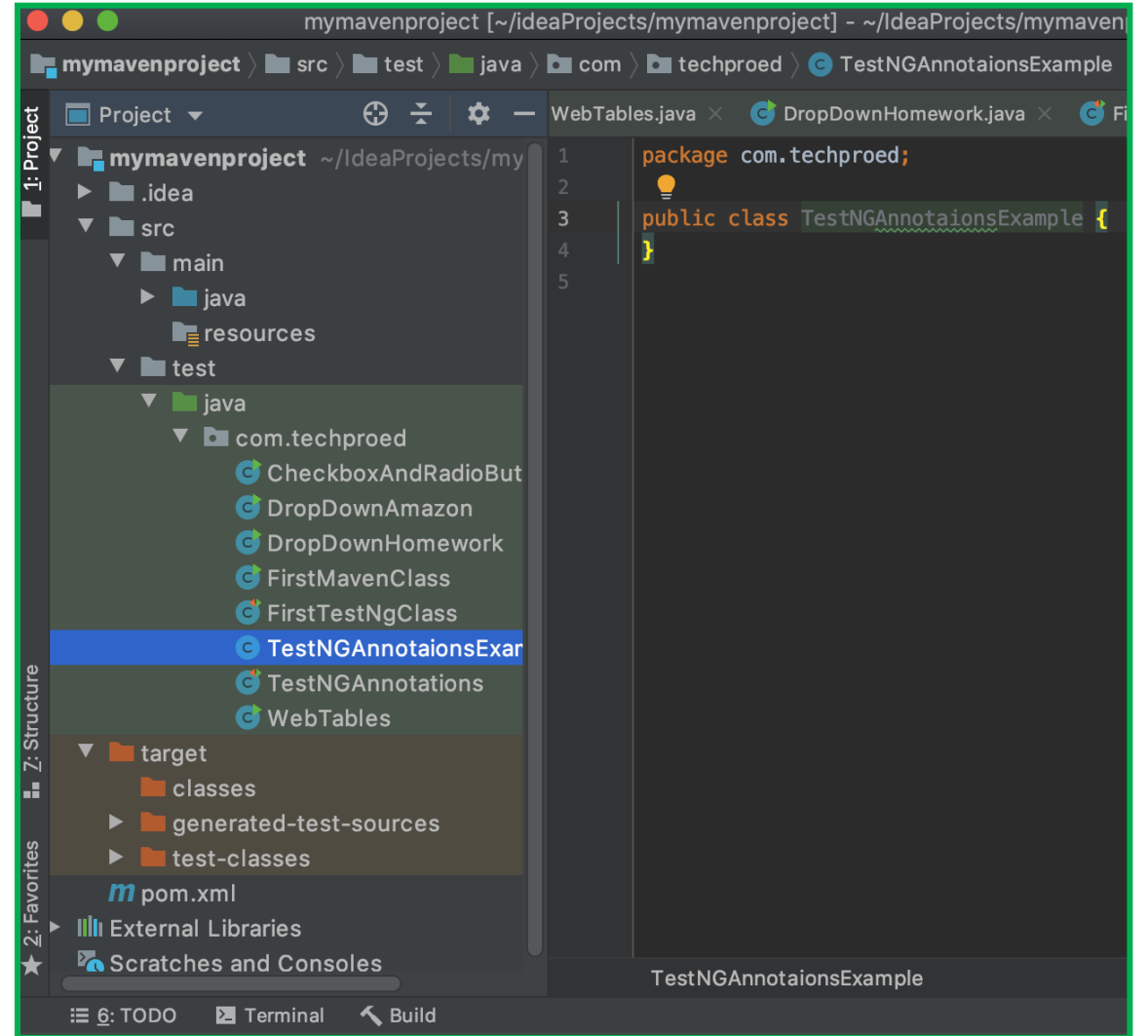


TEST AUTOMATION FRAMEWORK NEDİR?

- Test Otomasyon Framework (TAF), yazılım projelerindeki Test Case'lerin olusturulma 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 tiklayalım
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));
```

Istene 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



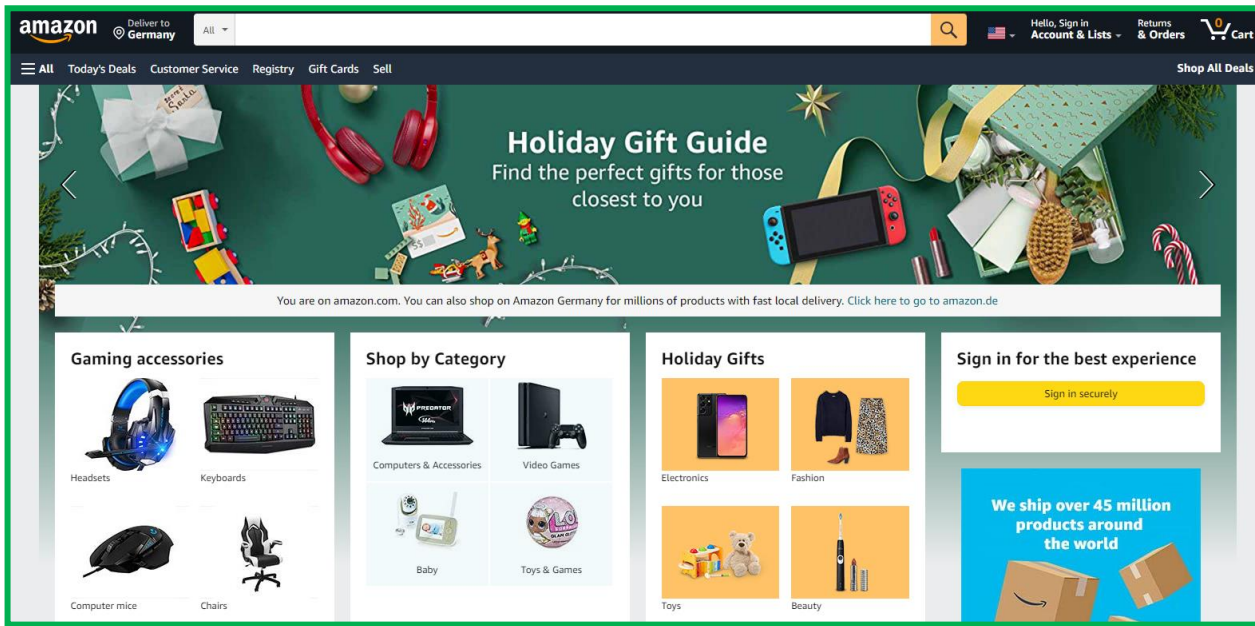
Tekrar Testi

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)

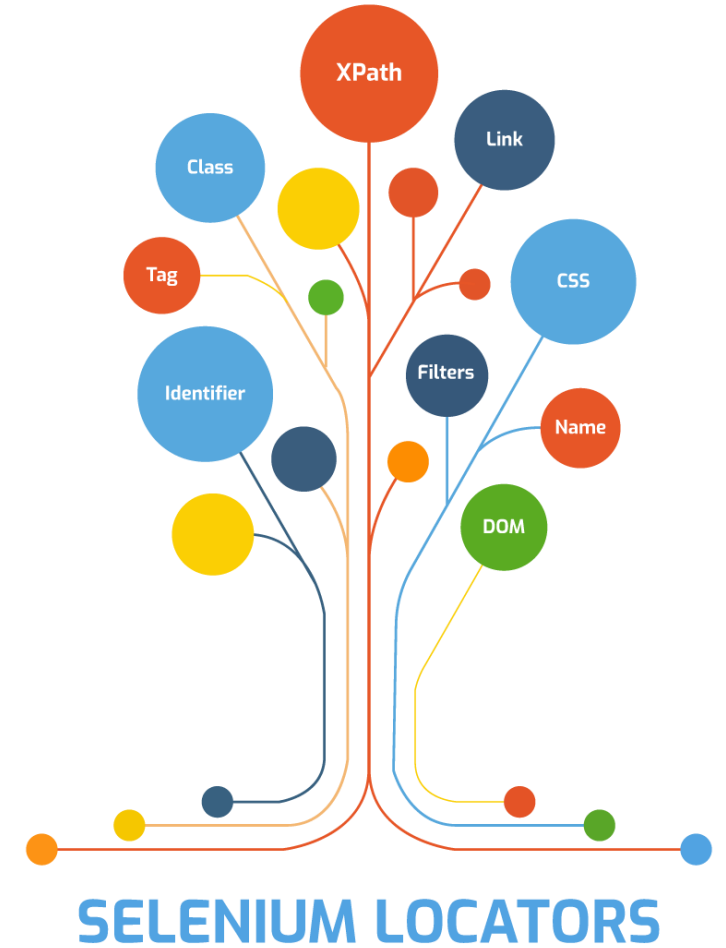
The screenshot shows the Amazon website with the browser's developer tools open. The 'Inspect' option in the right-click context menu is highlighted with a red circle. The Elements panel shows the HTML structure of the search bar.

```
<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>
    </form>
  </div>
</div>
```



LOCATORS (YER BULUCU-KONUM BELİRLEYİCİ)

- ❖ Selenium LOCATORS, web sayfasındaki web öğelerini tanımlamak için kullanılır.
- ❖ Selenium'da; metin kutuları, onay kutuları, linkler, radyo butonları, liste kutuları ve **diğer tum 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'ları kullanılır.





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 icine parameter olarak yazacagimiz Locator'in 8 locator'dan hangisi oldugunu belirtmek için de By.LocatorTuru("LocateBilgisi"); kullanilir

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

- 3) Locate ettigimiz 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 gecirilmedir



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. Aynı locator ile ulaşılabilen 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
 - 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");` 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> taglari 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" cellpadding="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'ın 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[@attributeismi='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/ 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/ 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 **C**ss**S**elector

Home **W**ork: **L**og **i**n **T**est **U**sing **C**ss

- 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-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 **R**elative **L**ocators

Relative **L**ocators 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 Area'nin 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

