

GUIDERSOFT

SELENIUM 04

14.07.2022

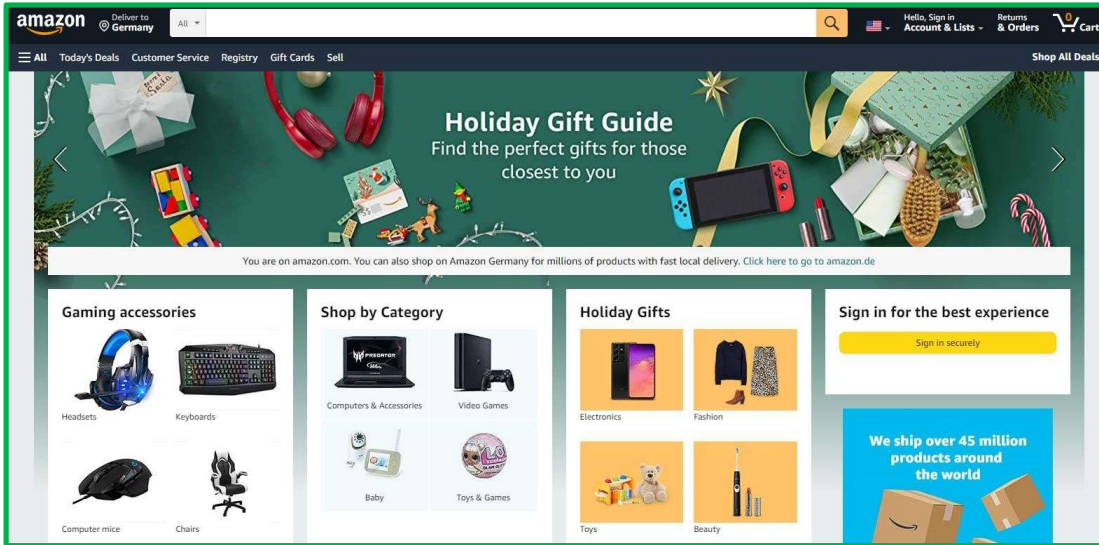
**WEBELEMENTS
LOCATORS
(YER BULUCU-KONUM BELIRLEYICI)**

TEKRAR TESTİ

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 turde WebElement tag'ları(etiketleri) vardır.

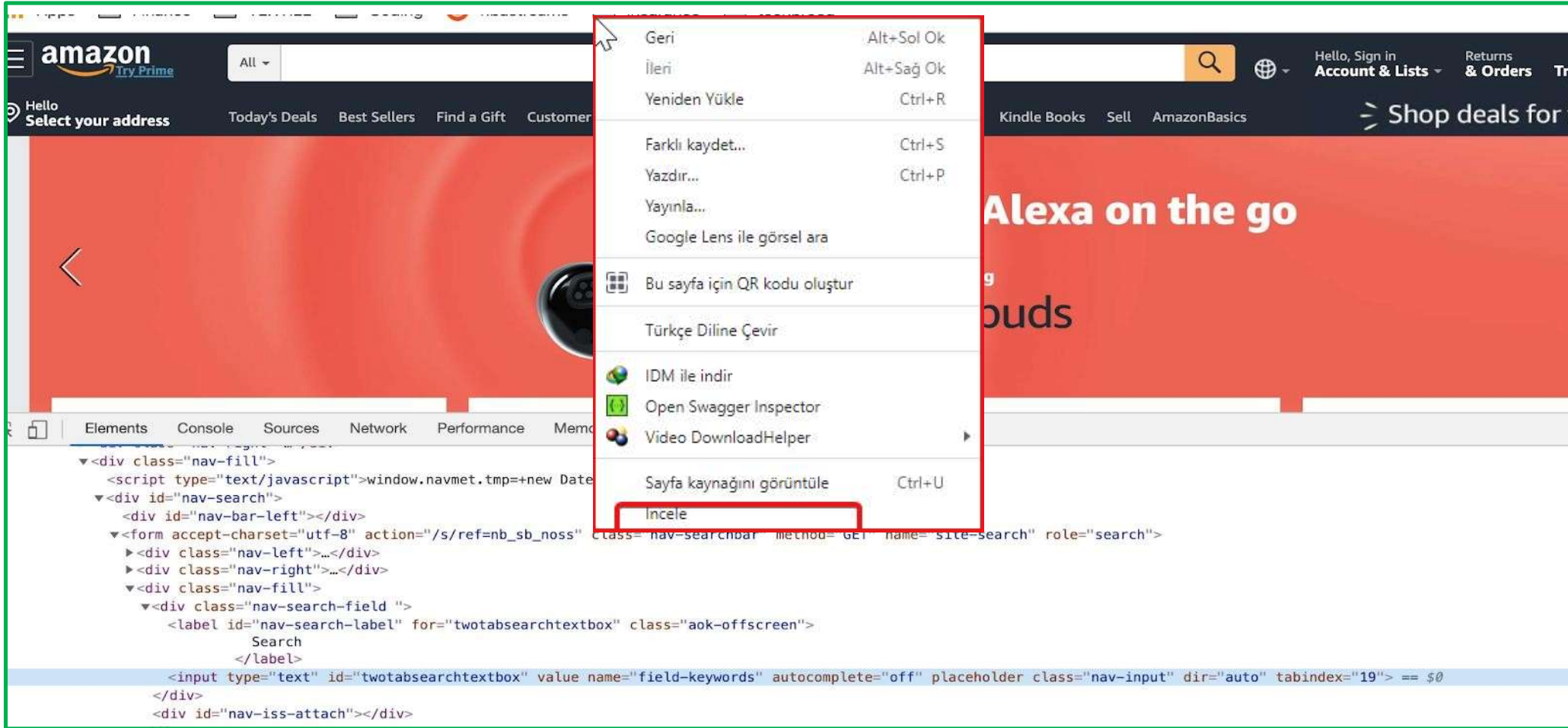
<html>,<body>,<form>,<label>,<input>,<a> vb.

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

WEBELEMENTS

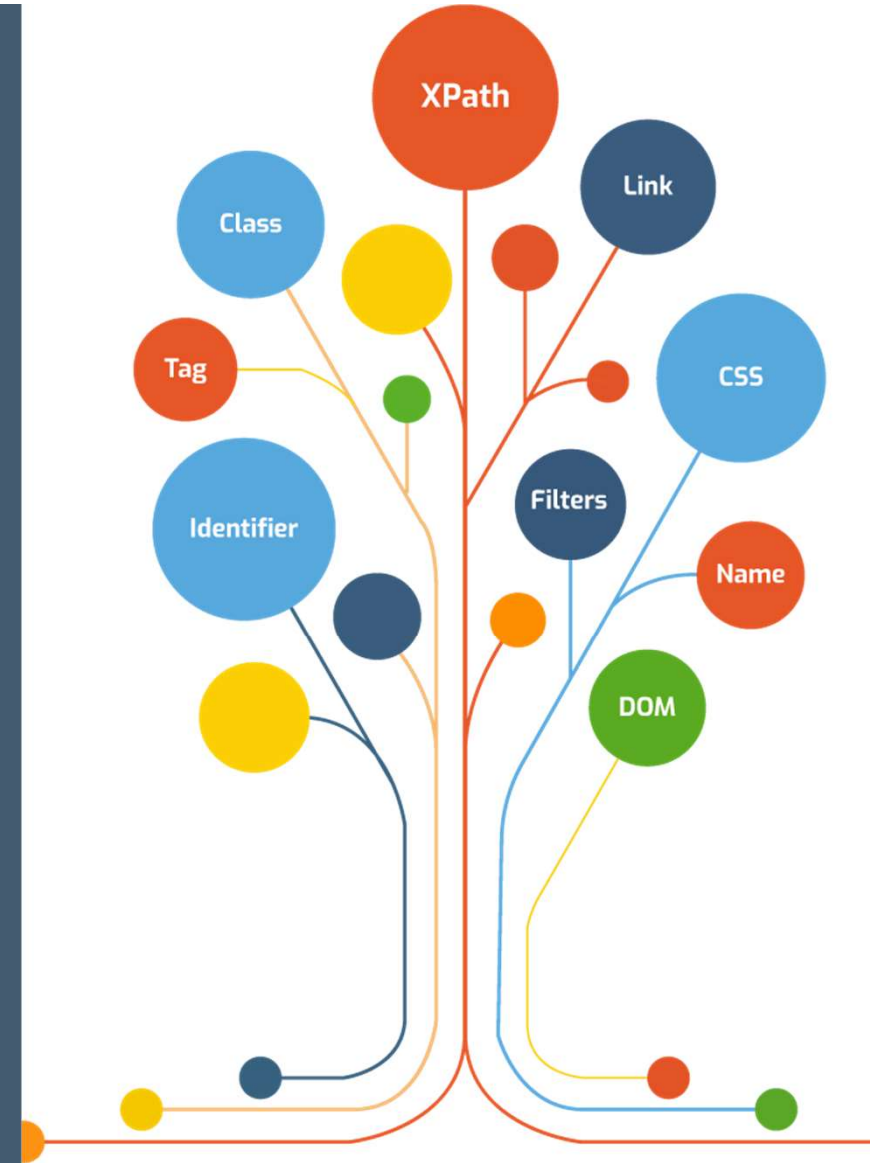
```
▼<tbody>
  ▶<tr>...</tr>
  ▼<tr> == $0
    ▼<td>
      <input type="email" class="inputtext login_form_input_box" name="email" id="email" data-testid="royal_email">
    </td>
    ▼<td>
      <input type="password" class="inputtext login_form_input_box" name="pass" id="pass" data-testid="royal_pass">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

WEB SAYFALARINI INCELEME (INSPECT)



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 tüm web öğeler üzerinde eylemler gerçekleştirmek için LOCATORS'a ihtiyacımız vardır.
- ❖ Konum belirleyiciler bize nesneleri tanımlamada yardımcı olur.
- ❖ Web Elementlerine ulaşmak için tag veya bazı attribute'ler kullanılır, bunlarla ulaşamayan webelementleri için özel olarak tanımlanan Xpath ve css locator'lari kullanılır.



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

LOCATOR'LARA GÖRE ELEMENTLERİ BULMA: 8

LOCATOR VARDIR

Locator'lara Göre Elementleri Bulma: 8 locator vardır

1. By.id => driver.findElement(**By.id**(""));
2. By.name => driver.findElement(**By.name**(""));
3. By.className => driver.findElement(**By.className**("")); 
4. By.tagName => driver.findElement(**By.tagName**(""));
5. By.linkText => driver.findElement(**By.linkText**(""));
6. By.partialLinkText => driver.findElement(**By.partialLinkText**(""));
7. By.xpath => driver.findElement(**By.xpath**(""));
8. By.css => driver.findElement(**By.css**(""));

LOCATORS BULMAK İÇİN 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** gördüğümüzde hata veren satırdaki locator gözden geçirilmelidir

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 arayuzundeki(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 bulması için `findElement()` yöntemini kullanırız.
- ❖ Bu, tek bir web elementini döndürür. Aynı locator ile ulaşılabilen birden fazla web element varsa ilkinin döndürülür.
- ❖ Driver elementi bulamazsa, runtime exception verir : **NoSuchElementException**.
- ❖ **NoSuchElementException**'i 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() arasındaki farklar

	findElement()	findElements()
websayfasında birden fazla Web Element Locator ile uyusursa	İlk elemanı döndürür	Tüm elemanları döndürür
websayfasında hiçbir Web Element Locator ile uyusmazsa	NoSuchElementException fırlatır	Exception fırlatmaz, boş bir liste döndürür
Return Type	WebElement	List<WebElement>
Elemana erişim	Direk ulaşılabilir	Liste'den index veya iterator ile ulaşılabilir

WEBELEMENT METHOD'LARI

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

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

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

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

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

WEBELEMENT METHOD'LARI Class Work: Login Test

1. Bir class oluşturun: LoginTest
2. Main method oluşturun ve aşağıdaki görevi tamamlayın.
 - a. <http://a.testaddressbook.com> adresine gidiniz.
 - b. Sign in butonuna basin
 - 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 oldugunu dogrulayin(verify).
 - f. “Addresses” ve “Sign Out” textlerinin görüntulendiğini(displayed) doğrulayin(verify).
3. Sayfada kac tane link oldugunu 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'lari

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

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

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

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