

Selenium Interview auf Englisch, Deutsch und Türkisch

Q1: What is **Automation Testing**?

A1: It is the process of testing a software or application using an automation testing tool to find the defects.

Q2: What are the **benefits** of Automation Testing?

A2:

- 1 Saves time and money.
- 2 Reusability of code.
- 3 Easy reporting.
- 4 Easy for compatibility testing.

Q1: Was ist Automatisierungstest (**Automationstest**)?

A1: Dies ist der Prozess des Testens einer Software oder Anwendung unter Verwendung eines Automatisierungstest-Tools, um die Fehler zu finden.

F2: Was sind die **Vorteile von Automatisierungstests (Automationstest)**?

A2:

1 Zeit und Geld sparen.

2 Wiederverwendbarkeit von Code.

3 Einfache Berichterstattung.

4 Einfach für Kompatibilitätstests.

S1: Automation Testing nedir?

C1: Defectleri bulmak için bir otomasyon test aracı kullanarak bir yazılımı veya uygulamayı test etme sürecidir.

S2: Automation Testin faydaları nelerdir?

C2:

1 Zaman ve paradan tasarruf.

2 Kodun yeniden kullanılabilirliği.

3 Kolay raporlama.

4 Uyumluluk testi için kolaylık. (Farklı browserlarda paralel testing gibi)

Not: Soru 2 çok popüler mutlaka bilinmeli.

Q3: What would happen if we didn't import library files into our project?

A3: Testing would not take place as we could not use the pre-written codes in the library.

Q4: What are the **challenges of Selenium WebDriver**?

A4:

1 We cannot test windows application

2 We cannot test mobile apps

3 Limited reporting

4 Handling dynamic Elements.

F3: Was würde passieren, wenn wir keine Bibliotheksdateien in unser Projekt importieren würden?

A3: Es würden keine Tests stattfinden, da wir die vorab geschriebenen Codes in der Bibliothek nicht verwenden könnten.

F4: Was sind die **Herausforderungen von Selenium WebDriver**?

A4:

1 Wir können die Windows- Applikation nicht testen.

2 Wir können keine mobilen Apps testen

3 Eingeschränkte Berichterstattung

4 Handhabung mit dynamischen Elementen.

S3: Kütüphane dosyalarını projemize import etmezsek ne olur?

C3: Kütüphanede hazır yazılmış kodları kullanamadığımız için test işlemi yapılamayacaktır.

S4: Selenium WebDriver'in **zorlukları** nelerdir?

C4:

1 Windows uygulamalarını test edemiyoruz

2 Mobil uygulamaları test edemiyoruz

3 Sınırlı raporlama

4 Dinamik Elementlerin İşlenmesi.

Q5: What **type of tests** have you automated?

A5: **Regression** testing, **Smoke** & **Sanity** testing.

Q6: What is **Sanity testing**?

A6: Sanity Testing is done during the release phase to check for the main functionalities of the application without going deeper. It is also called as a subset of Regression testing. It is done at the “release level”.

Q7: How many test cases you have automated per day?

A7: Actually, it depends on Test case scenario complexity and length. I did automate 2-5 test scenarios per day when the complexity is limited.

F5: Welche Art von Tests haben Sie automatisiert?

A5: **Regressionstests**, **Smoke** & **Sanity**-Tests.

F6: Was ist **Sanity Testing**?

A6: Während der Release-Phase werden Sanity-Tests durchgeführt, um die Hauptfunktionen der Anwendung zu überprüfen, ohne tiefer zu gehen. Es wird auch als Teilmenge von Regressionstests bezeichnet. Dies erfolgt auf der „Release-Ebene“.

F7: Wie viele Testfälle haben Sie pro Tag automatisiert?

A7: Tatsächlich hängt es von der Komplexität und Länge des Testfallszenarios ab. Ich habe 2-5 Testszenarien pro Tag automatisiert, wenn die Komplexität begrenzt ist. (In Situationen mit hoher Komplexität, in denen die Komplexität sehr hoch ist, kann es Situationen geben, in denen Sie nur einen Testfall pro Tag automatisieren können.)

S5: Hangi tür testleri Automation yaptınız?

C5: Regression testing, Smoke & Sanity testleri.

S6: **Sanity testing** nedir?

C6: Daha derine inmeden uygulamanın ana işlevlerini kontrol etmek için sürüm aşamasında (Sağlık) Sanity Testi yapılır. Regresyon testinin bir alt kümesi olarak da adlandırılır. "Yayın düzeyinde" yapılır.

S7: Günde kaç test senaryosunu Automation yapıyorsunuz?

C7: Aslında Test senaryosu karmaşıklığına ve uzunluğuna bağlıdır. Karmaşıklığın sınırlı olduğu durumlarda günde 2-5 test senaryosunu yapabiliyorum.

Not: Interview'lerde ki en çok sorulan tuzak sorulardandır. Şu kadar test diye net cevap vermeniz sizi eler. Karmaşıklığın çok yüksek olduğu complexity high durumlarda günde sadece 1 tane test case i Automation yapabildiğiniz durumlar olabilir.

Q8: What is a **Framework**?

A8: A framework defines a set of rules or best practices which we can follow in a systematic way to achieve the desired results.

Q9: What are the **advantages of the Test Automation Framework**?

A9: 1 Reusability of code.

2 Easy reporting.

3 Low-cost maintenance.

4 Maximum Coverage.

Q10: Have you **created any Framework**?

A10: “Yes, I have created framework from the scratch. There was no automation process in my previous company. I designed the framework from the scratch.”

F8: Was ist **ein Framework**?

A8: Ein Framework definiert eine Reihe von Regeln oder Best Practices, die wir systematisch befolgen können, um die gewünschten Ergebnisse zu erzielen.

F9: Was sind **die Vorteile des Test Automation Framework**?

A9: 1 Wiederverwendbarkeit von Code.

2 Einfache Berichterstattung.

3 Kostengünstige Wartung.

4 Maximale Abdeckung.

F10: Haben Sie ein **Framework erstellt**?

A10: „Ja, ich habe das Framework von Grund auf neu erstellt. In meiner vorherigen Firma gab es keinen Automatisierungsprozess. Ich habe das Framework von Grund auf neu entworfen“

S8: **Framework** nedir?

C8: Framework, istenen sonuçları elde etmek için sistematik bir şekilde izleyebileceğimiz bir dizi kuralı veya en iyi uygulamayı tanımlayan yapıdır.

S9: Test Automation framework un **avantajları** nelerdir?

C9: 1 Kodun yeniden kullanılabilirliği.

2 Kolay raporlama.

3 Düşük maliyetli bakım.

4 Maksimum Kapsama.

S10: Hiç framework **oluşturdunuz** mu?

C10: Evet, sıfırdan çerçeve oluşturdum. Önceki şirketimde otomasyon süreci yoktu. Çerçeveyi sıfırdan tasarladım.

Not: Soruya vereceğiniz cevap sizin Tester’lık seviyenizi gösterir.

Başlangıç seviyesinde iseniz; hayır, sıfırdan çerçeve oluşturma şansım olmadı. (“No, I didn’t get a chance to create framework from the scratch), (“Nein, ich hatte keine Chance, ein Framework von Grund auf neu zu erstellen.)

Orta seviyede iseniz; evet, şirketimdeki diğer otomasyon Tester ile birlikte çerçeve geliştirmeye dahil oldum. (Yes, I have involved in developing framework along with other automation tester in my company), (Ja, ich habe zusammen mit anderen Automatisierungstestern in meinem Unternehmen Framework entwickelt.)

Not 2: Farklı türlerde framework var unutmayın DİKKAT biz projelerimiz de Page Factory ile birlikte POM (Page Object Model) kullanarak Data-driven Framework kullandık demelisiniz. (In our project, we are using Data-driven Framework by using Page Object Model design pattern with Page Factory.), (In unserem Projekt verwenden wir ein datengesteuertes Framework, indem wir das Seitenmuster des Seitenobjektmodells mit Page Factory verwenden.)

Q11: Why do you **prefer Selenium Automation Tool**?

A11:

1-Free and open source

2-Help

3-Cross-browser compatibility

4-Cross Platform compatibility

5-Multiple programming languages

6-Parallel Execution

7-Continuous Integration – (We can achieve nightly execution using Jenkins.)

F11: Warum **bevorzugen** Sie das **Selenium Automation Tool**?

A11:

1- Kostenlos und Open Source

2-Hilfe

3-Cross-Browser-Kompatibilität

4-Cross-Plattform-Kompatibilität

5-Mehrere Programmiersprachen

6-Parallele Exekution (Ausführung)

7-Kontinuierliche Integration - (Mit Jenkins können wir eine nächtliche Ausführung erreichen.)

S11: **Selenium Automation** aracını neden **tercih** ediyorsunuz?

C11:

1-Ücretsiz ve açık kaynak

2-Yardım

3-Tarayıcılar arası uyumluluk

4-Cross Platform uyumluluğu

5-Çoklu programlama dilleri

6-Paralel Yürütme (Execution)

7-Sürekli Entegrasyon – (Jenkins kullanarak gece yürütme yapabiliriz.)

Q12: What is **selenium**?

A12: Selenium is an open source (free) automated testing suite to test web applications. It supports different platforms and browsers.

Q13: Explain name of **components**?

A13: It has four components namely;

1. Selenium IDE (Selenium Integrated Development Environment)
2. Selenium RC (Selenium Remote Control)
3. Selenium WebDriver
4. Selenium Grid

F12: Was ist **Selenium**?

A12: Selenium ist eine Open Source (kostenlose) automatisierte Testsuite zum Testen von Webanwendungen. Es unterstützt verschiedene Plattformen und Browser.

F13: Namen der **Komponenten** erklären?

A13: Es hat vier Komponenten, nämlich;

1. Selenium-IDE (Selenium Integrated Development Environment)
2. Selenium RC (Selenium-Remote-Control- Selenium Fernbedienung)
3. Selenium WebDriver
4. Selenium Grid

S12: **Selenium** nedir?

C12: Selenium, web uygulamalarını test etmek için açık kaynaklı (ücretsiz) otomatikleştirilmiş bir test paketidir. Farklı platformları ve tarayıcıları destekler.

S13: **Birleşenlerinin** adlarını açıklayınız?

C13: 4 tane birleşene sahiptir, isimleri;

1. Selenium IDE (Selenium Integrated Development Environment- entegre geliştirme ortamı)
2. Selenium RC (Selenium Remote Control- uzaktan kontrol)
3. Selenium WebDriver
4. Selenium Grid

Q14: What is **Selenium IDE**?

A14: Selenium IDE (Integrated Development Environment) is a Firefox plugin. It is the simplest framework in the Selenium Suite. It allows us to record and playback the script.

Q15: What is **Selenese**?

A15: Selenese is the language which is used to write test scripts in Selenium IDE.

Q16: What is **Selenium WebDriver**?

A16: It is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It controls the browser by directly communicating with it.

F14: Was ist **Selenium IDE**?

A14: Selenium IDE (Integrated Development Environment) ist ein Firefox-Plugin. Es ist das einfachste Framework in der Selenium Suite. Es ermöglicht uns, das Skript aufzunehmen und wiederzugeben.

F15: Was ist **Selenese**?

A15: Selenese ist die Sprache, in der Testskripte in Selenium IDE geschrieben werden.

F16: Was ist **Selenium WebDriver**?

A16: Es ist ein Browser-Automatisierungsframework, das Befehle akzeptiert und an einen Browser sendet. Es wird über einen browserspezifischen Treiber implementiert. Es steuert den Browser durch direkte Kommunikation mit ihm.

S14: Selenium IDE nedir?

C14: Selenium IDE (Entegre Geliştirme Ortamı) bir Firefox eklentisidir. Selenium Suite'teki en basit çerçevedir. Komut dosyasını kaydetmemizi ve oynatmamızı sağlar.

S15: Selenese nedir?

C15: Selenese, Selenium IDE'de test senaryolarını yazmak için kullanılan dildir.

S16: Selenium WebDriver nedir?

C16: Komutları kabul eden ve bunları bir tarayıcıya gönderen bir tarayıcı otomasyon çerçevesidir. Tarayıcıya özgü bir sürücü aracılığıyla uygulanır. Doğrudan onunla iletişim kurarak tarayıcıyı kontrol eder.

Q17: What is **Selenium Grid**?

A17: Selenium Grid is a tool used together with Selenium RC to run tests on different machines against different browsers in parallel.

Q18: **When** do you use Selenium Grid?

A18: It can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution.

Q19: What are the **advantages** of Selenium Grid?

A19: It allows running test cases in parallel thereby saving test execution time.

It allows multi-browser testing

It allows us to execute test cases on multi-platform

F17: Was ist **Selenium-Grid**?

A17: Selenium Grid ist ein Tool, das zusammen mit Selenium-RC verwendet wird, um Tests auf verschiedenen Computern mit verschiedenen Browsern parallel durchzuführen.

F18: **Wann** verwenden Sie Selenium Grid?

A18: Es kann verwendet werden, um dieselben oder unterschiedliche Testskripte auf mehreren Plattformen und Browsern gleichzeitig auszuführen, um eine verteilte Testausführung zu erreichen.

F19: Was sind die **Vorteile** von Selenium Grid

A19: Testfälle können parallel ausgeführt werden, wodurch Testausführungszeit gespart wird.

Es ermöglicht das Testen mehrerer Browser.

Es ermöglicht uns, Testfälle auf mehreren Plattformen auszuführen.

S17: **Selenium Grid** nedir?

C17: Selenium Grid, farklı makinelerde paralel olarak farklı tarayıcılarda testler yapmak için Selenium RC ile birlikte kullanılan bir araçtır.

S18: Selenium Grid'i **ne zaman kullanırsın?**

C18: Dağıtılmış test yürütmesi elde etmek için aynı veya farklı test komut dosyalarını birden fazla platformda ve tarayıcıda aynı anda çalıştırmak için kullanılır.

S19: Selenium Grid in **avantajları** nelerdir?

C19: Test senaryolarının paralel olarak yürütülmesine izin vererek test yürütme süresinden tasarruf sağlar.

Çoklu tarayıcı testine izin verir.

Çoklu platformda test senaryoları yürütmemize olanak tanır.

Q20: What is a **hub** in Selenium Grid?

A20: A hub is a server or a central point that controls the test executions on different machines.

Q21: What is a **node** in Selenium Grid?

A21: Node is the machine which is attached to the hub. There can be multiple nodes in Selenium Grid.

F20: Was ist ein **Hub** in Selenium Grid?

A20: Ein Hub ist ein Server oder ein zentraler Punkt, der die Testausführungen auf verschiedenen Computern steuert.

F21: Was ist ein **Node** im Selenium Grid?

A21: Knoten ist die Maschine, die an den Hub angeschlossen ist. Das Selenium Grid kann mehrere Knoten enthalten.

S20: Selenium Grid 'de **Hub** nedir?

C20: Hub, farklı makinelerdeki test yürütmelerini kontrol eden bir sunucu veya merkezi bir noktadır.

S21: Selenium Grid 'de **Node** nedir?

C21: Node Hub'a bağlı olan makinedir. Selenium Grid 'de birden fazla Node olabilir.

Not: Node kelimesi düğüm, Hub ise merkez ve göbek manalarına geliyor.

Q22: What are the **Programming Languages** supported by Selenium WebDriver?

A22: **Java**, C#, Python, Ruby, Perl and PHP.

Q23: What are the **Operating Systems** supported by Selenium WebDriver?

A23: **Windows, Linux, Mac OS X, iOS and Android.**

Q24: How many **parameters** can selenium **commands** have at **minimum**?

A24: Here are four parameters that you have to pass in Selenium are **Host, Port Number, Browser** and **URL**.

F22: Welche **Programmiersprachen** werden von Selenium WebDriver unterstützt?

A22: **Java**, C#, Python, Ruby, Perl und PHP.

F23: Welche **Betriebssysteme** werden von Selenium WebDriver unterstützt?

A23: Windows, Linux, Mac OS X, iOS und Android.

F24: Wie viele Parameter können Selenium-**Command** **mindestens** haben?

A24: Hier sind vier Parameter, die Sie in Selenium übergeben müssen:

Host, Port Number, Browser und URL.

S22: Selenium WebDriver **hangi program dillerini** destekler?

C22: Java, C#, Python, Ruby, Perl ve PHP.

S23: Selenium WebDriver **hangi kullanıcı sistemlerde** çalışır?

C23: Windows, Linux, Mac OS X, iOS ve Android.

S24: Selenium da **minimum kaç tane parametreye ihtiyacınız** var?

C24: Selenium 'da geçmeniz gereken dört parametre şunlardır:

Host, Port Number, Browser ve URL.

Q25: What are the **Open-source Frameworks** supported by Selenium WebDriver?

A25: **JUnit, TestNG** and **Cucumber**

F25: Welche **Open Source-Frameworks** werden von Selenium WebDriver unterstützt?

A25: **JUnit, TestNG** und **Cucumber**

S25: Selenium WebDriver tarafından **desteklenen** Açık Kaynak Çerçeveler (**Open-source Framework**) nelerdir?

C25: **JUnit, TestNG** ve **Cucumber**

Q26: What are the **Locators** available in Selenium?

A26: In Selenium WebDriver, there are 8 different types of locators:

ID,

ClassName, Name, TagName,

LinkText, PartialLinkText,

Xpath, CSS Selector

Q27: What is an **XPath**?

A27: XPath is used to locate the elements. Using XPath, we could navigate through elements and attributes in an XML document to locate web elements such as textbox, button, checkbox, Image etc., in a web page.

F26: Welche **Locators** sind in Selenium verfügbar?

A26: In Selenium WebDriver gibt es 8 verschiedene Arten von Lokatoren:

ID,

ClassName, Name, TagName,

LinkText, PartialLinkText,

Xpath, CSS Selector

F27: Was ist ein **XPath**?

A27: XPath wird verwendet, um die Elemente zu lokalisieren. Mit XPath können wir durch Elemente und Attribute in einem XML-Dokument navigieren, um Webelemente wie Textfeld, Schaltfläche, Kontrollkästchen, Bild usw. auf einer Webseite zu finden.

S26: Selenium da **Locator** (bulucu) lar nelerdir?

C26: Selenium da 8 farklı tipte locator vardır bunlar:

ID,

ClassName, Name, TagName,

LinkText, PartialLinkText,

Xpath, CSS Selector

S27: **Xpath** nedir?

C27: Xpath, elemanları bulmak için kullanılır. Xpath kullanarak, bir web sayfasındaki metin kutusu, düğme, onay kutusu, resim vb. gibi web öğelerini bulmak için bir XML belgesindeki öğeler ve nitelikler arasında gezinebiliriz.

Q28: When you use these locators **ID, Name, Xpath** or **CSS Selector**?

A28: ID & Name locators will be used when there are unique identifiers & unique names available on the web page.

CSS Selector can be used for performance and when ID & Name locators are not unique.

XPath is used when there is no preferred locators.

Q29: What is the **difference** between “/” and “//”?

A29: Single Slash “/” → Single slash is used to create XPath with absolute path i.e. the XPath would be created to start selection from the document node/start node.

Double Slash “//” → Double slash is used to create XPath with relative path i.e. the XPath would be created to start selection from anywhere within the document.

F28: **Wann** verwenden Sie diese Locators ID, Name, XPath oder CSS Selector?

A28: ID & Name Locators werden verwendet, wenn auf der Webseite eindeutige Bezeichner und eindeutige Namen verfügbar sind.

CSS Selector kann für die Leistung verwendet werden und wenn ID & Name Locators nicht eindeutig sind.

XPath wird verwendet, wenn keine bevorzugten Locators vorhanden sind.

F29: Was ist der Unterschied zwischen "/" und "//"?

A29: Einfacher Schrägstrich "/" - Ein einzelner Schrägstrich wird verwendet, um XPath mit absolutem Path zu erstellen, d. H. Der XPath wird erstellt, um die Auswahl vom Dokumentknoten / Startknoten aus zu starten.

Doppelter Schrägstrich "/" - Der doppelte Schrägstrich wird verwendet, um XPath mit einem relativen Path zu erstellen, d. H. Der XPath wird erstellt, um die Auswahl an einer beliebigen Stelle im Dokument zu starten.

S28: ID, Name, XPath veya CSS Selector locatorlarını ne zaman kullanırsınız?

C28: ID & Name locatorları web sayfasında benzersiz tanımlayıcılar ve benzersiz adlar mevcut olduğunda kullanılacaktır.

CSS Seçici, performans için ve ID & Name locatorlar benzersiz olmadığında kullanılabilir.

XPath, tercih edilen yer belirleyici olmadığında kullanılır.

S29: Single slash "/" ile Double Slash "/" arasındaki fark nedir?

C29: Tek Eğik Çizgi "/" → Tek eğik çizgi, mutlak yolla XPath oluşturmak için kullanılır, yani XPath, belge node undan (düğümünden)/ başlangıç node dundan (düğümünden) seçimi başlatmak için oluşturulur.

Çift Eğik Çizgi "/" → Çift eğik çizgi, görelî yolla XPath oluşturmak için kullanılır, yani, belge içinde herhangi bir yerden seçime başlamak için XPath oluşturulur.

Q30: What is the difference between **Absolute Path** and **Relative Path**?

A30:

Absolute XPath starts from the root node and ends with desired descendant element's node. It starts with top HTML node and ends with input node. It starts with a single forward slash(/)

Relative XPath starts from any node in between the HTML page to the current element's node(last node of the element). It starts with a double forward slash(//).

Absolute XPath: /html/body/div[3]/div[1]/form/table/tbody/tr[1]/td/input

Relative XPath: //input[@id='email']

F30: Was ist der Unterschied zwischen dem absoluten und dem relativen Path?

A30: Absoluter XPath beginnt am Wurzelknoten und endet mit dem Knoten des gewünschten Nachkommenelements. Es beginnt mit dem obersten HTML-Knoten und endet mit dem Eingabeknoten. Es beginnt mit einem einzelnen Schrägstrich (/)

Relativer XPath startet von einem beliebigen Knoten zwischen der HTML-Seite bis zum Knoten des aktuellen Elements (letzter Knoten des Elements). Es beginnt mit einem doppelten Schrägstrich (//).

Absolute Xpath: /html/body/div[3]/div[1]/form/table/tbody/tr[1]/td/input

Relative XPath: //input[@id='email']

S30: Absolute Path ve Relative Path arasındaki fark nedir?

C30: Absolute XPath, kök düğümden başlar ve istenen alt ögenin node u (düğümlüyle) biter. En üstteki HTML düğümü ile başlar ve giriş düğümü ile biter. Tek bir eğik çizgi (/) ile başlar.

Relative XPath, HTML sayfası ile geçerli elemanın düğümünün (elemanın son düğümü) arasındaki herhangi bir düğümden başlar. Çift eğik çizgi (//) ile başlar.

Absolute Xpath e örnek: /html/body/div[3]/div[1]/form/table/tbody/tr[1]/td/input

Relative XPath e örnek: //input[@id='email']

Q31: What is the **difference** between **Assert** and **Verify** in Selenium?

A31: **Assert**: In simple words, if the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.

Verify: In simple words, there won't be any halt in the test execution even though the verify condition is true or false.

Q32: What are **Soft Assert** and **Hard Assert** in Selenium?

A32: **Soft Assert**: Soft Assert collects errors during @Test Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement.

Hard Assert: Hard Assert throws an AssertionError immediately when an assert statement fails and test suite continues with next @Test.

F31: Was ist der Unterschied zwischen Assert und Verify in Selenium?

A31: Assert: In einfachen Worten, wenn die Assert-Bedingung wahr ist, führt die Programmsteuerung den nächsten Testschritt aus. Wenn die Bedingung jedoch falsch ist, wird die Ausführung gestoppt und der weitere Testschritt wird nicht ausgeführt.

Verify: Mit einfachen Worten, die Testausführung wird nicht angehalten, obwohl die Überprüfungsbedingung wahr oder falsch ist.

F32: Was sind Soft Assert und Hard Assert in Selenium?

A32: Soft Assert: Soft Assert sammelt Fehler während @Test Soft Assert löst keine Ausnahme aus, wenn eine Assert fehlschlägt, und fährt mit dem nächsten Schritt nach der Assert-Anweisung fort.

Hard Assert: Hard Assert löst sofort eine AssertionError aus, wenn eine Assert-Anweisung fehlschlägt und Die Testsuite wird mit dem nächsten @Test fortgesetzt.

S31: Assert ile Verify arasındaki farkı açıklayınız?

C31: Assert basit bir deyişle, assert koşulu doğruysa, program kontrolü bir sonraki test adımını yürütecek, ancak koşul yanlışsa, yürütme duracak ve sonraki test adımı yürütülmeyecektir.

Verify da ise basit bir deyişle, doğrulama verify koşulu doğru veya yanlış olsa bile test yürütmede herhangi bir durma olmayacaktır.

Not: Bu soru interviewlerin en popüler soruları arasında mutlaka bilmeniz gerekli.

S32: Soft assert ve hard Assert nelerdir?

C32: Soft Assert: Soft Assert @Test sırasında hataları toplar Soft Assert, bir assert başarısız olduğunda bir istisna atmaz ve assert ifadesinden sonraki adımla devam eder.

Hard Assert: Hard Assert, bir assert ifadesi başarısız olduğunda ve test paketi sonraki @Test ile devam ettiğinde hemen bir AssertException atar.

Q33: What is the **difference** between **setSpeed()** and **sleep()** methods?

A33: setSpeed(): It set up speed that will apply a delay time before every Selenium operation.

sleep(): It set up wait only for once when called in our Selenium script.

F33: Was ist der Unterschied zwischen den Methoden setSpeed() und sleep()?

A33: setSpeed(): Es wird eine Geschwindigkeit eingerichtet, die eine Verzögerungszeit vor jeder Selenium-Operation anwendet.

sleep(): Es wird nur einmal gewartet, wenn es in unserem Selenium-Skript aufgerufen wird.

S33: setSpeed() ile sleep() methodlari arasindaki fark nedir?

A33: setSpeed (): Her Selenium işleminden önce bir gecikme süresi uygulayacak hızı ayarlar.

sleep (): Selenium betiğimizde çağrıldığında sadece bir kez beklemeyi ayarlar.

Q34: Is the ChromeDriver a **Class** or an **Interface**?

A34: ChromeDriver is a Java class, and it implements the WebDriver interface.

Q35: How to launch a browser using Selenium WebDriver?

A35: To launch Chrome Driver:

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

F34: Ist der ChromeDriver eine Klasse oder Interface (eine Schnittstelle)?

A34: ChromeDriver ist eine Java-Klasse und implementiert die WebDriver-Schnittstelle.

F35: Wie starte ich einen Browser mit Selenium WebDriver?

A35: So starten Sie den Chrome-Driver:

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

S34: ChromeDriver bir **class** mı yoksa **interface** mi?

C34: ChromeDriver bir Java **Classıdır** ve WebDriver ara yüzünü **interface** i uygular.

S35: Selenium WebDriver kullanarak bir tarayıcı nasıl başlatılır?

C35: Chrome Sürücüsünü başlatmak için:

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


Q36: What is **WebElement** selenium?

A36: WebElement in Selenium represents an **HTML element**. It basically represents a DOM element in a HTML document.

F36: Was ist **WebElement**-Selenium?

A36: WebElement in Selenium repräsentiert ein HTML-Element. Es repräsentiert im Grunde ein DOM-Element in einem HTML-Dokument.

S36: Selenium da **Webelement** nedir?

C36: Selenyumdaki WebElement bir HTML ögesini temsil eder. Temelde bir HTML belgesindeki bir DOM ögesini temsil eder.

Not: DOM (Document Object Model), Javascript'in içerdiği sayfaların verilerini görme şeklidir. Tarayıcı durumunun yanı sıra HTML / XHTML / XML'nin nasıl formatlandığını da içeren bir nesnedir.

Q37: What are the **exceptions** you have faced in Selenium WebDriver?

A37:

- 1) ElementNotVisibleException
- 2) StaleElementReferenceException
- 3) WebDriverException
- 4) IllegalStateException
- 5) TimeoutException
- 6) NoAlertPresentException
- 7) NoSuchWindowException
- 8) NoSuchElementException

F37: Mit welchen Ausnahmen (**exceptions**) sind Sie in Selenium WebDriver konfrontiert?

A37:

- 1) ElementNotVisibleException
- 2) StaleElementReferenceException
- 3) WebDriverException
- 4) IllegalStateException
- 5) TimeoutException
- 6) NoAlertPresentException
- 7) NoSuchWindowException
- 8) NoSuchElementException

S37: Selenium WebDriver da karşılaştığınız Exception lar nelerdir?

C37:

- 1) ElementNotVisibleException
- 2) StaleElementReferenceException
- 3) WebDriverException
- 4) IllegalStateException
- 5) TimeoutException
- 6) NoAlertPresentException
- 7) NoSuchWindowException
- 8) NoSuchElementException

Not: Popüler bir soru interviewlerde, bu sorunun pesine vereceğiniz cevaplardaki bir exception seçilerek onu nasıl handle edeceğiniz sorulacaktır. Örneğin; How to handle NoSuchElementException? gibi...

Cevap olarak: when the locators (i.e. id / xpath/ css selectors etc) we mentioned in the Selenium Program code is unable to find the web element on the web page and in order to handle this, we have to use NoSuchElementException WebDriver Class in the catch block.

Selenium Program kodunda bahsettiğimiz yer belirleyiciler (yani id / xpath / css seçiciler vb.) web sayfasındaki web ögesini bulamadığında ve bunun üstesinden gelmek için catch bloğunda NoSuchElementException WebDriver Class kullanmamız gerekir.

Q38: What are the types of **waits** available in Selenium WebDriver?

A38: **Implicit Waits** and **Explicit Waits**

Q39: What is **Implicit Wait** in Selenium WebDriver?

A39: It tells to the WebDriver to wait for a certain amount of time before it throws an exception. Once we set the time, WebDriver will wait for the element based on the time we set before it throws an exception.

F38: Welche Arten von **Wartezeiten** sind in Selenium WebDriver verfügbar?

A38: **Implicit Waits** und **Explicit Waits**

F39: Was ist **Implicit Waits** in Selenium WebDriver?

A39: Der WebDriver wird angewiesen, eine bestimmte Zeit zu warten, bevor eine Ausnahme ausgelöst wird. Sobald wir die Zeit festgelegt haben, wartet WebDriver basierend auf der von uns festgelegten Zeit auf das Element, bevor es eine (Ausnahme) exception. auslöst.

S38: Selenium WebDriver da hangi çeşit wait ler mevcuttur?

C38: **Implicit Waits** ve **Explicit Waits**

S39: **Implicit Wait** nedir?

C39:

WebDriver'a bir (istisna) exception atmadan önce belirli bir süre beklemesini söyler. Zamanı belirledikten sonra, WebDriver bir istisna atmadan önce belirlediğimiz süreye göre öğeyi bekleyecektir.

Q40: How to **input text** in the text box using Selenium WebDriver?

A40: By using **sendKeys()** method

Q41: How to input text in the text box without **calling** the sendKeys()?

A41: to **initialize JavaScript Object** and **using JavaScript executer...**

Q42: How to **clear** the text in the text box using Selenium WebDriver?

A42: By using **clear()** method

F40: Wie gebe ich mit Selenium WebDriver Text in das Textfeld ein?

A40: Mit der Methode `sendKeys ()`

F41: Wie kann ich Text in das Textfeld eingeben, ohne `sendKeys ()` aufzurufen?

A41: JavaScript Objekt initialisieren und JavaScript Executer verwenden ...

F42: Wie lösche ich den Text im Textfeld mit Selenium WebDriver?

A42: Mit der Methode `clear ()`

S40: Selenium WebDriver kullanarak metin kutusuna **metin nasıl girilir?**

C40: **sendKeys()** methodu kullanılarak.

S41: sendKey methodu **kullanmadan** text box a metin nasıl girilir?

C41: **JavaScript objesi initialize** edilerek ve **JavaScript executor** methodu **kullanarak**

S42: Text box içindeki metni temizlemek için hangi method kullanılır?

C42: **clear()** methodu kullanılır

Q43: How to get a **text** of a **web element**?

A43: By using `getText()` method.

F43: Wie erhalte ich einen **Text** eines **Webelements**?

A43: Mit der Methode `getText()`.

S43: Bir **WebElement** in metnini nasıl elde edebilirim?

C43: `getText()` methodunu kullanarak.

Q44: How to **press ENTER** key on text box In Selenium WebDriver?

A44: We need to use **Selenium Enum** Keys with its constant ENTER.

Example => `driver.findElement(By.xpath("xpath")).sendKeys(Keys.ENTER);`

F44: Wie drücke ich die ENTER im Textfeld in Selenium WebDriver?

A44: Wir müssen Selenium Enum Keys mit der konstanten ENTER verwenden.

Beispiel=> `driver.findElement (By.xpath ("xpath")). sendKeys (Keys.ENTER);`

S44: Selenium WebDriver'da metin kutusunda ENTER tuşuna nasıl basılır?

C44: Selenium Enum Anahtarlarını sürekli ENTER ile kullanmamız gerekiyor.

Örnek => `driver.findElement(By.xpath("xpath")).sendKeys(Keys.ENTER)`

Q45: How to pause a test execution for 5 seconds at a specific point?

A45: By using `java.lang.Thread.sleep(long milliseconds)` method we could pause the execution for a specific time. Example => `Thread.sleep(5000)`

F45: Wie kann eine Testausführung an einem bestimmten Punkt für 5 Sekunden angehalten werden?

A45: Mit der Methode `java.lang.Thread.sleep` (lange Millisekunden) können wir die Ausführung für eine bestimmte Zeit anhalten. Beispiel: `Thread.sleep(5000)`

S45: Bir testin yürütülmesi belirli bir noktada 5 saniye boyunca nasıl duraklatılır?

C45: `Java.lang.Thread.sleep` (uzun milisaniye) yöntemini kullanarak yürütmeyi belirli bir süre duraklatabiliriz. Örnek => `Thread.sleep(5000)`

Q46: What happens if I run this command. => `driver.get("www.softwaretestingmaterial.com");`

A46: If the URL doesn't contain http or https prefix then an exception is thrown. So, we need to pass HTTP protocol within `driver.get()` method.

S46: Kodu `driver.get("www.softwaretestingmaterial.com");` bu şekilde çalıştırırsam ne olur?

C46: URL, http veya https öneki içermiyorsa, bir istisna (exception hata verir) atılır. Bu yüzden, `driver.get ()` metodu içerisinde HTTP protokolünü geçmemiz gerekiyor.

F46: Was passiert, wenn ich diesen Befehl ausführe?

`driver.get("www.softwaretestingmaterial.com");`

A46: Wenn die URL kein http- oder https-Präfix enthält, wird eine Ausnahme ausgelöst. Daher müssen wir das HTTP-Protokoll innerhalb der `driver.get()` - Methode übergeben.

Q47: What is the alternative to `driver.get()` method to open an URL using Selenium WebDriver?

A47: Alternative method to `driver.get("url")` method is `driver.navigate.to("url")`

Q48: What are the different types of navigation commands?

A48:

1. `driver.navigate().forward();`
2. `driver.navigate().back();`
3. `driver.navigate().refresh();`
4. `driver.navigate().to("url");`

F47: Was ist die Alternative zur Methode `driver.get ()`, um eine URL mit Selenium WebDriver zu öffnen?

A47: Alternative Methode zur Methode `driver.get ("url")` ist `driver.navigate.to ("url")`

F48: Was sind die verschiedenen Typen von Navigationsbefehlen?

A48:

1. `driver.navigate().forward();`
2. `driver.navigate().back();`
3. `driver.navigate().refresh();`
4. `driver.navigate().to("url");`

S47: Selenium WebDriver kullanarak bir URL açmak için driver.get () yöntemine alternatif nedir?

C47: Alternatif method `http://driver.navigate.to("url");` dir.

S48: navigate komutunun çeşitleri nelerdir?

C48:

1. `driver.navigate().forward();`
2. `driver.navigate().back();`
3. `driver.navigate().refresh();`
4. `driver.navigate().to("url");`

Q49: How to fetch the current page URL in Selenium?

A49: To fetch the current page URL, we use `driver.getCurrentUrl();`

Q50: How can we maximize browser window in Selenium?

A50: we use the => `driver.manage().window().maximize();`

Q51: How to delete cookies in Selenium?

A51: using by => `driver.manage().deleteAllCookies();`

F49: Wie kann ich die aktuelle Seiten-URL in Selenium abrufen?

A49: Um die aktuelle Seiten-URL abzurufen, verwenden wir =>

```
driver.getCurrentUrl();
```

F50: Wie können wir das Browserfenster in Selenium maximieren?

A50: Wir verwenden die => `driver.manage ().window().maximize();`

F51: Wie lösche ich Cookies in Selenium?

A51: using by => `driver.manage (). DeleteAllCookies ();`

S49: Selenium 'da mevcut sayfa URL'si nasıl alınır?

C49: `driver.getCurrentUrl();` komutu ile.

S50: Selenium 'da tarayıcı penceresini nasıl büyütebiliriz?

C50: `driver.manage().window().maximize();` komutunu kullanarak.

S51: Selenium 'da çerezler nasıl silinir?

C51: `driver.manage().deleteAllCookies();` komutunu kullanarak.

Q52: What are the ways to refresh a browser using Selenium WebDriver?

A52: There are multiple ways to refresh a page in selenium,

1_Using **driver.navigate().refresh()** command ,

2_Using **driver.get("URL")** on the current URL or using **driver.getCurrentUrl()**,

3_Using **driver.navigate().to("URL")** or **driver.navigate().to(driver.getCurrentUrl())**,

4_Using **sendKeys(Keys.F5)** on any textbox on the webpage

S52: Selenium WebDriver kullanarak bir tarayıcıyı yenilemenin yolları nelerdir?

C52: Selenyumda bir sayfayı yenilemenin birden fazla yolu vardır,

1- **driver.navigate().refresh()** komutu ,

2- **driver.get("URL")** mevcut URL'de veya **driver.getCurrentUrl()**,

3- **driver.navigate().to("URL")** veya **driver.navigate().to(driver.getCurrentUrl())**,

4- Web sayfasındaki herhangi bir metin kutusunda **sendKeys(Keys.F5)** kullanma.

Not: interviewlerde popüler değil ama Seleniuma ne kadar hakim olduğunuzu gösteren bir soru.

F52: Wie kann ein Browser mit Selenium WebDriver aktualisiert werden?

A52: Es gibt mehrere Möglichkeiten, eine Seite in Selenium zu aktualisieren:

1. Verwenden von **driver.navigate().refresh()** command ,
2. Verwenden von **driver.get("URL")** für die aktuelle URL oder Verwenden von **driver.getCurrentUrl()**,
3. Verwenden von **driver.navigate().to("URL")** oder **driver.navigate().to(driver.getCurrentUrl())**,
4. Verwenden Sie **sendKeys(Keys.F5)** in einem beliebigen Textfeld auf der Webseite

Q53: What is the difference between **driver.getWindowHandle()** and **driver.getWindowHandles()** in Selenium WebDriver?

A53:

`driver.getWindowHandle()` => It returns a handle of the current page (a unique identifier)

`driver.getWindowHandles()` => It returns a set of handles of the all the pages available

Q54: What is the difference between **driver.close()** and **driver.quit()** methods?

A54:

`driver.close()`: To close current WebDriver instance

`driver.quit()`: To close all the opened WebDriver instances

F53: Was ist der Unterschied zwischen **driver.getWindowHandle ()** und **driver.getWindowHandles ()** in Selenium WebDriver?

A53:

driver.getWindowHandle () - Gibt ein Handle der aktuellen Seite zurück (eine eindeutige Kennung).

driver.getWindowHandles () - Gibt eine Reihe von Handles aller verfügbaren Seiten zurück

F54: Was ist der Unterschied zwischen den Methoden **driver.close ()** und **driver.quit ()**?

A54:

driver.close (): Zum Schließen der aktuellen WebDriver-Instanz

driver.quit (): Zum Schließen aller geöffneten WebDriver-Instanzen

S53: `driver.getWindowHandle()` ve `driver.getWindowHandles()` farkı nedir
Selenium WebDriver da?

C53: `driver.getWindowHandle ()` - Geçerli sayfanın bir tanıtıcısını döndürür
(benzersiz bir tanımlayıcı)

`driver.getWindowHandles ()` - Mevcut tüm sayfaların bir dizi tutamaçını döndürür

S54: `driver.close()` ve `driver.quit()` methodlarının farkı nedir?

C54:

`driver.close ()`: Geçerli WebDriver örneğini kapatmak için

`driver.quit ()`: Tüm açık WebDriver örneklerini kapatmak için kullanılır.

Q55: What is the difference between **driver.findElement()** and **driver.findElements()** commands?

A55:

1. `findElement()` returns a single `WebElement` (found first) based on the locator passed as parameter. Whereas `findElements()` returns a list of `WebElements`, all satisfying the locator value passed.
2. Another difference between the two is- if no element is found then `findElement()` throws `NoSuchElementException` whereas `findElements()` returns a list of 0 elements.

F55: Was ist der Unterschied zwischen den Befehlen **driver.findElement ()** und **driver.findElements ()**?

A55:

1. `findElement ()` gibt ein einzelnes `WebElement` (zuerst gefunden) basierend auf dem als Parameter übergebenen Locator zurück. Während `findElements ()` eine Liste von `WebElements` zurückgibt, die alle den übergebenen Locator-Wert erfüllen.
2. Ein weiterer Unterschied zwischen beiden besteht darin, dass `findElement ()` eine `NoSuchElementException` auslöst, wenn kein Element gefunden wird, während `findElements ()` eine Liste mit 0 Elementen zurückgibt.

S55: **driver.findElement()** ve **driver.findElements()** komutunun farkı nedir?

C55:

1. `findElement ()`, parametre olarak iletilen konumlandırıcıya bağlı olarak tek bir `WebElement` (ilk bulunur) döndürür. `FindElements ()`, tümü geçilen konum belirleyici değerini karşılayan bir `WebElement` listesi döndürür.
2. İkisi arasındaki diğer bir fark, eğer hiçbir öge bulunmazsa, `findElement ()` `NoSuchElementException` atarken, `findElements ()` 0 öğeden oluşan bir liste döndürür.

Q56: How to find whether an element is displayed on the web page?

A56:

1. `isDisplayed()`
2. `isSelected()`
3. `isEnabled()`

example: `boolean eleEnabled =>`

```
driver.findElement(By.xpath("xpath")).isEnabled();
```

F56: Wie finde ich heraus, ob ein Element auf der Webseite angezeigt wird?

A56:

1. `isDisplayed()`
2. `isSelected()`
3. `isEnabled()`

Beispiel: `boolean eleEnabled =>`

```
driver.findElement(By.xpath("xpath")).isEnabled();
```


S56: Web sayfasında bir öğenin görüntülenip görüntülenmediğini nasıl öğrenebilirim?

C56:

1. `isDisplayed()`
2. `isSelected()`
3. `isEnabled()`

Örnek: `boolean eleEnabled =>`

```
driver.findElement(By.xpath("xpath")).isEnabled();
```

Q57: How to **select** a value in a **dropdown**?

A57: By using **Select** class.

example =>

```
WebElement mySelectElement = driver.findElement(By.name("dropdown"));
```

```
Select dropdown = new Select(mySelectElement);
```

```
dropdown.selectByVisibleText(Text);
```

F57: Wie **wähle** ich einen Wert in einer **Dropdown**-Liste aus?

A57: Mit **Select class**.

Beispiel =>

```
WebElement mySelectElement = driver.findElement(By.name("dropdown"));
```

```
Select dropdown = new Select(mySelectElement);
```

```
dropdown.selectByVisibleText(Text);
```

S57: **dropdown** da bir değer nasıl seçilir?

C57: **select** class i kullanılarak.

Örnek =>

```
WebElement mySelectElement = driver.findElement(By.name("dropdown"));
```

```
Select dropdown = new Select(mySelectElement);
```

```
dropdown.selectByVisibleText(Text);
```

Not: sadece **selectByVisibleText(Text);** ile değil

selectByIndex(Index); ve **selectByValue(Value);** ile de bulabiliriz .

Q58: How to **mouse hover** on a web element using WebDriver?

A58: By using **Actions** class

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

```
Actions action = new Actions(driver);
```

```
action.moveToElement(element).perform();
```

F58: Wie kann ich mit WebDriver mit der **Maus** über ein Webelement fahren?

A58: Mit der **Action**-Klasse

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

```
Actions action = new Actions(driver);
```

```
action.moveToElement(element).perform();
```

S58: WebDriver kullanarak bir web öğesinin üzerine **fareyle** gelme olayını nasıl yaparsınız?

C58: **Action** class kullanarak.

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

```
Actions action = new Actions(driver);
```

```
action.moveToElement(element).perform();
```

Q59: How can we **handle Web-based Pop-ups** or **Alerts** in Selenium?

A59: To handle Web-based alerts or popups, we need to do switch to the alert window and call Selenium WebDriver Alert API methods.

dismiss(): To click on Cancel button.

accept(): To Click on OK button.

getText(): To get the text which is present on the Alert.

sendKeys(): To enter the text into the alert box.

Q60: How can we **handle windows-based pop up?** (**Önemli**)

A60: Selenium doesn't support windows-based applications. It is an automation testing tool which supports only web application testing. We could handle windows-based popups in Selenium using some third-party tools such as Auto IT, Robot class etc.

F59: Wie können wir mit **Web-Base Pop-ups** oder **Alerts** in Selenium umgehen?

A59: Um Web-Base Pop-ups oder Alerts zu verarbeiten, müssen Sie zum Warnfenster wechseln und die Selenium WebDriver-Alerts-API Methoden aufrufen.

dismiss (): Klicken Sie auf die Schaltfläche Abbrechen.

accept (): Klicken Sie auf OK.

getText (): Um den Text abzurufen, der in der Alert vorhanden ist.

sendKeys (): Um den Text in das Alert-Box einzugeben.

F60: Wie können wir mit **Windows-Base Pop-Up** umgehen? (**Önemli**)

A60: Selenium unterstützt keine **Windows-Base** Anwendungen. Es ist ein Automatisierungstest-Tool, das nur das Testen von Webanwendungen unterstützt. Wir könnten Windows-Base Pop-Up in Selenium mit einigen Tools von Drittanbietern wie Auto-IT, Roboterklasse usw. verarbeiten.

S59: Selenium 'da **Web Tabanlı Açılır Pencere**leri veya **Uyarı**ları nasıl handle edebiliriz?

C59: Web tabanlı uyarıları veya açılır pencereleri işlemek için, uyarı penceresine geçmemiz ve Selenium WebDriver Alert API yöntemlerini çağdırmamız gerekir.

dismiss(): To click on Cancel button.

accept(): To Click on OK button.

getText(): To get the text which is present on the Alert.

sendKeys(): To enter the text into the alert box.

S60: Windows tabanlı açılır pencereyi nasıl elde edebiliriz? (**Önemli**)

C60: Selenium, Windows tabanlı uygulamaları desteklemez. Yalnızca web uygulama testini destekleyen bir otomasyon test aracıdır. Auto-IT, Robot sınıfı vb. Gibi bazı üçüncü taraf araçları kullanarak Selenium'da pencere tabanlı açılır pencereleri yönetebiliriz.

Q61: How to **handle hidden elements** in Selenium WebDriver?

A61: We can handle hidden elements by using JavaScript executor

```
(JavascriptExecutor(driver)).executeScript("document.getElementsByClassName(ElementLocator).click();");
```

F61: Wie gehe ich mit **versteckten Elementen** in Selenium WebDriver um?

A61: Wir können versteckte Elemente mit JavaScript Executor behandeln

```
(JavascriptExecutor(driver)).executeScript("document.getElementsByClassName(ElementLocator).click();");
```

S61: **Selenium WebDriver**'da **gizli öğeler** nasıl edinilebilir?

C61: JavaScript executor kullanarak,

```
(JavascriptExecutor(driver)).executeScript("document.getElementsByClassName(ElementLocator).click();");
```

 komutu ile .

Q62: How to find more than one web element in the list? (Önemli)

F62: Wie finde ich mehr als ein Webelement in der Liste?

S62: Birden fazla WebElement i nasıl bulabiliriz? (C62: List içine alarak ve for loop döngüsü kurarak.)

```
A62:                                     // To store the list
List <WebElement> elementList = driver.findElement(By.xpath("xpath"));
int listSize = elementList.size();       // To fetch the size of the list
for (int i=0; i<listSize; i++)
{
    links.get(i).click();                 // Clicking on each link
    driver.navigate().back();             // Navigating back to the previous page that stores the links
}
```

Q63: How do you **read test data** from **excels**?

A63: Using JXL or Apache POI.

F63: Wie liest man Testdaten von Excel?

A63: Verwenden von JXL oder Apache POI.

S63: Excel den test data lari nasıl okuruz?

C63: JXL veya Apachi-POI kullanarak.

Q64: Is it possible to automate the **captcha** using Selenium?

A64: No, it's not possible to automate captcha and bar code reader.

F64: Ist es möglich, das Captcha mit Selenium zu automatisieren?

A64: Nein, Captcha und Barcode-Leser können nicht automatisiert werden.

S64: Captcha'yı Selenium kullanarak otomatikleştirmek mümkün mü?

C64: Hayır, captcha ve barkod (QR) okuyucuyu otomatikleştirmek mümkün değildir.

Q65: Can You Use Selenium for **Rest API Testing** or **Web Services Testing**?

A65: Selenium is not a tool for API Testing. It automates web browsers. Rest API & Web Services contains no UI. So, we cannot automate using Selenium.

F65: Können Sie Selen für Rest-API-Testing oder Web Services Testing verwenden?

A65: Selen ist kein Werkzeug für API-Tests. Es automatisiert Webbrowser. Rest API & Web Services enthält keine Benutzeroberfläche. Daher können wir die Verwendung von Selenium nicht automatisieren.

S65: Rest API Testi veya Web Service Testi için Selenium kullanabilir misiniz?

C65: Selenium, API Testi için bir araç değildir. Web tarayıcılarını otomatikleştirir. Rest API ve Web Hizmetleri, UI içermez. Bu yüzden Selenium'u kullanarak otomatik hale getiremiyoruz.

Q66: List some **scenarios** which we **cannot automate** using **Selenium WebDriver**?

A66: 1. Bitmap comparison is not possible using Selenium WebDriver
2. Automating Captcha is not possible using Selenium WebDriver
3. We cannot read bar code using Selenium WebDriver

F66: Einige Szenarien auflisten, die wir mit Selenium WebDriver nicht automatisieren können? (**Önemli**)

A66: 1. Ein Bitmap-Vergleich ist mit Selenium WebDriver nicht möglich
2. Die Automatisierung von Captcha ist mit Selenium WebDriver nicht möglich
3. Mit Selenium WebDriver können wir keinen Barcode lesen

S66: Selenium WebDriver kullanarak otomatikleştiremediğimiz bazı senaryoları listeleyin. (**Önemli**)

C66: 1. Selenium WebDriver kullanılarak bit eşlem karşılaştırması yapılamaz
2. Captcha'yı otomatikleştirmek Selenium WebDriver kullanarak mümkün değildir
3. Selenium WebDriver kullanarak barkodu okuyamıyoruz.

Q67: What is **Object Repository** in Selenium WebDriver?

A67: We create a property file (.properties) to store all the element locators and these property files act as an object repository in Selenium WebDriver.

Q68: How you **build Object Repository** in your project? (**Önemli**)

A68: The best way is to use Page Object Model. In the Page Object Model Design Pattern, each web page is represented as a class.

F67: Was ist das **Objekt-Repository** in Selenium WebDriver?

A67: Wir erstellen eine Eigenschaftendatei (.properties) zum Speichern aller Element-Locators. Diese Eigenschaftendateien fungieren als Objekt-Repository in Selenium WebDriver.

F68: Wie **erstellen** Sie das **Objekt-Repository** in Ihrem Projekt?

A68: Der beste Weg ist die Verwendung des Seitenobjektmodells. Im Entwurfsmuster für Seitenobjektmodelle wird jede Webseite als Klasse dargestellt.

S67: Selenium WebDriver 'daki **Object repository** (Nesne Deposu) nedir?

C67: Tüm eleman locatorları depolamak için bir özellik dosyası (.properties) oluştururuz ve bu özellik dosyaları Selenium WebDriver 'da bir (nesne deposu) Object repository görevi görür.

S68: Projenizde Object repository nesne deposunu nasıl oluşturursunuz? (**Önemli**)

C68: En iyi yol, Page Object Model (POM) kullanmaktır. Sayfa Nesne Modeli Tasarım Deseninde, her web sayfası bir class olarak temsil edilir.

Sharing is caring – Paylaşmak önemsemektir.

Q69: What is **Page Object Model (POM)** in Selenium?

A69: Page Object Model is a Design Pattern which has become popular in Selenium Test Automation. It is widely used design pattern in Selenium for enhancing test maintenance and reducing code duplication. Page object model (POM) can be used in any kind of framework such as modular, data-driven, keyword driven, hybrid framework etc.

Q70: What is **Page Factory**?

A70: We have seen that 'Page Object Model' is a way of representing an application in a test framework. For every 'page' in the application, we create a Page Object to reference the 'page' whereas a 'Page Factory' is one way of implementing the 'Page Object Model'.

F69: Was ist das **Page Object Model (POM)** in Selenium?

A69: Das Seitenobjektmodell ist ein Entwurfsmuster, das in der Selenium Test Automation populär geworden ist. Es ist ein weit verbreitetes Entwurfsmuster in Selenium, um die Testwartung zu verbessern und die Codeduplizierung zu reduzieren. Das Page Object Model (POM) kann in jeder Art von Framework verwendet werden, z. B. in modularen, datengesteuerten, schlüsselwortgesteuerten, hybriden Frameworks usw.

F70: Was ist Page Factory?

A70: Wir haben gesehen, dass 'Page Object Model' eine Möglichkeit ist, eine Anwendung in einem Testframework darzustellen. Für jede "Seite" in der Anwendung erstellen wir ein Seitenobjekt, um auf die "Seite" zu verweisen, während eine „Page Factory “ eine Möglichkeit ist, das "Page Object Model" zu implementieren.

S69: **POM** nedir?

C69: POM, Selenium Test Otomasyonunda popüler hale gelen bir Tasarım Modelidir. Selenium'da test bakımını geliştirmek ve kod tekrarını azaltmak için yaygın olarak kullanılan tasarım modelidir. Sayfa nesne modeli (POM); modüler, veriye dayalı, anahtar kelimeye dayalı, hibrit çerçeve vb. Gibi her türlü (çerçevede) framework ta kullanılabilir.

S70: **Page Factory** nedir?

C70:"POM" un bir uygulamayı test çerçevesinde temsil etmenin bir yolu olduğunu gördük. Uygulamadaki her "sayfa" için, "sayfa" yı referans alan bir Sayfa Nesnesi oluştururken Page Factory "Sayfa Fabrikası", "Sayfa Nesne Modeli" ni uygulamanın bir yoludur.

Q71: What is the **difference** between Page Object Model (POM) and Page Factory?

A71:

Page Object is a class that represents a web page and hold the functionality and members.

Page Factory is a way to initialize the web elements you want to interact with within the page object when you create an instance of it.

Q72: What are the **advantages** of Page Object Model Framework?

A72:

1. Code reusability;
2. Code maintainability;
3. Object Repository;
4. Readability

F71: Was ist der **Unterschied** zwischen Page Object Model und Page Factory?

A71: Page Object ist eine Klasse, die eine Webseite darstellt und die Funktionen und Mitglieder enthält.

Page Factory ist eine Möglichkeit, die Webelemente zu initialisieren, mit denen Sie innerhalb des Seitenobjekts interagieren möchten, wenn Sie eine Instanz davon erstellen.

F72: Was sind die **Vorteile** von Page Object Model Framework?

A72:

1. Wiederverwendbarkeit des Codes;
2. Code-Wartbarkeit;
3. Objekt-Repository;
4. Lesbarkeit.

S71: Page Object Model ile Page Factory arasındaki fark nedir?

C71: Page Object, bir web sayfasını temsil eden ve işlevselliği ve üyeleri tutan bir sınıftır.

Page Factory, sayfa nesnesinin bir örneğini oluşturduğunuzda etkileşime girmek istediğiniz web öğelerini başlatmanın bir yoludur.

S72: Page Object Model framework un **avantajları** nelerdir?

C72:

1-Kodun yeniden kullanılabilirliği;

2-Kod sürdürülebilirliği;

3-Nesne Deposu;

4-Okunabilirlik

Q73: How can you use the **Recovery Scenario** in Selenium WebDriver?

A73: By using “**Try Catch Block**” within Selenium WebDriver Java tests.

example:

```
try {  
    driver.get("www.softwaretestingmaterial.com");  
} catch (Exception e) {  
    System.out.println(e.getMessage());  
}
```

F73: Wie können Sie das **Recovery Scenario** (Wiederherstellungsszenario) in Selenium WebDriver verwenden?

A73: Durch Verwendung von "**Try Catch Block**" in Selenium WebDriver Java-Tests.

Beispiel:

```
try {  
    driver.get("www.softwaretestingmaterial.com");  
} catch (Exception e) {  
    System.out.println(e.getMessage());  
}
```

S73: Selenium WebDriver'da Kurtarma Senaryosunu nasıl kullanabilirsiniz?

C73: **Try Catch Block** kullanarak.

Örneğin:

```
try {  
    driver.get("www.softwaretestingmaterial.com");  
} catch (Exception e) {  
    System.out.println(e.getMessage());  
}
```

Q74: How to **Upload** a file in Selenium WebDriver?

A74: There are two cases which are majorly used to upload a file in Selenium WebDriver such as using **SendKeys Method** and using **Auto IT Script**.

Q75: How to **Download** a file in Selenium WebDriver?

A75: By using **Auto IT script**, we could download a file in Selenium WebDriver.

F74: Wie **lade** ich eine Datei in Selenium WebDriver **hoch**?

A74: Es gibt zwei Fälle, die hauptsächlich zum Hochladen einer Datei in Selenium WebDriver verwendet werden, z. B. die **SendKeys-Methode** und **Auto IT-Skript**.

F75: Wie **lade** ich eine Datei in Selenium WebDriver **herunter**?

A75: Mit dem **Auto IT-Skript** konnten wir eine Datei in Selenium WebDriver herunterladen.

S74: Selenium WebDriver'da nasıl dosya **yüklenir**?

C74: Selenium WebDriver'da bir dosya yüklemek için kullanılan, **SendKeys Yöntemi** ve **Auto IT Komut Dosyası** gibi başlıca iki durum vardır.

S75: Selenium WebDriver'da bir dosya nasıl **indirilir**?

C75: **Auto IT komut dosyasını** kullanarak, Selenium WebDriver'da bir dosya indirebiliriz.

Q76: How to **switch between frames** in Selenium?

A76: By using the following code, we could switch between frames.

```
driver.switchTo().frame();
```

F76: Wie **wechsle** ich in Selenium **zwischen Frames**?

A76: Mit dem folgenden Code können wir zwischen Frames wechseln.

```
driver.switchTo().frame();
```

S76: Selenium'da (çerçeve) frameler arasında nasıl geçiş yapılır?

C76: Aşağıdaki kodu kullanarak çerçeveler arasında geçiş yapabiliriz

```
driver.switchTo().frame();
```


Q77: How to **connect** a **Database** in selenium?

A77: We use **JDBC Driver** to connect the Database in Selenium (While using Java Programming Language).

F77: Wie **verbinde** ich eine **Database** in Selenium?

A77: Wir verwenden den **JDBC-Driver**, um die Datenbank in Selen zu verbinden (unter Verwendung der Java-Programmiersprache).

S77: Selenyumda bir **Veritabanına** nasıl **bağlanılır**?

C77: Veritabanına Selenium'da bağlamak için **JDBC** Sürücüsünü kullanıyoruz (Java Programlama Dilini kullanırken).

Q78: How to **Scroll Web Page Down** or **UP** Using Selenium WebDriver?

A78: **JavaScript scrollBy()** method scrolls the document by the specified number of pixels.

Q79: How to Perform **Drag and Drop** Action in Selenium WebDriver?

A79: To achieve this, we use **Actions class** in Selenium WebDriver.

F78: Wie **scrolle** ich mit Selenium WebDriver nach **unten** oder **oben**?

A78: Die JavaScript-Methode **scrollBy ()** scrollt das Dokument um die angegebene Pixelanzahl.

F79: Ausführen einer **Drag & Drop**-Aktion in Selenium WebDriver?

A79: Um dies zu erreichen, verwenden wir die **Actions Class** in Selenium WebDriver.

S78: Selenium WebDriver kullanarak **Web Sayfasını Aşağı veya Yukarı Kaydırma**

C78: JavaScript **scrollBy ()** yöntemi, belgeyi belirtilen piksel sayısı kadar kaydırır.

S79: Selenium WebDriver'da (Sürükle Bırak) **Drag and Drop** işlemi nasıl gerçekleştirilir?

C79: Bunu başarmak için Selenium WebDriver'da **Actions Class** kullanıyoruz.

Q80: What is **Continuous Integration**?

A80: Continuous integration is a **process of verifying the correctness of a software**. Some of the continuous integration tools are **Jenkins**, TeamCity, Bamboo, Travis, Circle Ci, Bitbucket.

F80: Was ist kontinuierliche Integration?

A80: Bei der kontinuierlichen Integration wird die Richtigkeit einer Software überprüft. Einige der Tools für die kontinuierliche Integration sind Jenkins, TeamCity, Bamboo, Travis, Circle Ci und Bitbucket.

S80: Sürekli Entegrasyon nedir?

C80: sürekli entegrasyon, **bir yazılımın doğruluğunu doğrulama sürecidir**. Sürekli entegrasyon araçlarından bazıları **Jenkins**, TeamCity, Bamboo, Travis, Circle Ci, Bitbucket'tir.