Expected Conditions provided by [Selenium WebDriver](https://www.lambdatest.com/blog/selenium-webdriver-tutorial-with-examples/) are used for performing Explicit Waits (Явные ожидания) on a certain condition. The Selenium WebDriver waits for the specified condition to occur before it can proceed further with the execution.

In Implicit Wait, the DOM is polled/опрошенный by the Selenium WebDriver for a specified amount of time when a particular WebElement is not immediately available. Once Implicit Wait is set, it is available for the entire life of the WebDriver object.

On the other hand, Explicit Wait using ExpectedConditions in Selenium lets you wait for the occurrence of a specified condition before execution can proceed to the next step. If the condition is not met within the expected time duration, an appropriate exception is raised.

driver = new ChromeDriver();

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

By elem\_dynamic = By.id("dynamic-element");

wait.until(ExpectedConditions.presenceOfElementLocated(elem\_dynamic));

In the above snippet demonstrating expected conditions in Selenium Java, the Selenium WebDriver waits for a maximum duration of 10 seconds until the WebElement with ID ‘dynamic element is found. The ExpectedCondition is invoked by the Selenium WebDriver every 500 ms until it returns success. If the WebElement is available within 10 seconds (i.e. maximum wait duration), the ExpectedCondition returns true, and execution proceeds to the next step. A TimeoutException is thrown if the element is not present in the DOM even at the elapse/истекать of the maximum duration of 10 seconds.

**Types of Expected Conditions In Selenium Java:**(or categories)

-**ExpectedCondition <WebElement>** -the locator of the WebElement is used as the parameter in the ExpectedCondition. Depending on the condition type, the explicit wait is performed until the expected condition is satisfied or the wait duration has elapsed. For example:

**1)**.numberOfElementsToBeLessThan() - method in ExpectedConditions class returns a List of WebElements if the number of WebElements located using the web locator (passed as the argument) is less than the expected number (also passed as an argument).

**2)**.elementToBeClickable(Webelement element) - method in Expected Conditions in Selenium Java returns a WebElement if the located element is clickable (i.e. it can be clicked). The appropriate [Selenium locator](https://www.lambdatest.com/learning-hub/selenium-locators) for locating the WebElement is passed to the method. На мой взгляд метод не работает(не рекомендую)

**3).wait.until(ExpectedConditions.elementToBeSelected(Webelement element)));**предпологается, что произойдёт ожидание пока элемент не будет выбран. Я попробовал с разными элементами, но всегда получал Exception. Возможно предполагается,что элемент имеет какой то атрибут, значение которого изменяется после клика, на этот атрибут “смотрит” данный метод, **но** если такого атрибута у элемента нет то и зацепится данному методу не за что, вот он и выдаёт Exception

**4)wait.until(ExpectedConditions.visibilityOf​(WebElement element));**ожидание будет продолжатся заданное количество времени до тех пор пока элемент не начнёт быть видимым, если по истечению заданного количества времени элемент всё же не стал видимым то Exception выбрасывается К примеру хотим кликнуть по элементу а он долго не загружается, для этого и используем явное ожидание и данный метод, что-то типа ожидать до того как элемент станет видимым, когда он станет видимым кликнуть по нему.

**5)ExpectedConditions.visibilityOfElementLocated​(By locator)** Ожидает до появления указанного элемента указанного по локатору, в случае успешного выполнения возвращает Webelement указанный в локаторе. По сути очень схож с методом **visibilityOf(Webelement element)**

**6)wait.until(ExpectedConditions.invisibilityOf(Webelement element));** Ожидание будет продолжаться до тех пор пока элемент не исчезнет, т.е. если элемент будет всё ещё видим то программа дальше выполнятся не будет и если элемент не проподёт в течение заданного времени то Exception выбросится. Применять - к примеру если этот элемент уже невидим то кликнуть туда-то…

-**ExpectedCondition <WebDriver>** - This category of ExpectedCondition **returns** a WebDriver instance after the required operation is successfully performed.For example:

**1)** .frameToBeAvailableAndSwitchToIt() - method in Expected Conditions in Selenium Java switches the WebDriver to the specified frame that is located using the frameLocator (which is passed as the argument to the method). The frame can be located using the ID or Name web locators.

**-ExpectedCondition <Boolean>** - The condition of the type Boolean takes a String parameter, and the wait is applied to the condition of the parameter. On successful execution of the condition, the Boolean value **true** is returned, whereas **false** is returned if the condition is not met.For example: **1)**.textToBePresentInElementLocated() - method returns **true** when the WebElement located by the web locator (which is passed as a parameter to the method) contains the specified text.

**2).** **wait.until(ExpectedConditions.attributeContains(By Locator, String attribute, String value)**; мы ждём появления элемента с таким-то атрибутом, и предпологаем, что в значении этого атрибута будут присутствовать такие-то параметры, мы как бы не знаем всего значения, а предпологаем вхождение наших параметров в значение, если это так то вернётся true Пример:wait.until(ExpectedConditions.attributeContains(By.xpath("//script[@type='text/template']"),"id","temp"))

**3)wait.until(ExpectedConditions.attributeContains(Webelement element, String attribute, String value)**;то же самое но первым элементом передаётся Webelement Пример wait.until(**ExpectedConditions.attributeContains(element,"id","temp"**)));

**4).ExpectedConditions.attributeToBe(Webelement element, ”attribute”, “value”);** ждём появления элемента с определенным атрибутом с определённым значением точным значением Пример ExpectedConditions.attributeToBe(element,"id","template-demo2"))); Но лучше сначала вывести в консоль значение данного атрибута в данном элементе с помощью element.getAttribute(“название атрибута”); так-как значения могут отличаться от представленных в DOM, т.е. данный метод достаточно привередлив по сравнению с .attributeContains().

**5).attributeToBeNotEmpty(element, String “имя атрибута”);**ждём определённый элемент(который передаём в параметрах) с определённым атрибутом (его название-это второй параметр)

**-ExpectedCondition <Alert>** - This category of Expected Conditions in Selenium Java returns an alert if a window of type Alert is present on the page. Once the alert is present, the WebDriver switches to the Alert window. Post switching, various operations like Accept(), Dismiss(), sendKeys(), and getText() can be performed on the Alert Window.