### **Expected Condition**

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[contains(text(),'COMPOSE')]")));

drv.findElement(By.xpath("//div[contains(text(),'COMPOSE')]")).click();

The above command waits for a stipulated amount of time or an expected condition to occur whichever occurs or elapses first.

Thus to be able to do this, we use the “wait” reference variable of WebDriverWait class created in the previous step with ExpectedConditions class and an actual condition which is expected to occur. Therefore, as soon as the expected condition occurs, the program control would move to the next execution step instead of forcefully waiting for the entire 30 seconds.

In our specimen, we wait for the “compose” button to be present and loaded as a part of home page load and thus, then we move forward with calling the click command on the “compose” button.

**Types of Expected Conditions**

ExpectedConditions class provides a great help to deal with scenarios where we have to ascertain for a condition to occur before executing the actual test step.

ExpectedConditions class comes with a wide range of expected conditions that can be accessed with the help of the WebDriverWait reference variable and until() method.

**Let us discuss a few of them at length:**

**#1) elementToBeClickable()** – The expected condition waits for an element to be clickable i.e. it should be present/displayed/visible on the screen as well as enabled.

**Sample Code**  
wait.until(ExpectedConditions.elementToBeClickable(By.xpath(“//div[contains(text(),’COMPOSE’)]”)));

**#2) textToBePresentInElement() –** The expected condition waits for an element having a certain string pattern.

**Sample Code**  
wait.until(ExpectedConditions.textToBePresentInElement(By.xpath(“//div[@id= ‘forgotPass'”), “text to be found”));

**#3) alertIsPresent()-** The expected condition waits for an alert box to appear.

**Sample Code**  
wait.until(ExpectedConditions.alertIsPresent()) !=null);

**#4) titleIs()** – The expected condition waits for a page with a specific title.

**Sample Code**  
wait.until(ExpectedConditions.titleIs(“gmail”));

**#5) frameToBeAvailableAndSwitchToIt()** – The expected condition waits for a frame to be available and then as soon as the frame is available, the control switches to it automatically.

**Sample Code**  
wait.until(ExpectedConditions.frameToBeAvailableAndSwitchToIt(By.id(“newframe”)));

### **Navigation Using WebDriver**

There is a very common user action where the user clicks on the back and forward buttons of the web browser back n forth to navigate to the different web pages visited in the current session on the browser’s history. Thus to simulate such actions performed by the users, WebDriver introduces Navigate commands.

**Let us examine these commands in detail:**

**#1) navigate().back()**  
This command lets the user to navigate to the previous web page.

**Sample code:**  
driver.navigate().back();  
The above command requires no parameters and takes back the user to the previous webpage in the web browser’s history.

**#2) navigate().forward()**  
This command lets the user to navigate to the next web page with reference to the browser’s history.

**Sample code:**  
driver.navigate().forward();  
The above command requires no parameters and takes forward the user to the next webpage in the web browser’s history.

**#3) navigate().refresh()**  
This command lets the user to refresh the current web page there by reloading all the web elements.

**Sample code:**  
driver.navigate().refresh();  
The above command requires no parameters and reloads the web page.

**#4) navigate().to()**  
This command lets the user to launch a new web browser window and navigate to the specified URL.

**Sample code:**  
driver.navigate().to(“http://google.com”);  
The above command requires a web URL as a parameter and then it opens the specified URL on a freshly launched web browser.