Synchronization is to make Program execution and Application response with the same speed

Program doesn’t know anything about application. It simply executes one by one step. Application takes minimum amount of time for loading the controls. In such cases program may fail.

We are using PageLoadTimeout and ImplicitlyWait which works as some default wait for page navigations/For element load.

In case if any element is taking more time than implicitly wait, or if we have to wait for element visibility/enable/disable/attribute we have to use explicit waiting.

PageLoadTimeout/Implicitly wait are part of Implicit wait. These are not specific to single element.

We can make program to wait for a specific situation using explicit wait.

Thread.Sleep given by java and wait for only time and not for element.

As part of synchronization we have to

* Wait for element existence
* Wait for Element State
  + We can use webdriverwait class for above two situations
* Wait for time
  + can be done by using thread.sleep
  + It is static

**Fluent Wait**

We can write custom condition waiting using fluent wait. Each FluentWait instance defines the maximum amount of time to wait for a condition, as well as the frequency with which to check the condition. Furthermore, the user may configure the wait to ignore specific types of exceptions whilst waiting, such as NoSuchElementExceptions when searching for an element on the page.

If explicit wait is failed it throws timeoutexception.

**public** **static** **boolean** isElementExist(WebDriver driver, By locator) {

Wait<WebDriver> wait = **new** FluentWait<WebDriver>(driver)

.withTimeout(30, TimeUnit.***SECONDS***)

.pollingEvery(2, TimeUnit.***SECONDS***).ignoring(NoSuchElementException.**class**);

**boolean** elmFound = wait.until(**new** Function<WebDriver, Boolean>() {

**public** Boolean apply(WebDriver driver) {

//custom condition code

System.***out***.println("condition executed");

driver.findElement(locator);

**return** **true**;

}

});

**return** elmFound;

}