Selenium Waits

These days most web apps are using AJAX techniques and angular. When the browser loads a page, the elements within that page may load at different time intervals. This makes locating elements difficult: if an element is not yet present in the DOM, a find element function will raise an ElementNotVisibleException exception.  
  
Using waits, we can solve this issue. Waiting provides some slack between locating an element and operating on the element.  
  
Selenium Python provides two types of waits -**implicit & explicit**. An explicit wait makes selenium wait for a specific condition to occur before proceeding further with execution.  
  
An implicit wait makes selenium python poll the DOM for a certain amount of time with a 300ms interval when trying to locate an element.

Implicit wait:

Implicitly wait is one of the ways to request selenium not throw any exception until provided time. **A default wait time of the selenium is 500 milliseconds**, implicitly wait overrides the default wait time of the selenium python.

If the element is found before implicitly wait time, selenium moves to the next commands in the program without waiting further, this wait is also called dynamic wait.  
  
The implicit wait is set for the entire duration of your selenium driver and is set at the start of your program. Most of the automation tester writes the implicit wait after the creation of the browser object.  
  
Let's consider the implicit wait of 30 seconds, Implicit wait tries to find the element in the first go, if the element is not present implicit wait tries to find the element after 300ms of first polling if the element is not available on the second time also, then implicit wait tries the third time after 300 ms of the second try and it goes on till the time reaches the 30 seconds.  
  
What it does is, if your selenium python doesn't find any element then instead of throwing an exception, the implicit wait makes your driver wait for the specified wait time and then try to find the element once again till the time limit is reached.  
  
If the driver still does not find the element, then it throws an exception. Implicit wait does the same for all the elements in your program, so you just have to set it once.

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**driver.implicitly\_wait(30)**

Explicit waits

The explicit wait is used to tell the Web Driver to wait for specific conditions or the maximum time limit before throwing an Exception.  
  
We can reuse the WebdriverWait object once we create it. The explicit wait will be applicable for only one line, and we have to use it with ExpectedConditions class.  
  
ExplicitWait does not have any effect on find element and find elements. ExplicitWait also called WebdriverWait. WebDriverWait by default calls the ExpectedCondition every 500 milliseconds until it returns successfully.

**Syntax : wait=new WebDriverWait( driver, timeoutInSeconds);**

Explicit waits are an excellent way to organize a test script and provide more flexibility, by allowing us to design out tests in a way, where we can wait for some predefined or custom conditions and then carry on with what we want.  
  
Below code waits for the element to become clickable

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**driver.get("https://chercher.tech/practice/explicit-wait-sample-selenium-webdriver");**

**wait = new WebDriverWait(driver, 30 /\*timeout in seconds\*/);**

**wait.until(ExpectedConditions.element\_to\_be\_clickable(By.xpath("//button[@id='btn1']"))));**

Scenarios:

Validate whether products selected in Page1 are showing in Page2 check page

Verify if Price Decreases on Discount

Verify if sum of products in checkout page matches with Total amount

Verify is search functionality in home page is working or not