**+Waits in Selenium Python:**

**time.sleep(seconds)** is used as a static wait in python.

The types of Selenium WebDriver waits are:

* **Implicit Wait**
* **Explicit Wait**

**Implicit Wait:**

An implicit wait instructs Selenium WebDriver to poll DOM for a certain amount of time, this time can be specified, when trying to find an element or elements that are not available immediately. The default setting is 0 seconds which means WebDriver will not wait before any operations on element.Once set, the implicit wait is set for the life of the WebDriver object i.e. all actions will be delayed by given time.

**Syntax:**  
driver.implicitly\_wait(10)

Pass number of seconds to wait as an argument

from selenium import webdriver

driver = webdriver.Firefox()

driver.implicitly\_wait(15)

driver.get("http://url")

driver.find\_element\_by\_id("id\_of\_element").click()

**Explicit Wait:**

Explicit wait is used to specify wait condition for a particular element. Here we define to wait for a certain condition to occur before proceeding further in the code.There can be instance when a particular element takes more than usual time to load. In that case no need to set a huge time to implicit wait, as this will make browser to wait for the same time for every element. To avoid that situation explicit wait is used by defining a separate wait time only on the required element.

There are some predefined methods provided that will help your script to wait only as long as required. WebDriverWait in combination with ExpectedCondition is one way this can be accomplished.

Following two packages are required to set explicit wait  
**from selenium.webdriver.support.ui import WebDriverWait  
from selenium.webdriver.support import expected\_conditions as ec**

from selenium import webdriver

fromselenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.common.by import By

fromselenium.webdriver.support.ui import WebDriverWait

fromselenium.webdriver.support import expected\_conditions as ec

driver = webdriver.Chrome()

driver.get("http://url")

driver.maximize\_window()

# wait for element to appear, then hover it

wait = WebDriverWait(driver, 10)

men\_menu = wait.until(ec.visibility\_of\_element\_located((By.XPATH, "//a[@data-tracking-id='men']")))

ActionChains(driver).move\_to\_element(men\_menu).perform()

# wait for Fastrack menu item to appear, then click it

fastrack = WebDriverWait(driver, 10).until(ec.visibility\_of\_element\_located((By.XPATH, "//a[@data-tracking-id='0\_Fastrack']")));

fastrack.click()

**Standard Expected Conditions:**

There is a no. of standard conditions which you may commonly encounter while automating the web pages. Below is the list displaying the names of each of them. All of these classes are available in the **“selenium.webdriver.support.expected\_conditions”** Python module.

1. **classalert\_is\_present**  
   It allows waiting for an alert to appear.
2. **classelement\_located\_selection\_state\_to\_be(ui\_locator, is\_selected)**  
   It allows waiting for an element to locate having a specified state.
3. **classelement\_located\_to\_be\_selected(ui\_locator)**  
   It allows waiting for an element to locate in the selected state.
4. **classelement\_selection\_state\_to\_be(ui\_element, is\_selected)**  
   It allows waiting to find an element having a specified state.
5. **classelement\_to\_be\_clickable(ui\_locator)**  
   It allows waiting for an element to locate which is in a clickable state.
6. **classelement\_to\_be\_selected(ui\_element)**  
   It allows waiting for an element to find which is in a selected state.
7. **classframe\_to\_be\_available\_and\_switch\_to\_it(ui\_locator)**  
   It allows waiting for a frame to become available for the switchover.
8. **classinvisibility\_of\_element\_located(ui\_locator)**  
   It allows waiting for an element which is invisible or not in the DOM.
9. **classstaleness\_of(ui\_element)**  
   It allows waiting for an element getting removed from the DOM.
10. **classtext\_to\_be\_present\_in\_element(ui\_locator, inner\_text)**  
    It allows waiting for an element to find with a given text.
11. **classtext\_to\_be\_present\_in\_element\_value(ui\_locator, value)**  
    It allows waiting for an element to find with a given text inside the locator.
12. **classtitle\_contains(title\_text)**  
    It allows waiting to find a title containing the specified text in a case-sensitive format.
13. **classtitle\_is(title)**  
    It allows waiting to find a title matching the exact specified text.
14. **classvisibility\_of(ui\_element)**  
    It allows waiting to find an element which is functionally visible.
15. **classvisibility\_of\_all\_elements\_located(ui\_locator)**  
    It allows waiting to find all functionally visible elements the specified locator will return.
16. **classvisibility\_of\_any\_elements\_located(ui\_locator)**  
    It allows waiting to find at least one functionally visible elements the specified locator will return.
17. **classvisibility\_of\_element\_located(ui\_locator)**  
    It allows waiting to find a specific and functionally visible element the specified locator will return.

**Fluent Wait:**

This waits up to 10 seconds before throwing a TimeoutException unless it finds the element to return within 10 seconds. WebDriverWait by default calls the ExpectedCondition every 500 milliseconds, which is called poll frequency, until it returns successfully. A successful return is for ExpectedCondition type is Boolean return true or not null return value for all other ExpectedCondition types.

We have the option to change the poll frequency by passing keyword argument ‘poll\_frequency’ which accepts values in seconds.Default value is 0.5 second. For instance, lets change poll frequency to 1 second.

wait = WebDriverWait(driver, 10, poll\_frequency=1)