WebDriver Commands:

1. void get(String url);
2. @Nullable String getCurrentUrl();
3. @Nullable String getTitle();
4. @Nullable String getPageSource();
5. List<WebElement> findElements(By by);
6. WebElement findElement(By by);
7. void close(); // no dispose method in java
8. void quit();
9. Set<String> getWindowHandles();
10. String getWindowHandle();
11. TargetLocator switchTo();
12. Navigation navigate();
13. Options manage();

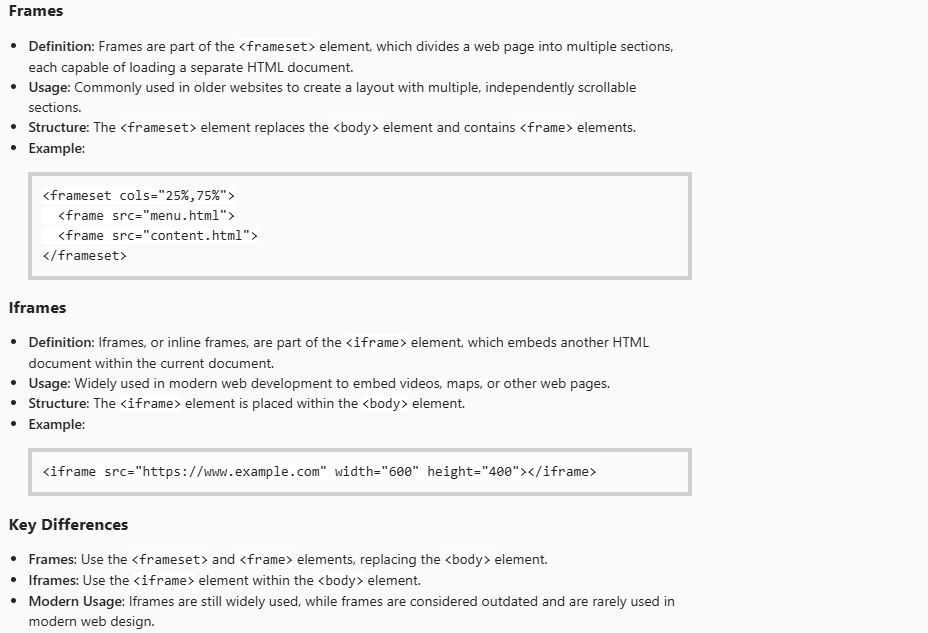
Web Element Commands:

1. void click();
2. void submit();
3. void sendKeys(CharSequence... keysToSend);
4. void clear();
5. boolean isSelected();
6. boolean isEnabled();
7. boolean isDisplayed();
8. String getText();
9. String getTagName();
10. Point getLocation();
11. Dimension getSize();
12. Rectangle getRect();
13. default @Nullable String getDomProperty(String name) {  
     throw new UnsupportedOperationException("getDomProperty");  
     }
14. default @Nullable String getDomAttribute(String name) {  
     throw new UnsupportedOperationException("getDomAttribute");  
     }
15. default @Nullable String getAriaRole() {  
     throw new UnsupportedOperationException("getAriaRole");  
     }
16. default @Nullable String getAccessibleName() { // **aria-label, aria-labelledby, label**  
     throw new UnsupportedOperationException("getAccessibleName");  
     }
17. List<WebElement> findElements(By by);
18. WebElement findElement(By by);
19. default SearchContext getShadowRoot() {  
     throw new UnsupportedOperationException("getShadowRoot");  
     }
20. String getCssValue(String propertyName);

Locators:

A screenshot of a search engine

Description automatically generated



In Selenium WebDriver for Java, the driver.switchTo().activeElement() method is used to switch the focus to the element that currently has focus within the web page. This is particularly useful when dealing with elements that are dynamically focused, such as input fields or buttons.

| **Attribute** | **Implicit Wait** | **Page Load Timeout** | **Script Timeout** |
| --- | --- | --- | --- |
| **Purpose** | Specifies a global wait time for locating elements. WebDriver will retry locating an element within the specified time before throwing an exception. | Specifies the maximum time WebDriver waits for a page to load completely before throwing an exception. | Specifies the maximum time WebDriver waits for asynchronous JavaScript execution (such as **executeAsyncScript()**) to finish before throwing an exception. |
| **Scope** | Applies to all element searches globally during the session. | Applies to page load operations globally during the session. | Specifically applies to asynchronous JavaScript scripts executed via Selenium's **executeAsyncScript()** method. |
| **Exception Thrown** | Throws **NoSuchElementException** if the element is not found within the wait time. | Throws **TimeoutException** if the page does not load fully within the specified time. | Throws **TimeoutException** if the JavaScript execution exceeds the timeout duration. |

In Selenium WebDriver for Java, the default timeouts are as follows:

Implicitwait: 0 sec

* **PageLoadTimeout**: The default timeout is **300 seconds** (5 minutes). This means Selenium will wait up to 300 seconds for a page to load before throwing an error
* **ScriptTimeout**: The default timeout is **30 seconds**. This is the time Selenium will wait for an asynchronous script to finish execution before throwing an error

Default polling timeout : 500 ms