**0. get xpath/ element/ elements**

*//xpath  
public* By getXpath(String locator) {  
 *return* By.xpath(locator);  
}

*//find element  
public* WebElement getElement(WebDriver driver, String locator) {  
 *return* driver.findElement(getXpath(locator));  
}

*//find elements  
public* List<WebElement> getElements(WebDriver driver, String locator) {  
 *return* driver.findElements(getXpath(locator));  
}

**1. refresh page**

*//refresh page  
public void* refreshPage(WebDriver driver) {  
 driver.navigate().refresh();  
}

**2. back to page**

*//back to page  
public void* backToPage(WebDriver driver) {  
 driver.navigate().back();  
}

**3. forward to page**

*//forward to page  
public void* forwardToPage(WebDriver driver) {  
 driver.navigate().forward();  
}

**4. drag and drop element**

*//drag and drop  
public void* dragAndDropElement(WebDriver driver, String locator, String target) {  
 waitForElementVisible(driver, locator);  
 action = *new* Actions(driver);  
 action.dragAndDrop(getElement(driver, locator), getElement(driver, target)).perform();  
}

**5. select element in dropdown list**

*//select element in dropdown by text  
public void* selectElementInDropdownByText(WebDriver driver, String locator, String text) {  
 waitForElementVisible(driver, locator);  
 select = *new* Select(getElement(driver, locator));  
 select.selectByVisibleText(text);  
}

*//select element in dropdown by value  
public void* selectElementInDropdownByValue(WebDriver driver, String locator, String value) {  
 waitForElementVisible(driver, locator);  
 select = *new* Select(getElement(driver, locator));  
 select.selectByValue(value);  
}

*//select element in dropdown by index  
public void* selectElementInDropdownByIndex(WebDriver driver, String locator, *int* index) {  
 waitForElementVisible(driver, locator);  
 select = *new* Select(getElement(driver, locator));  
 select.selectByIndex(index);  
}

**6. switch window by title**

*//switch window by title  
public void* switchWindowByTitle(WebDriver driver, String text) {  
 String currentWindow = driver.getWindowHandle();  
 *for* (String winHandle : driver.getWindowHandles()) {  
 *if* (driver.getTitle().contains(text)) {  
 driver.switchTo().window(currentWindow);  
 } *else* {  
 driver.switchTo().window(winHandle);  
 }  
 *break*;  
 }  
}

**7. get attribute of element**

*//get attribute  
public* String getElementAttribute(WebDriver driver, String locator, String attribute) {  
 waitForElementVisible(driver, locator);  
 *return* getElement(driver, locator).getAttribute(attribute);  
}

**8. check element displayed**

*//element displayed  
public boolean* isElementDisplayed(WebDriver driver, String locator) {  
 *return* getElement(driver, locator).isDisplayed();  
}

**9. check element selected**

*//element selected  
public boolean* isElementSelected(WebDriver driver, String locator) {  
 *return* getElement(driver, locator).isSelected();  
}

**10. right click to element**

*//right click  
public void* rightClickToElement(WebDriver driver, String locator) {  
 waitForElementClickable(driver, locator);  
 action = *new* Actions(driver);  
 action.contextClick(getElement(driver, locator)).perform();  
}

**11. press key**

*//press key  
public void* pressKey(WebDriver driver, Keys key) {  
 action = *new* Actions(driver);  
 action.keyDown(key).perform();  
}

*//release key  
public void* releaseKey(WebDriver driver, Keys key) {  
 action = *new* Actions(driver);  
 action.keyUp(key).perform();  
}

**12. wait all elements visible**

*//wait all elements visible  
public void* waitForAllElementsVisible(WebDriver driver, String locator) {  
 explicitWait = *new* WebDriverWait(driver, Duration.ofSeconds(GlobalConstant.SHORT\_TIMEOUT));  
 explicitWait.until(ExpectedConditions.visibilityOfAllElements(getElements(driver, locator)));  
}

**13. wait all elements invisible**

*//wait all elements invisible  
public void* waitForAllElementsInvisible(WebDriver driver, String locator) {  
 explicitWait = *new* WebDriverWait(driver, Duration.ofSeconds(GlobalConstant.SHORT\_TIMEOUT));  
 explicitWait.until(ExpectedConditions.invisibilityOfAllElements(getElements(driver, locator)));  
}

**14. accept alert/ dismiss alert**

*//accept alert  
public void* acceptAlert(WebDriver driver) {  
 driver.switchTo().alert().accept();  
}

*//dismiss alert  
public void* dismissAlert(WebDriver driver) {  
 driver.switchTo().alert().dismiss();  
}

**15. send key in alert/ get text in alert**

*//send key in alert  
public void* sendKeyInAlert(WebDriver driver, String value) {  
 driver.switchTo().alert().sendKeys(value);  
}

*//get text in alert  
public* String getTextInAlert(WebDriver driver) {  
 *return* driver.switchTo().alert().getText();  
}

**16. click to element**

*//click  
public void* clickToElement(WebDriver driver, String locator) {  
 waitForElementClickable(driver, locator);  
 getElement(driver, locator).click();  
}

**17. wait element clickable**

*//wait element clickable  
public void* waitForElementClickable(WebDriver driver, String locator) {  
 explicitWait = *new* WebDriverWait(driver, Duration.ofSeconds(GlobalConstant.SHORT\_TIMEOUT));  
 explicitWait.until(ExpectedConditions.elementToBeClickable(getXpath(locator)));  
}

**18. wait element visible/ invisible**

*//wait element visible  
public void* waitForElementVisible(WebDriver driver, String locator) {  
 explicitWait = *new* WebDriverWait(driver, Duration.ofSeconds(GlobalConstant.SHORT\_TIMEOUT));  
 explicitWait.until(ExpectedConditions.visibilityOfElementLocated(getXpath(locator)));  
}

*//wait element invisible  
public void* waitForElementInvisible(WebDriver driver, String locator) {  
 explicitWait = *new* WebDriverWait(driver, Duration.ofSeconds(GlobalConstant.SHORT\_TIMEOUT));  
 explicitWait.until(ExpectedConditions.invisibilityOfElementLocated(getXpath(locator)));  
}

**19. get text in element**

*//get text  
public* String getElementText(WebDriver driver, String locator) {  
 waitForElementVisible(driver, locator);  
 *return* getElement(driver, locator).getText().trim();  
}

**20. compare message**

*//compare message  
public boolean* compareMessage(WebDriver driver, String locator, String message) {  
 *return* getElementText(driver, locator).equals(message);  
}