# TestNG

**TestNG xml**

- Control the execution flow
- Define parameters
- Include and exclude tests
- Parallel execution

**Skip Tests**

- @Test(enable=false)
- Throw new SkipException(“reason”)

  In TestNG, you can use the skipException attribute to specify a list of exceptions that, when encountered during test method execution, will cause the test method to be skipped rather than marked as a failure. This feature can be useful in certain testing scenarios where you want to gracefully handle specific exceptions without failing the entire test.

**Prioritize (Ordered execution)**

- @Test(priority = 3)

**Assertion**

- Hard assertion
  Hard assertions are the traditional assertions that stop test execution immediately when an assertion fails. They use the Assert class provided by TestNG.

- Soft assertion
  Soft assertions are assertions that continue test execution even after a failure. TestNG does not stop the test immediately; it collects all assertion failures and reports them at the end of the test.

- Assertion method
    - AssertEqual
    - AssertTrue
    - AssertFalse

**DependsOnMethod**

This feature allows you to control the order of test method execution, ensuring that one test method runs only after another has successfully completed.

**ITestListener**

In TestNG, an ITestListener is an interface that allows you to implement custom behavior or actions during different phases of the test execution lifecycle. It provides callback methods that get invoked before and after test methods, test classes, suites, and other test-related events. TestNG's ITestListener interface is a powerful tool for customizing your test execution and reporting.

**TestNGReport**

## Xpath //HTMLElement[@attribute = ‘value’] (resouce => xpath.txt)

- and
- text()
- parent/child
- Hide parent->child via //
- Wildcard(*)
- Calling parent by /.. And why it’s important
- contains() with text() or @attribute
- text() with normalize-space
- starts-with/ ends-with()
- self/attribute/child/descendant/descendant-or-self/following-sibling/following
- parent/ancestor/ancestor-or-self/preceding-sibling

## CSS path HTMLElement[attribute= ‘value’] (resouce => xpath.txt)

- Whild card ^, $, *
- CSS combinator -> descendant(“ “), child(“>”), adjacent sibling(“+”), general sibling(“~”)

## Selenium

- Configuration

  Update pom.xml dependency (30th of September, 2023)

```
<dependency>
   <groupId>org.seleniumhq.selenium</groupId>
   <artifactId>selenium-java</artifactId>
   <version>4.13.0</version>
</dependency>
```

Download and save chromedriver.exe file somewhere, and then set it to System.property as follows. In Selenium 4 and later versions, the requirement to set the system property for the WebDriver executable (e.g., ChromeDriver) using System.setProperty is not mandatory but is still recommended.

```
@Test
public void searchOnEbay(){
   System.setProperty("webdriver.chrome.driver",
"C:\\UCSC\\Day_04\\chromeexe\\chromedriver-win64\\chromedriver.exe");
```

```
   WebDriver driver = new ChromeDriver();
   driver.get("https://www.ebay.com/");

driver.findElement(By.xpath("//input[@id=\"gh-ac\"]")).sendKeys("mobile
phones");
   driver.findElement(By.xpath("//input[@id=\"gh-btn\"]")).click();
}
```

- First program

```
public class MyFirstSeleniumProgram {

   @Test
   public void searchOnEbay(){
       System.setProperty("webdriver.chrome.driver",
"C:\\UCSC\\Day_04\\chromeexe\\chromedriver-win64\\chromedriver.exe");
       WebDriver driver = new ChromeDriver();
       driver.get("https://www.ebay.com/");

driver.findElement(By.xpath("//input[@id=\"gh-ac\"]")).sendKeys("mobile
phones");
       driver.findElement(By.xpath("//input[@id=\"gh-btn\"]")).click();
   }
}
```