Qualité Logicielle

TD/TP N° 9 – Automatisation de test fonctionnel avec Selenium

IDE et Selenium webdriver

Objectif : Selenium est un framework de test informatique développé en Java. Il permet d'interagir avec des navigateurs web de même que le ferait un utilisateur de l'application. Il entre ainsi dans la catégorie des outils de test dynamique (à l'inverse des tests statiques qui ne nécessitent pas l'exécution du logiciel) facilitant le test fonctionnel.
Dans ce TP, Il est associé à :

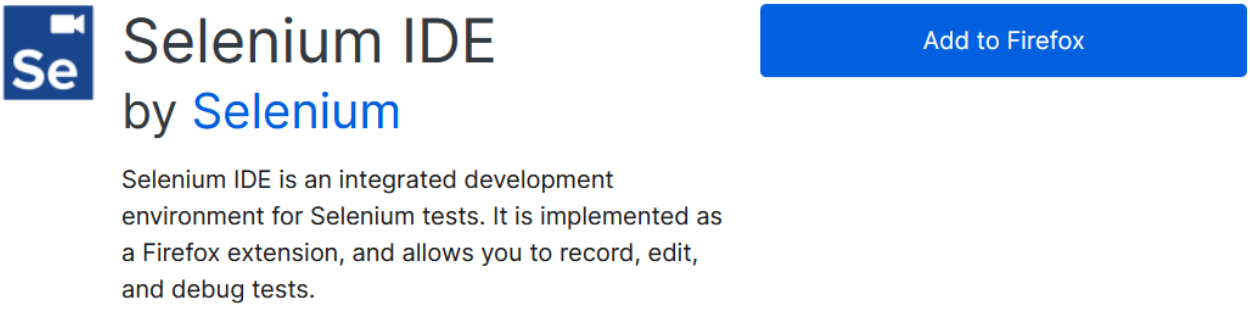
- Selenium IDE, extension Google Chrome, pour l'utiliser ;
- Selenium WebDriver. Il permet d'écrire des tests automatisés en différents langages (PHP, Python, Ruby, .NET, Perl et Java).

Nous allons donc voir dans cet atelier comment installer et utiliser Selenium IDE pour créer des tests fonctionnels reproductibles.

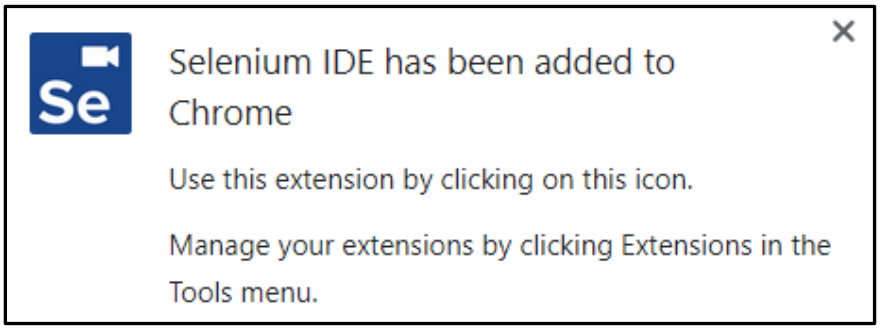
Partie 1 : Premier projet de test fonctionnel avec Selenium IDE

1. Télécharger chrome la dernière version de firefox:
   https://www.firefox.com/fr/?utm_campaign=SET_DEFAULT_BROWSER

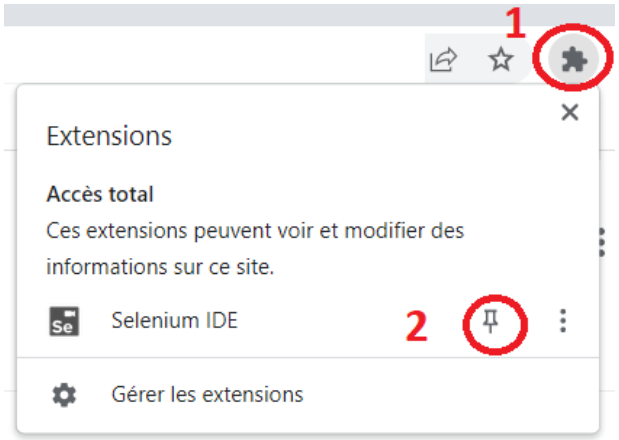
2. Télécharger la dernière version de Selenium IDE : https://selenium.dev/selenium-ide/, choisir l'extension de Chrome et installer la version.



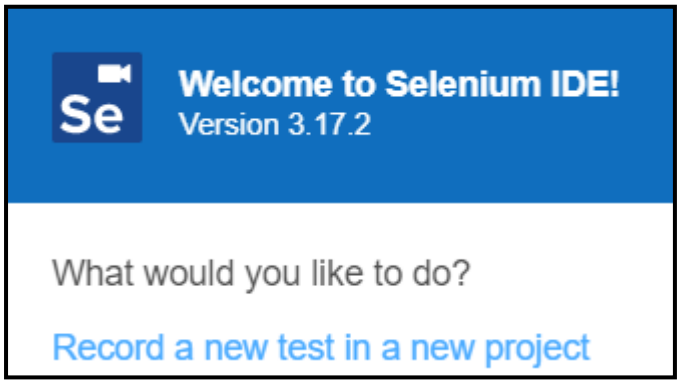
Vérifier qu'une extension dans le navigateur vient de s'ajouter :



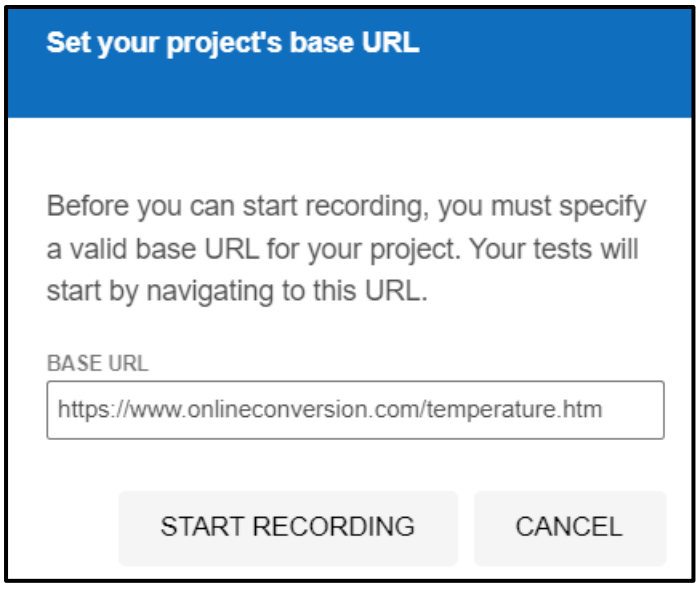
3. Cochez le symbole correspondant au chiffre 2 sur l'image ci-dessous, Selenium IDE sera activé (il devient bleu) et sera affiché en haut à droite de la barre d'adresses de votre navigateur web google chrome.



4. Effectuer un premier enregistrement en suivant les étapes suivantes :
   - Ouvrir le navigateur Google Chrome et cliquer sur l'icône Se : 

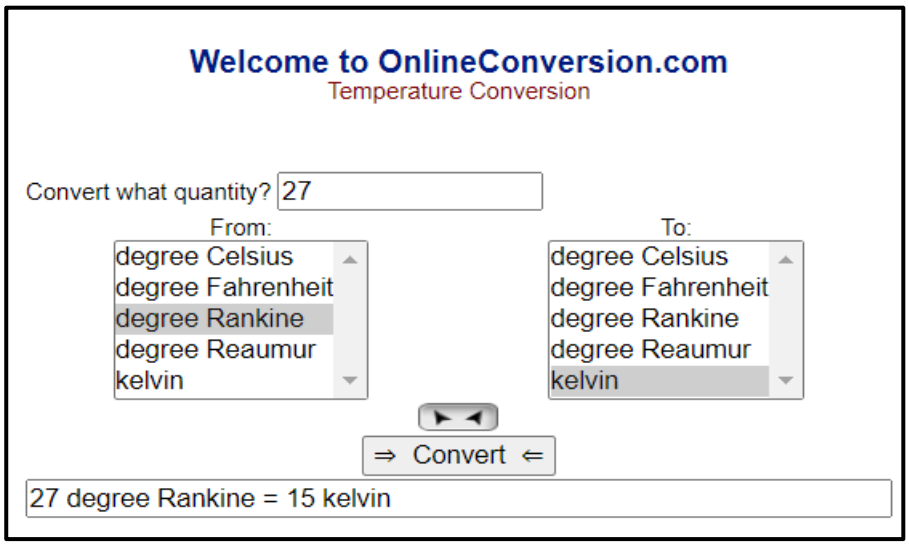


   - Créer un projet et le nommer « premierExempleSeleniumIDE »
   - Dans base url, entrer :
     https://www.calculator.net/bmi-calculator.html ou bien
   - https://www.onlineconversion.com/temperature.htm



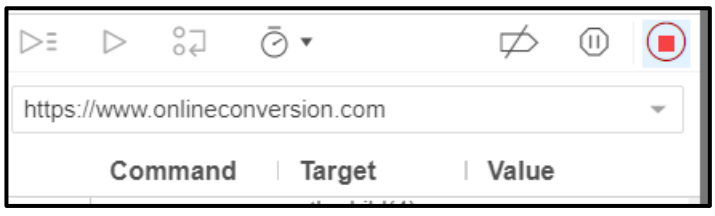
- Puis cliquer sur Start Recording.
- L'enregistrement est démarré :

- Saisissez une valeur de température et choisissez le type source et le type destination de conversion que vous voulez faire. Répétez cette action pour quelques valeurs de températures et des types de conversions.

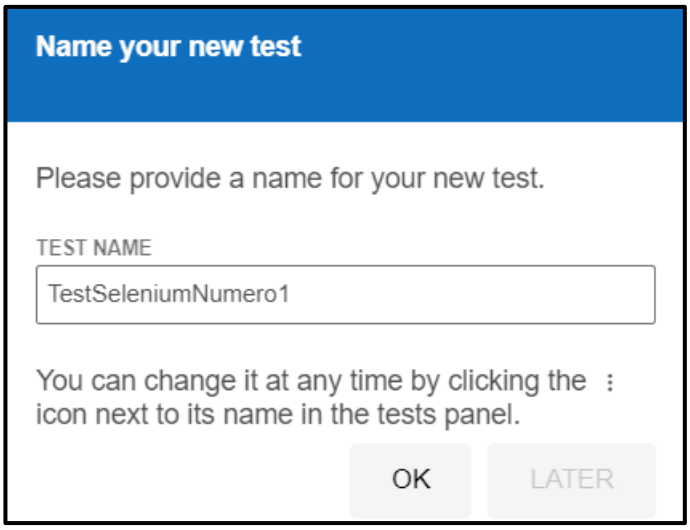


- Cliquer sur l'icone de Se :

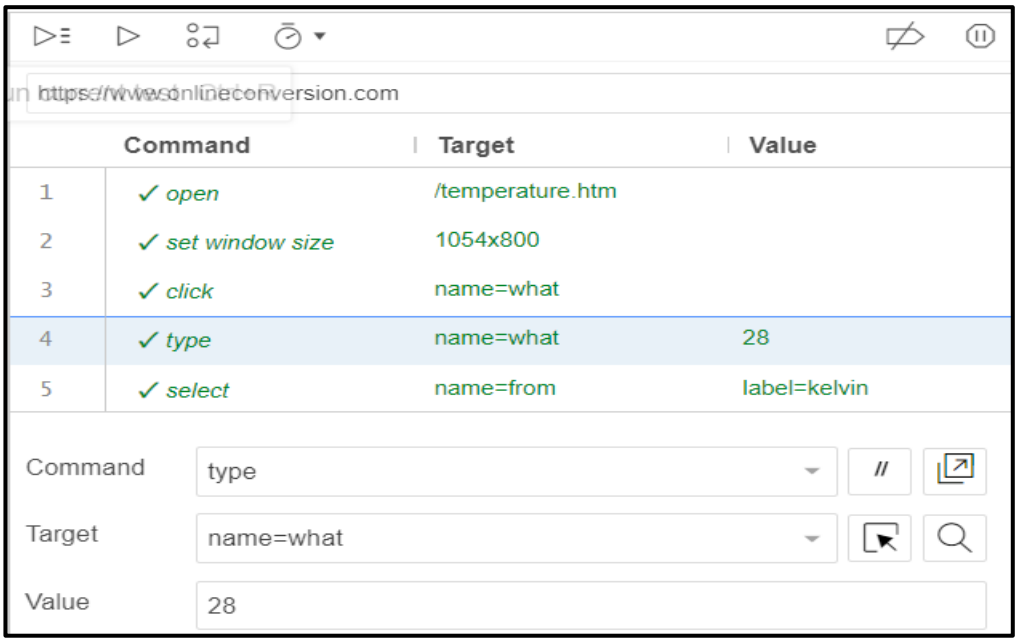
- Cliquer sur sur Ctrl+U ou sur le bouton rouge pour commencer l'enregistrement « stop-recording ».

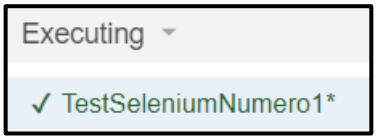


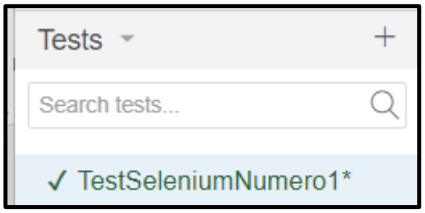
- Nommez votre test : TestSeleniumNumero1



- Lancer le test : 
- Réglez la vitesse d'exécution de test
- Editer la valeur (28 dans ce texemple), remplacez-la par une autre et relancer le test.

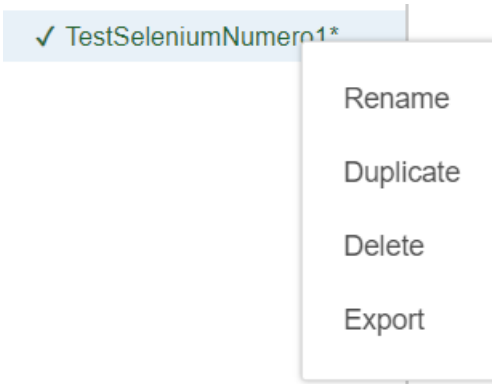


- La liste des commandes est disponible à : https://www.selenium.dev/selenium-ide/docs/en/api/commands
- Changez : 

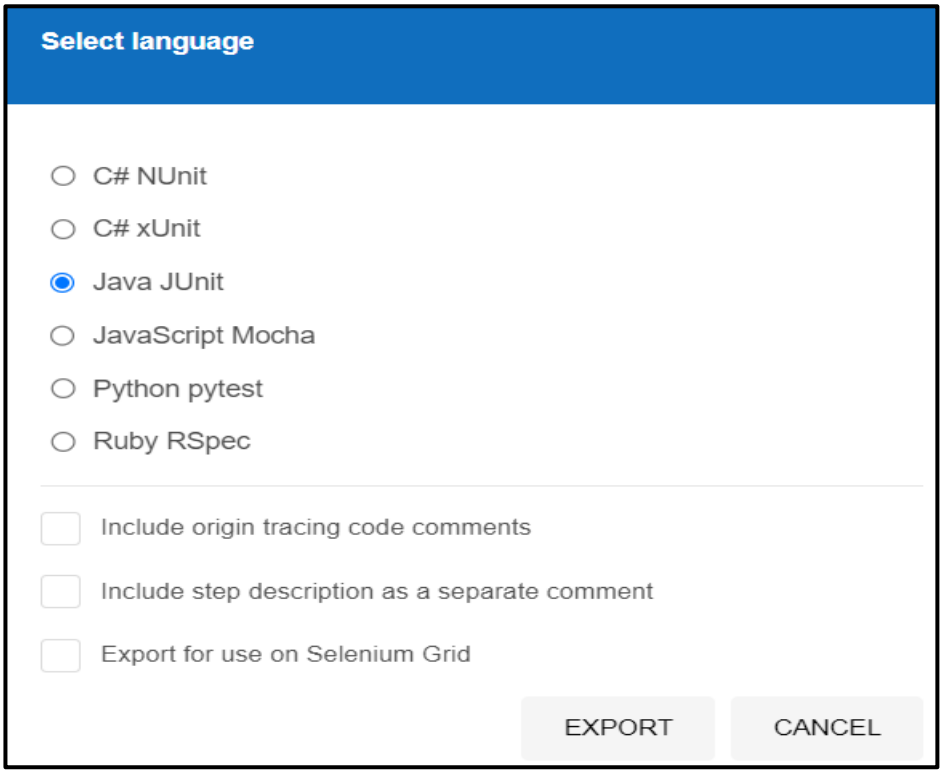
par 

- Enregistrez votre projet sous le nom : « projetTp1SeleniumIDE.side » avec le bouton  en haut à droite ou (ctrl+s).
- Quittez Selenium IDE et ouvrir de nouveau votre projet enregistré :

- Bouton droit sur le nom du test, puis cliquez sur Export :



➔Sélectionner Java-JUNIT.



- Nommez le fichier java : TestSeleniumNumero1Test.java
  Voici un extrait du fichier obtenu :

```
@Test
 public void testSeleniumNumero1() {
  driver.get("https://www.onlineconversion.com/temperature.htm");
  driver.manage().window().setSize(new Dimension(1054, 800));
  driver.findElement(By.name("what")).click();
  driver.findElement(By.name("what")).sendKeys("28");
  { WebElement dropdown = driver.findElement(By.name("from"));
    dropdown.findElement(By.xpath("//option[. = 'kelvin']")).click();   }
  driver.findElement(By.cssSelector("td:nth-child(1) option:nth-child(5)")).click();
  { WebElement dropdown = driver.findElement(By.name("to"));
    dropdown.findElement(By.xpath("//option[. = 'degree Fahrenheit']")).click();   }
  driver.findElement(By.cssSelector("td:nth-child(2) option:nth-child(2)")).click();
  driver.findElement(By.name("Go")).click();
  driver.switchTo().frame(3);
  driver.findElement(By.cssSelector("#cbb > svg")).click();  }
```

Partie 2 : Premier projet de test Selenium Webdriver

1. Créer un projet java maven intitulé seleniumproject
2. Ajouter les dépendances :

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>Tps_Tests_fonctionnels_Selenium</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
   <dependencies>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>RELEASE</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-chrome-driver</artifactId>
        <version>4.16.1</version>
        <scope>test</scope>
    </dependency>
    <!-- pour import org.openqa.selenium.support.ui.Select; dans TP
selectTest   -->
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>4.18.1</version>
    </dependency>
</dependencies>
</project>
```

3. **Exécutez les tests suivants :**

```
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.ValueSource;
import org.openqa.selenium.By;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
public class conversionTemperatureTestParametree {
private static WebDriver driver;
    @BeforeAll
    public static void setUpAll() {    driver = new ChromeDriver();}
    @AfterAll
    public static void tearDown() {       driver.quit();    }
  @ParameterizedTest
  @ValueSource(strings = {"33","41","19","25","-14","35845"})
  public void premiertest(String temp) {
    driver.get("http://www.onlineconversion.com/temperature.htm");
    driver.manage().window().setSize(new Dimension(1053, 799));
    { WebElement element = driver.findElement(By.name("Go"));
      Actions builder = new Actions(driver);
      builder.moveToElement(element).perform();}
    {
      WebElement element = driver.findElement(By.tagName("body"));
      Actions builder = new Actions(driver);
      builder.moveToElement(element, 0, 0).perform();
    }
    {
      WebElement dropdown = driver.findElement(By.name("from"));
      dropdown.findElement(By.xpath("//option[. = 'degree
Fahrenheit']")).click();    }
    driver.findElement(By.cssSelector("td:nth-child(1) option:nth-
child(2)")).click();
    driver.findElement(By.cssSelector("tr:nth-child(1) > td")).click();
    driver.findElement(By.name("what")).sendKeys(temp);
    {
      WebElement dropdown = driver.findElement(By.name("to"));
      dropdown.findElement(By.xpath("//option[. = 'degree
Rankine']")).click();    }
    driver.findElement(By.cssSelector("td:nth-child(2) option:nth-
child(3)")).click();
    driver.findElement(By.name("Go")).click();
    {
      WebElement dropdown = driver.findElement(By.name("from"));
      dropdown.findElement(By.xpath("//option[. = 'degree
ankine']")).click();
    }
    driver.findElement(By.cssSelector("td:nth-child(1) option:nth-
child(3)")).click();
    {
      WebElement dropdown = driver.findElement(By.name("to"));
      dropdown.findElement(By.xpath("//option[. = 'degree
Fahrenheit']")).click();
    }
    driver.findElement(By.cssSelector("td:nth-child(2) option:nth-
child(2)")).click(); }
}
```

4. **Exécutez les tests suivants :**

```
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import static org.junit.jupiter.api.Assertions.assertEquals;
public class exempleTest_assertEqualsFireFox {
    private static WebDriver driver;
    @BeforeAll
    public static void setUpAll() {
    driver=new FirefoxDriver();     }
    @AfterAll
    public static void tearDown() {driver.quit();}
    @Test
    public void fonctionTest1() {
    String baseUrl = "http://google.com";
    String expectedTitle = "Googlefr";
    String actualTitle;
    driver.get(baseUrl);
    actualTitle = driver.getTitle();
    assertEquals(actualTitle, expectedTitle);
    driver.close();
    }
}
```

5. **Exécutez les tests suivants :**

```
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class facebookSeleniumFindElementBy_Test {
    private WebDriver driver;
      @BeforeEach
      public void setUp() {driver = new ChromeDriver();  }
      @AfterEach
      public void tearDown() {   driver.quit();  }
      @Test
      public void premiertest() {
        String baseUrl = "http://www.facebook.com" ;
        driver.get(baseUrl);String texte;
        texte = driver.findElement(By.partialLinkText("nect")).getText();
        System.out.println("\n===1==="+texte);
        texte = driver.findElement(By.tagName("div")).getText();
        System.out.println("\n===2==="+texte);
        texte = driver.findElement(By.linkText("Se
connecter")).getTagName();
        System.out.println("\n===3==="+texte);
        texte= driver.findElement(By.id("email")).getTagName();
        System.out.println("\n===4==="+texte);
        texte=
driver.findElement(By.cssSelector("input#email")).getTagName();
        System.out.println("\n===5==="+texte);}  }
```

6. **Exécutez les tests suivants :**

```
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import static org.junit.jupiter.api.Assertions.assertEquals;

public class facebookSeleniumTest {
    private WebDriver driver;
      @BeforeEach
      public void setUp() {driver = new ChromeDriver();  }
      @AfterEach
      public void tearDown() {   driver.quit();  }
      @Test
      public void premiertest() {
        String baseUrl = "http://www.facebook.com" ;
        driver.get(baseUrl);
        //String tagName =
driver.findElement(By.id("email")).getTagName();
        String tagName =
driver.findElement(By.cssSelector("input#email")).getTagName();
        assertEquals("input",tagName);}  }
```

7. **Exécutez les tests suivants :**

```
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import static org.junit.jupiter.api.Assertions.assertEquals;

public class pageGoogleTest {
    private static WebDriver driver;
    @BeforeAll
    public static void setUpAll() {
        driver = new ChromeDriver();}
    @AfterAll
    public static void tearDown() {
        driver.quit(); }
    @Test
    public void fonctionTest1() {
        String baseUrl = "http://google.com";
        String expectedTitle = "Google";
        String actualTitle; driver.get(baseUrl);
        actualTitle = driver.getTitle();
        assertEquals(actualTitle, expectedTitle);
        driver.close();
    }
}
```

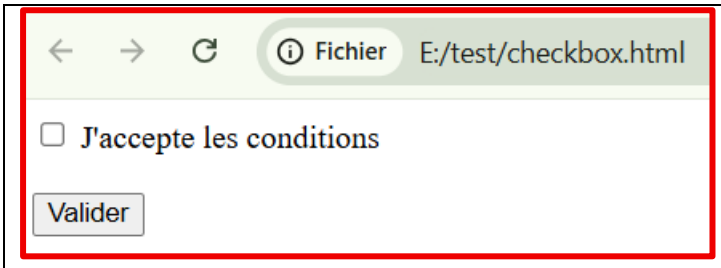
8. ***Faire un test fonctionnel pour la page web suivante et le code html associé***

<!—contenu de connexion_login_password.html-->

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Login</title>
</head>
<body>
<h2>Connexion</h2>
<form>
  <label>Username:</label>
  <input type="text" id="username"><br><br>
  <label>Password:</label>
  <input type="password" id="password"><br><br>
  <button type="button" id="loginBtn"
onclick="login()">Login</button>
</form>
<p id="message"></p>
<script>
  function login() {
    let user = document.getElementById("username").value;
    let pass = document.getElementById("password").value;

    if (user === "admin" && pass === "1234") {
```

```
            document.getElementById("message").innerText =
        "Connexion réussie";
            } else {
            document.getElementById("message").innerText = "Échec
        de connexion";
            }
          }
        </script>

        </body>
</html>
```

9. ***Faire un test fonctionnel pour la page web suivante et le code html associé***

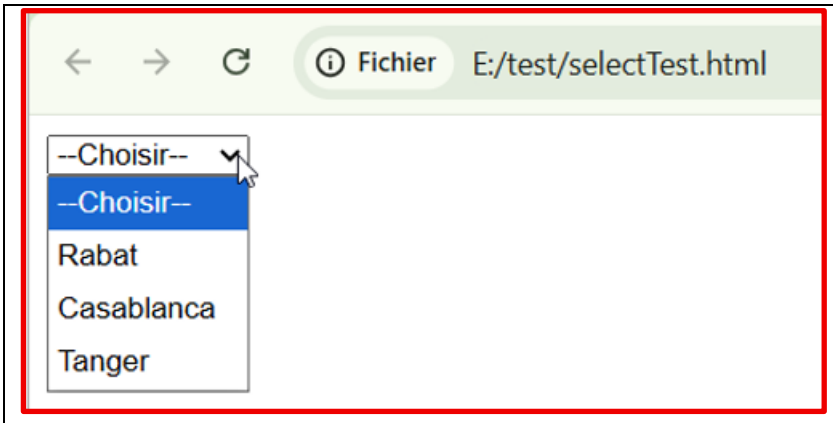


<!—contenu de checkbox.html-->

```
        <!DOCTYPE html>
        <html>
        <head>
          <title>Checkbox</title>
        </head>
        <body>

        <input type="checkbox" id="agree"> J'accepte les conditions
        <br><br>
        <button onclick="verifier()">Valider</button>

        <p id="result"></p>

        <script>
```

```
        function verifier() {
            let checked = document.getElementById("agree").checked;
            document.getElementById("result").innerText =
                checked ? "Accepté" : "Refusé";
        }
    </script>

    </body>
</html>
```

10. ***Faire un test fonctionnel pour la page web suivante et le code html associé***



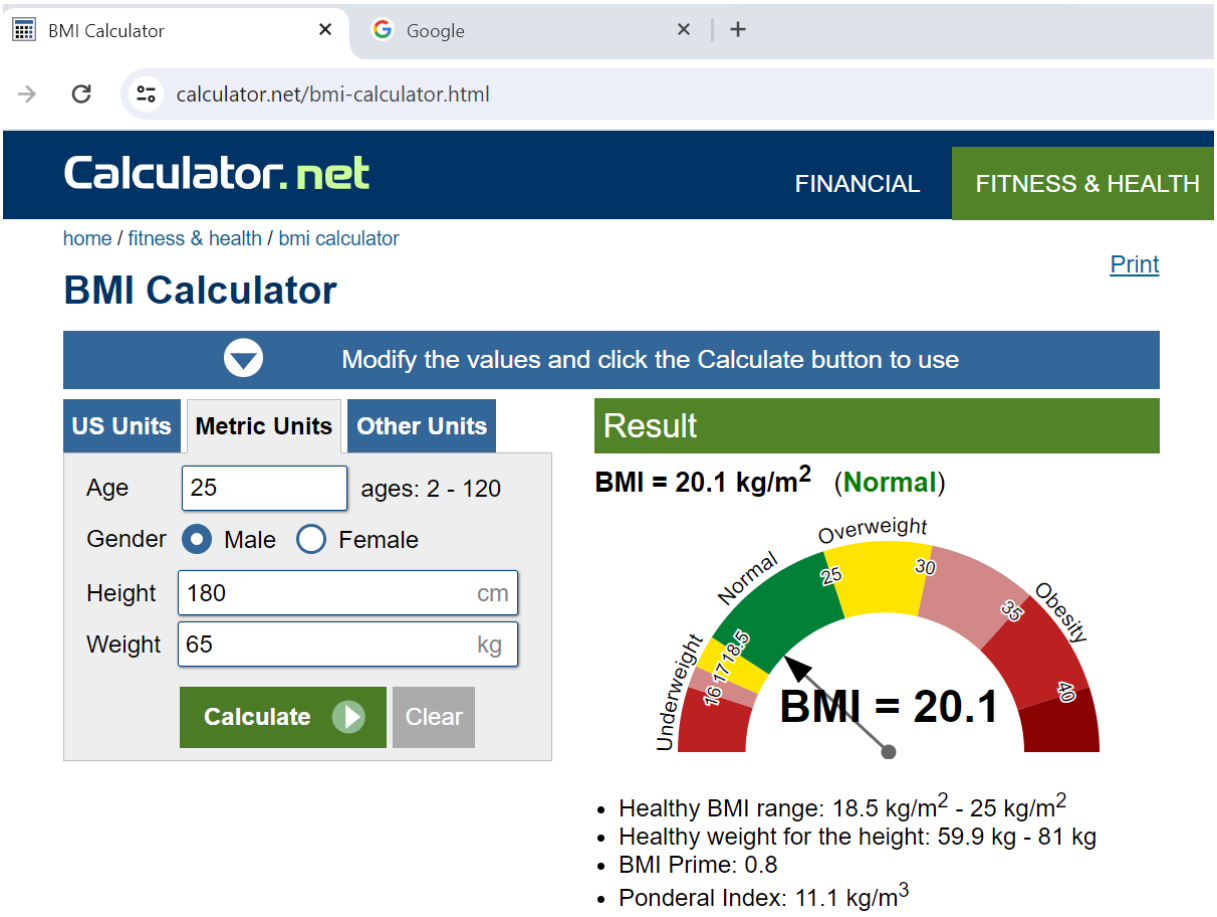
<!—contenu de selectTest.html-->

```
    <!DOCTYPE html>
    <html>
    <body>
    <select id="ville">
        <option value="">--Choisir--</option>
        <option value="Rabat">Rabat</option>
        <option value="Casablanca">Casablanca</option>
        <option value="Tanger">Tanger</option>
    </select>
    <p id="choix"></p>
    <script>
        document.getElementById("ville").onchange = function() {
            document.getElementById("choix").innerText =
```

```
                this.value;
            }
        </script>
        </body>
</html>
```

11. ***Un autre exemple… …***

Objectif : nous voulons tester les fonctionnalités de l'application web suivant :



Travail à faire

1. Aller le site suivant : https://www.calculator.net/bmi-calculator.html
2. Préparer le test fonctionnel
3. Automatiser le test en utilisant seleniumwebdriver.
    - Utiliser les tests paramétrés (CsvFileSource, valueSource, ….)
    - Utiliser les navigateurs Google Chrome, Edge et Firefox