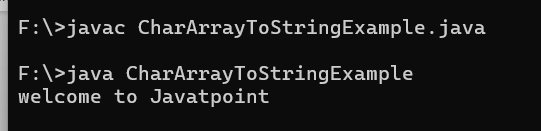
STEPS TO RUN A JAVA FILE USING NOTEPAD

**Save the file with the name**

**CharArrayToStringExample.java**

1. **public** **class** CharArrayToStringExample
2. {
3. **public** **static** **void** main(String args[])
4. {
5. //character array
6. **char**[] ch = {'w', 'e', 'l', 'c', 'o', 'm', 'e', ' ' , 't', 'o', ' ', 'J', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't'};
7. //constructor of the String class that parses char array as a parameter
8. String string = **new** String(ch);
9. //prints the string
10. System.out.println(string);
11. }
12. }



Actions class

**driver.get() :** It's used to go to the particular website , But it doesn't maintain the browser History and cookies so , we can't use forward and backward button , if we click on that , page will not get schedule

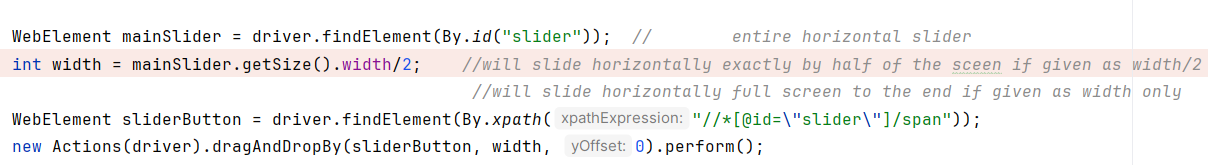
**driver.navigate() :** it's used to go to the particular website , but it maintains the browser history and cookies, so we can use forward and backward button to navigate between the pages during the coding of Testcase

In **Selenium WebDriver**for the closing browser session, there are two WebDriver commands driver.quit() and driver.close(). And both have different objective **close()** command closes the browser window which is currently active and **quit()** command closes all the browser windows and terminate a WebDriver session.

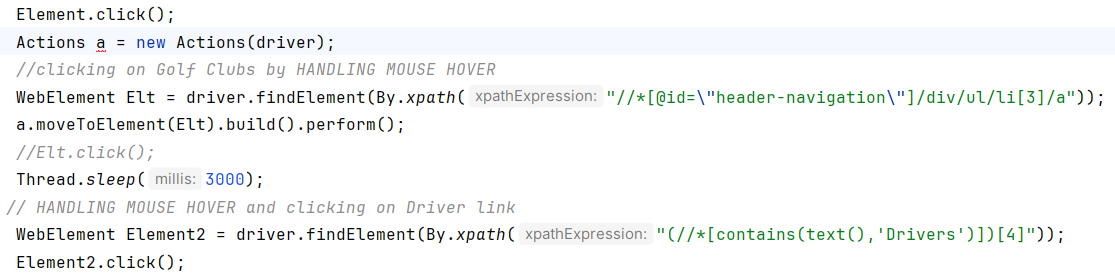
========================================================================

Horizontal Slider

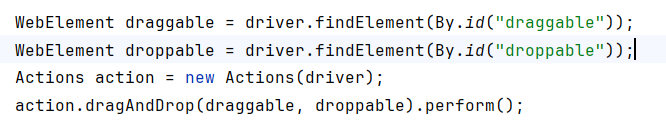
URL = driver.get("https://jqueryui.com/resources/demos/slider/default.html");



MOUSE HOVER



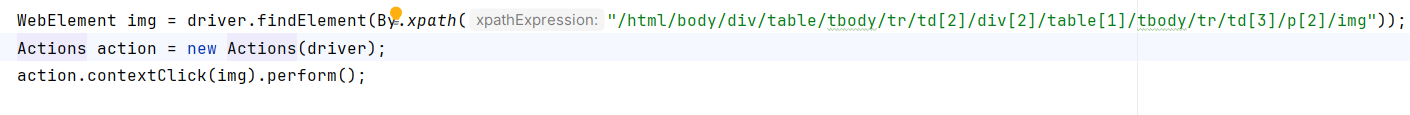
DRAG AND DROP

url = [https://jqueryui.com/resources/demos/droppable/default.html](https://jqueryui.com/resources/demos/droppable/default.html)

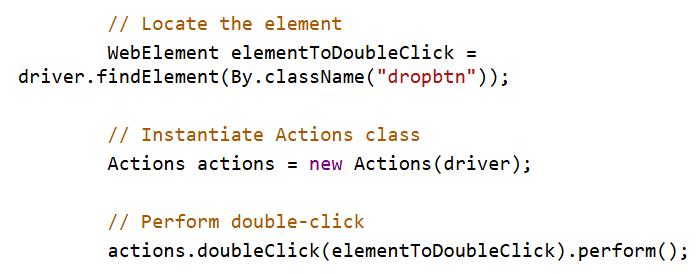
RESIZE



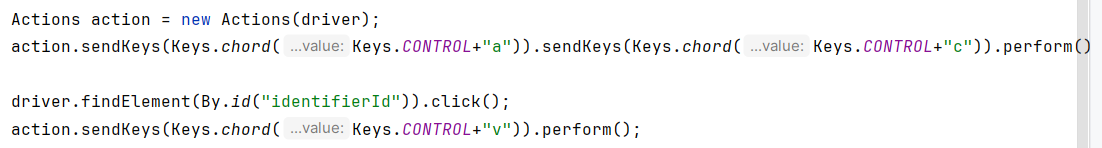
RIGHT CLICK



DOUBLE CLICK

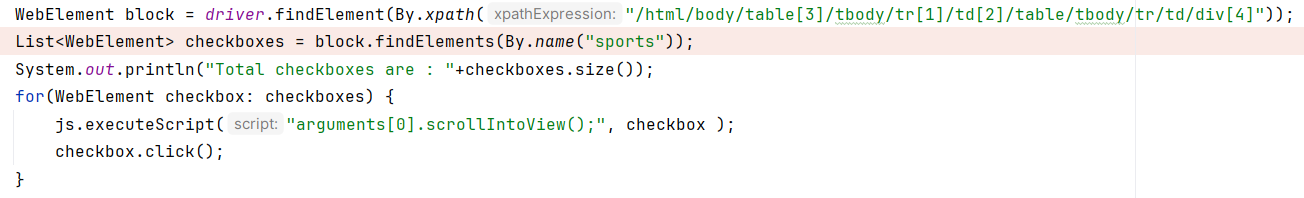


SELECT, COPY, and PASTE

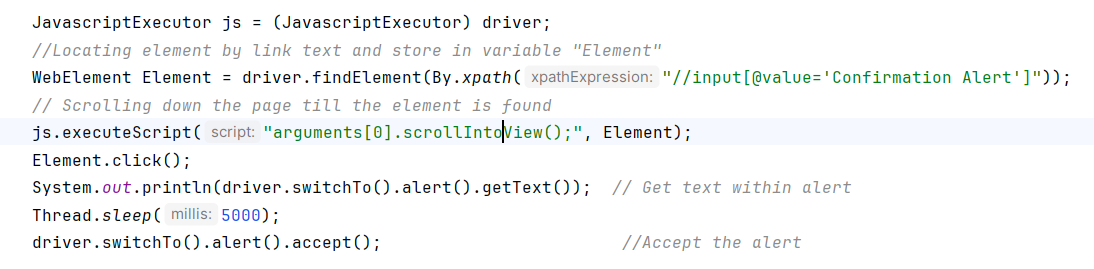


NOT INTERACTABLE CHECKBOXES, need to use javascriptexecutor

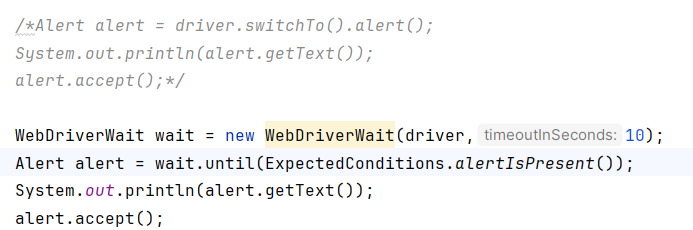
*driver*.get("http://www.tizag.com/htmlT/htmlcheckboxes.php");



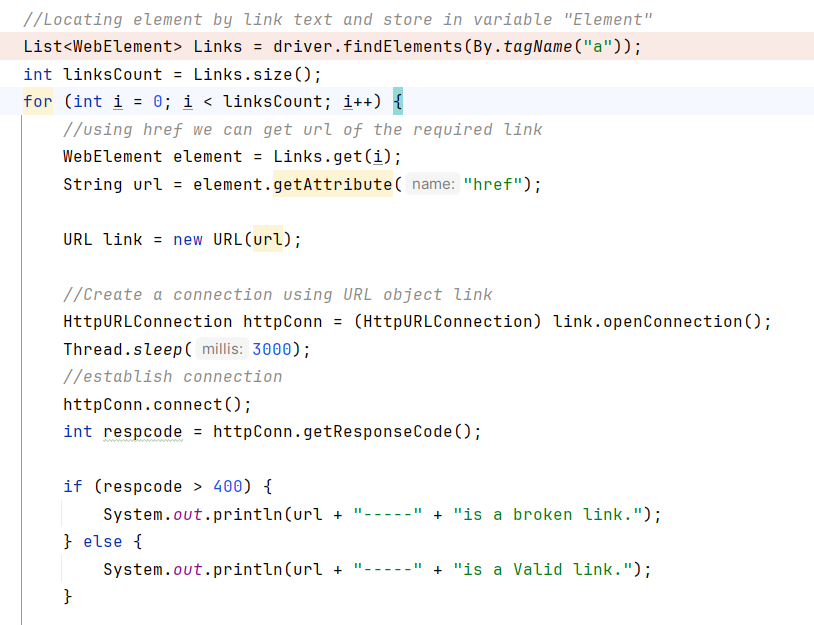
ScrollIntoView



ALERTS

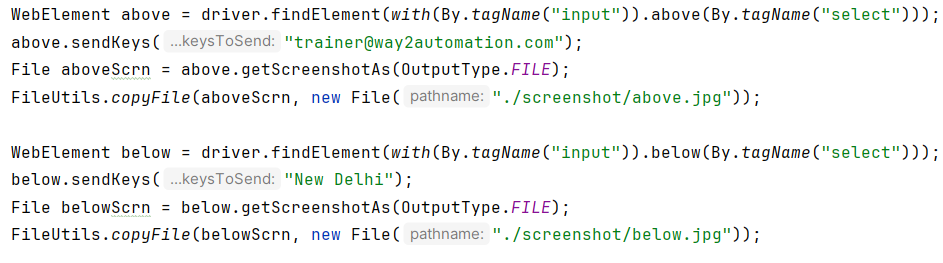


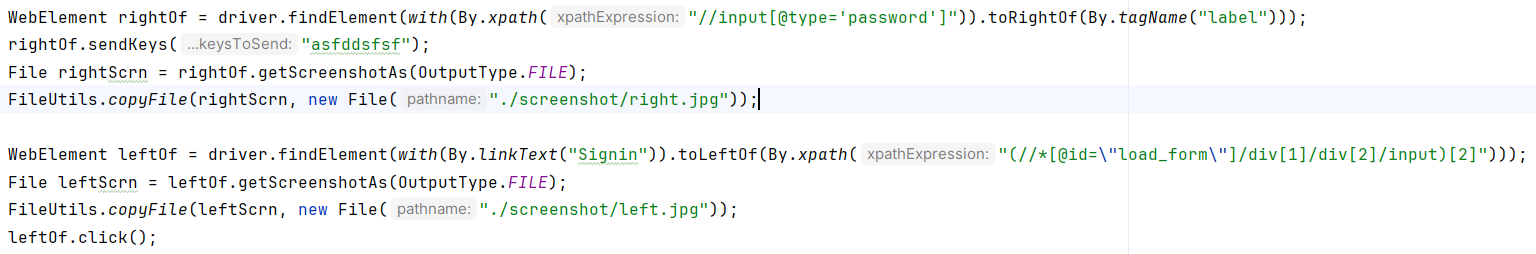
BROKEN LINKS



RELATIVE LOCATORS :

driver.get("https://www.way2automation.com/way2auto\_jquery/index.php");





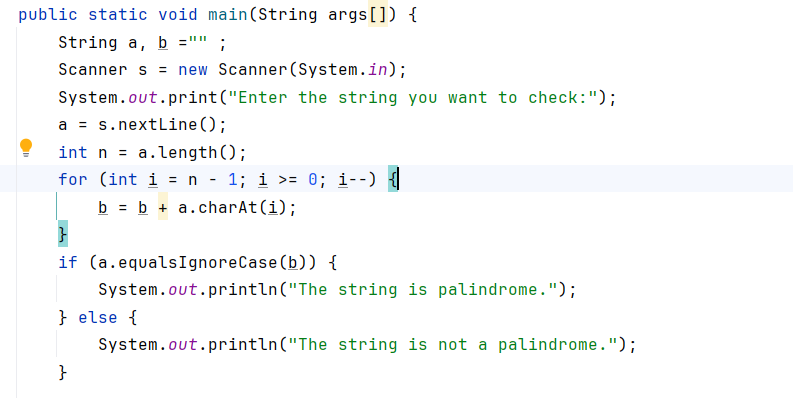


WEBTABLE

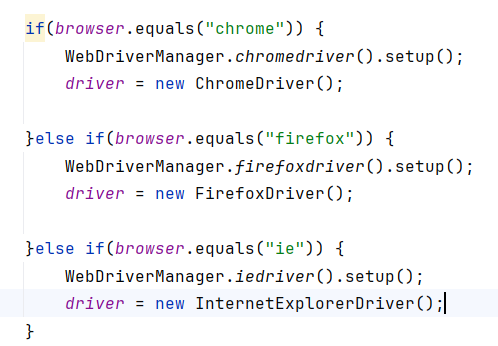
driver.get("https://money.rediff.com/gainers/bsc/daily/groupa");



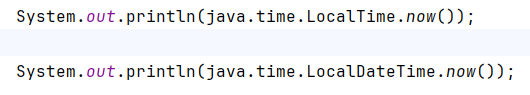
PALLINDROME



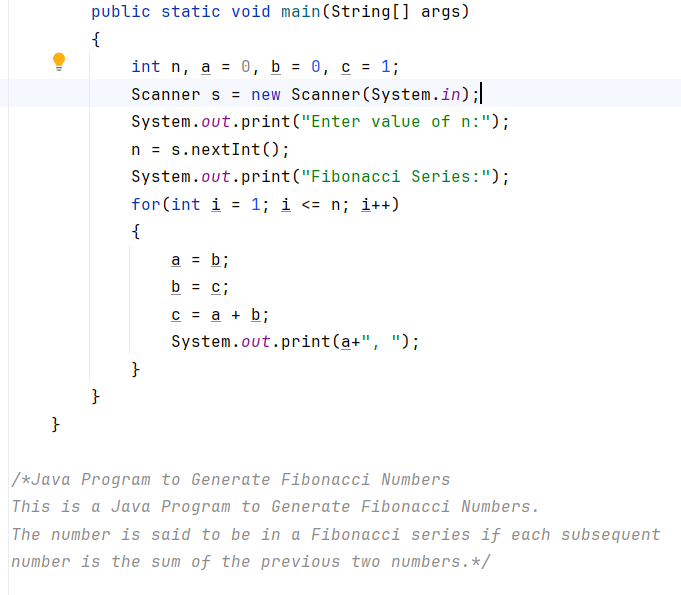
WEBDRIVER MANAGER



DATE AND TIME



FIBONACCI SERIES

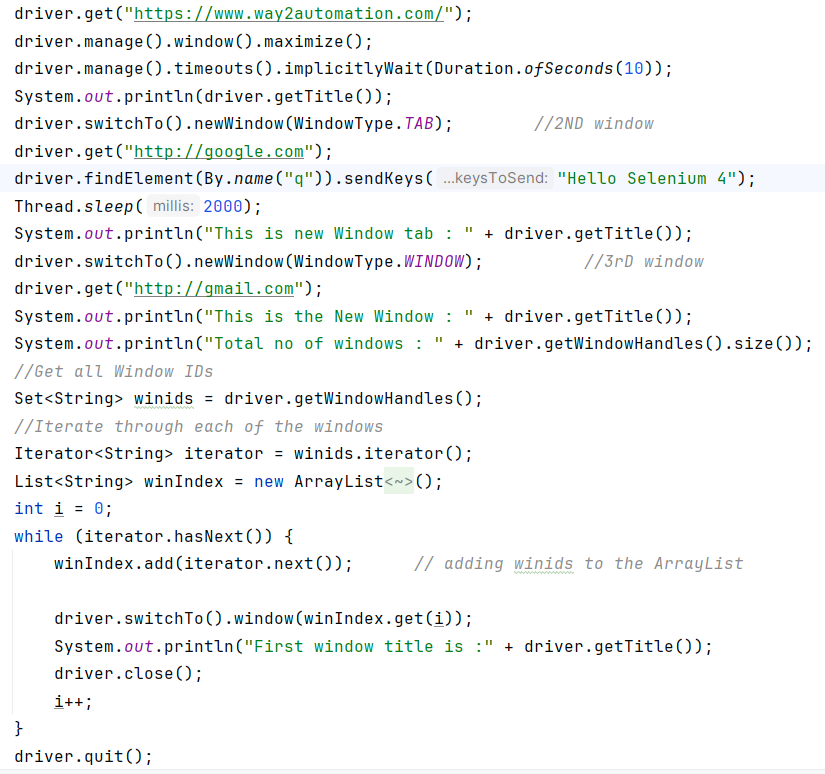


SCREEN COORDINATES

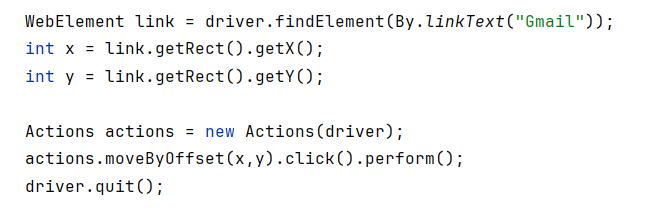


SWITCH TO WINDOWS

driver.get("https://www.way2automation.com/");



CLICKING ON AN OBJECT USING X, Y COORDINATES



JDBC

Selenium Webdriver is limited to[Testing](https://www.guru99.com/software-testing.html)your applications using Browser. To use Selenium Webdriver for Database Verification you need to use the JDBC ("Java Database Connectivity").

JDBC (Java Database Connectivity) is a[SQL](https://www.guru99.com/sql.html)level API that allows you to execute SQL statements. It is responsible for the connectivity between the[Java](https://www.guru99.com/java-tutorial.html)Programming language and a wide range of databases. The JDBC API provides the following classes and interfaces

* Driver Manager
* Driver
* Connection
* Statement
* ResultSet
* SQLException
* Package htmldriver;
* import java.sql.Connection;
* import java.sql.Statement;
* import java.sql.ResultSet;
* import java.sql.DriverManager;
* import java.sql.SQLException;
* public class SQLConnector {
* public static void main(String[] args) throws ClassNotFoundException, SQLException {
* //Connection URL Syntax: "jdbc:mysql://ipaddress:portnumber/db\_name"
* String dbUrl = "jdbc:mysql://localhost:3036/emp";
* //Database Username
* String username = "root";
* //Database Password
* String password = "guru99";
* //Query to Execute
* String query = "select \* from employee;";
* //Load mysql jdbc driver
* Class.forName("com.mysql.jdbc.Driver");
* //Create Connection to DB
* Connection con = DriverManager.getConnection(dbUrl,username,password);
* //Create Statement Object
* Statement stmt = con.createStatement();
* // Execute the SQL Query. Store results in ResultSet
* ResultSet rs= stmt.executeQuery(query);
* // While Loop to iterate through all data and print results
* while (rs.next()){
* String myName = rs.getString(1);
* String myAge = rs.getString(2);
* System. out.println(myName+" "+myAge);
* }
* // closing DB Connection
* con.close();
* }
* }

DATAPROVIDER

public class getExcelData {  
  
 @DataProvider(name= "getXLData")  
 public Object[][] getXLData() throws IOException{  
   
 String filepath= System.*getProperty*("user.dir")+"\\src\\test\\java\\com\\Pat\\TestData\\LoginData.xlsx";  
   
 *//FileInputStream fis= new FileInputStream("D:\\MyWorkspace\\pat\_Auto\\src\\test\\java\\com\\Pat\\TestData\\LoginData.xlsx");* FileInputStream fis= new FileInputStream(filepath);  
   
 XSSFWorkbook wb= new XSSFWorkbook(fis);  
 XSSFSheet ws=wb.getSheet("Sheet1");  
 int totalRows= ws.getLastRowNum();  
 int totalColumns= ws.getRow(0).getPhysicalNumberOfCells();  
   
 Object obj[][]= new Object[totalRows][totalColumns];  
 Hashtable<String,String> table = null;

for(int i=0; i<totalRows; i++) {

table = new Hashtable<String,String>();

for (int j = 0; j < totalColumns; j++) {  
  
 obj[i][j] = ws.getRow(i + 1).getCell(j).toString();  
 *//obj[i][1] = ws.getRow(i + 1).getCell(1).toString();  
 //obj[i][2] = ws.getRow(i + 1).getCell(2).toString();  
 //obj[i][j] = ws.getRow(i + 1).getCell(j).*

*//* System.*out*.println(obj[i][j]);

*//table.put(excel.getCellData(sheetName, colNum, 1), excel.getCellData(sheetName, //colNum, rowNum));*

*table.put( ws.getRow(1).getCell(j).toString, ws.getRow(i + 1).getCell(j).toString();*

*data[*totalRows *- 2][0] = table;*  
 }  
 }  
 return obj;  
 }  
  
}

@DataProvider(name="dp")  
public Object[][] getData(Method m) {  
  
 String sheetName = m.getName();  
 int rows = *excel*.getRowCount(sheetName);  
 int cols = *excel*.getColumnCount(sheetName);  
  
 Object[][] data = new Object[rows - 1][1];  
   
 Hashtable<String,String> table = null;  
  
 for (int rowNum = 2; rowNum <= rows; rowNum++) { *// 2* table = new Hashtable<String,String>();  
   
 for (int colNum = 0; colNum < cols; colNum++) {  
  
 *// data[0][0]* table.put(*excel*.getCellData(sheetName, colNum, 1), *excel*.getCellData(sheetName, colNum, rowNum));  
 data[rowNum - 2][0] = table;  
 }  
  
 }  
  
 return data;  
  
}