**Datepicker\_menu\_slider\_tabs\_tooltip**

**package** com.Datepicker\_menu\_slider\_tabs\_tooltip;

**import** java.util.Set;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.JavascriptExecutor;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** Datepicker\_menu\_slider\_tabs\_tooltip {

**public** **static** **void** main(String args[]) **throws** InterruptedException{

String path = "lib\\chromedriver.exe";

System.*setProperty*("webdriver.chrome.driver", path);

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("http://demoqa.com");

Thread.*sleep*(5000);

JavascriptExecutor js = (JavascriptExecutor) driver;

// This will scroll down the page by 1000 pixel vertical

js.executeScript("window.scrollBy(0,1000)");

Thread.*sleep*(5000);

// This will scroll up the page by 1000 pixel vertical

js.executeScript("window.scrollBy(0,-1000)");

/\*//DatePicker

WebElement datepicker = driver.findElement(By.xpath("//a[contains(text(),'Datepicker')]"));

Thread.sleep(2000);

datepicker.click();

//Default functionality

WebElement click\_on\_datefield = driver.findElement(By.xpath("//\*[@id='datepicker1']"));

Thread.sleep(2000);

click\_on\_datefield.click();

WebElement previous = driver.findElement(By.xpath("//\*[@id='ui-datepicker-div']/div/a[1]/span"));

Thread.sleep(2000);

previous.click();

WebElement next = driver.findElement(By.xpath("//\*[@id='ui-datepicker-div']/div/a[2]/span"));

Thread.sleep(2000);

next.click();

DateFormat dateFormat2 = new SimpleDateFormat("dd");

Date date2 = new Date();

String today = dateFormat2.format(date2);

System.out.println("Today's date: " +today+"");

System.out.println("Today's date: " +date2+"");

List<WebElement> select\_date = driver.findElements(By.xpath("//a[@class='ui-state-default']"));

int i = 29;

String str = null;

for(WebElement date1 : select\_date){

str = Integer.toString(i);

if(date1.getText().equals(str)){

date1.click();

Thread.sleep(3000);

break;

}

}

//Animations

WebElement animations = driver.findElement(By.xpath("//a[contains(text(),'Animations')]"));

animations.click();

Thread.sleep(3000);

WebElement pick\_date = driver.findElement(By.xpath("//\*[@id='datepicker2']"));

pick\_date.click();

Thread.sleep(3000);

List<WebElement> select\_date1 = driver.findElements(By.xpath("//a[@class='ui-state-default']"));

int j = 25;

String str1 = null;

for(WebElement date1 : select\_date1){

str1 = Integer.toString(j);

//System.out.println(date1.getText());

if(date1.getText().equals(str1)){

date1.click();

System.out.println(date1.getText());

Thread.sleep(3000);

break;

}

}

WebElement select\_animations = driver.findElement(By.xpath("//select[@id='anim']"));

Select animations\_dropdown= new Select(select\_animations);

for(int i1 = 0;i1<=9;i1++){

animations\_dropdown.selectByIndex(i1);

Thread.sleep(3000);

}

//Display month & year

WebElement month\_year = driver.findElement(By.xpath("//a[contains(text(),'Display month & year')]"));

month\_year.click();

Thread.sleep(3000);

WebElement date\_txt = driver.findElement(By.xpath("//\*[@id='datepicker3']"));

date\_txt.click();

Thread.sleep(3000);

WebElement select\_month = driver.findElement(By.xpath("//select[@class='ui-datepicker-month']"));

Select month = new Select(select\_month);

month.selectByVisibleText("Dec");

Thread.sleep(3000);

WebElement select\_year = driver.findElement(By.xpath("//select[@class='ui-datepicker-year']"));

Select year = new Select(select\_year);

year.selectByVisibleText("2019");

Thread.sleep(3000);

List<WebElement> select\_date2 = driver.findElements(By.xpath("//a[@class='ui-state-default']"));

int l = 25;

String str2 = null;

for(WebElement date1 : select\_date2){

str2 = Integer.toString(l);

//System.out.println(date1.getText());

if(date1.getText().equals(str2)){

date1.click();

System.out.println(date1.getText());

Thread.sleep(3000);

break;

}

}

//Format Date

WebElement format\_date = driver.findElement(By.xpath("//a[contains(text(),'Format date')]"));

format\_date.click();

Thread.sleep(2000);

WebElement pick\_dt = driver.findElement(By.xpath("//\*[@id='datepicker4']"));

Thread.sleep(2000);

pick\_dt.click();

List<WebElement> select\_date21 = driver.findElements(By.xpath("//a[@class='ui-state-default']"));

int l1 = 9;

String str21 = null;

for(WebElement date1 : select\_date21){

str21 = Integer.toString(l1);

//System.out.println(date1.getText());

if(date1.getText().equals(str21)){

date1.click();

System.out.println(date1.getText());

Thread.sleep(3000);

break;

}

}

WebElement format = driver.findElement(By.xpath("//select[@id='format']"));

Select frmat = new Select(format);

for(int p=0;p<=5;p++){

frmat.selectByIndex(p);

Thread.sleep(3000);

}

//Menu

WebElement menu = driver.findElement(By.xpath("//a[contains(text(),'Menu')]"));

menu.click();

Thread.sleep(2000);

WebElement simple\_menu = driver.findElement(By.xpath("//a[contains(text(),'Simple Menu')]"));

simple\_menu.click();

Thread.sleep(2000);

WebElement home = driver.findElement(By.xpath("//div[1]/div/div/ul/li/a[contains(text(),'Home')]"));

home.click();

Thread.sleep(2000);

System.out.println("Click on Home :" +home.getText()+"");

WebElement about = driver.findElement(By.xpath("//div[1]/div/div/ul/li/a[contains(text(),'About')]"));

about.click();

Thread.sleep(2000);

System.out.println("Click on About :" +about.getText()+"");

// Menu with Sub Menu

WebElement menu\_with\_sub\_menu = driver.findElement(By.xpath("//a[contains(text(),'Menu With Sub Menu')]"));

menu\_with\_sub\_menu.click();

Thread.sleep(2000);

WebElement home1 = driver.findElement(By.xpath("//div/div[2]/div/div/ul/li/a[contains(text(),'Home')]"));

home1.click();

System.out.println("Click on Home :" +home1.getText()+"");

Thread.sleep(2000);

WebElement sub\_menu\_1 = driver.findElement(By.xpath("//\*[@id='navigate']/ul/li[1]/ul/li[1]/a"));

sub\_menu\_1.click();

System.out.println("Click on Sub menu :" +sub\_menu\_1.getText()+"") ;

Thread.sleep(2000);

WebElement about1 = driver.findElement(By.xpath("//div/div[2]/div/div/ul/li/a[contains(text(),'About')]"));

about1.click();

Thread.sleep(2000);

System.out.println("Click on About :" +about1.getText()+"");

//Slider

WebElement slider = driver.findElement(By.xpath("//a[contains(text(),'Slider')]"));

slider.click();

//Thread.sleep(2000);

//Range Slider

WebElement slider\_move = driver.findElement(By.xpath("//\*[@id='slider-range-max']/span"));

Actions move = new Actions(driver);

Thread.sleep(2000);

for(int n =-10;n<=140;n=n+10){

move.dragAndDropBy(slider\_move,n , 0).build().perform();

Thread.sleep(1000);

}

Thread.sleep(3000);

move.dragAndDropBy(slider\_move,-1200 , 0).build().perform();

Thread.sleep(3000);

for (int i1 = 1; i1 <= 10 ; i1++) {

slider\_move.sendKeys(Keys.ARROW\_RIGHT);

Thread.sleep(2000);

}

//Tabs

WebElement tabs = driver.findElement(By.xpath("//div/ul/li/a[contains(text(),'Tabs')]"));

tabs.click();

Thread.sleep(5000);

WebElement tab2 = driver.findElement(By.xpath("//a[contains(text(),'Tab 2')]"));

tab2.click();

Thread.sleep(5000);

//Tooltip

WebElement tooltip = driver.findElement(By.xpath("//a[contains(text(),'Tooltip')]"));

tooltip.click();

Thread.sleep(5000);

//Default functionality

WebElement age\_txt = driver.findElement(By.xpath("//\*[@id='age']"));

Actions hover = new Actions(driver);

hover.moveToElement(age\_txt).build().perform();

Thread.sleep(5000);

\*/

//Frames and windows

WebElement frames\_windows1 = driver.findElement(By.*xpath*("//a[contains(text(),'Frames and windows')]"));

frames\_windows1.click();

Thread.*sleep*(3000);

String root = driver.getWindowHandle();

//FrameSet button

WebElement frameset = driver.findElement(By.*xpath*("//a[(text()='Frameset')]"));

frameset.click();

Thread.*sleep*(3000);

//Open FrameSet Window link

WebElement frameset\_window = driver.findElement(By.*xpath*("//a[(text()='Open Frameset Window')]"));

frameset\_window.click();

Thread.*sleep*(3000);

Set<String> all\_handles = driver.getWindowHandles();

**for**(String handle : all\_handles){

**if**(!root.equals(handle)){

driver.switchTo().window(handle);

System.***out***.println("Window handle:" +handle);

WebElement content\_frame = driver.findElement(By.*xpath*("//frame[@name='contentFrame']"));

**if**(content\_frame.isDisplayed())

System.***out***.println("Move to the Content Frame");

Thread.*sleep*(5000);

}

}

driver.switchTo().window(root);

Thread.*sleep*(6000);

WebElement frames\_windows = driver.findElement(By.*xpath*("//a[contains(text(),'Frames and windows')]"));

frames\_windows.click();

Thread.*sleep*(3000);

String original\_window\_handle = driver.getWindowHandle();

//New Browser Tab link

WebElement new\_browser\_tab = driver.findElement(By.*xpath*("//a[contains(text(),'New Browser Tab')]"));

new\_browser\_tab.click();

Thread.*sleep*(3000);

String new\_browser\_tab\_handle = driver.getWindowHandle();

//Open Seperate New Window button

WebElement open\_seperate\_new\_wondow = driver.findElement(By.*xpath*("//a[contains(text(),'Open Seprate New Window')]"));

open\_seperate\_new\_wondow.click();

Thread.*sleep*(3000);

//Open Seperate New Window link

WebElement open\_new\_window\_link = driver.findElement(By.*xpath*("//\*[@id='tabs-2']/div/p/a"));

open\_new\_window\_link.click();

Thread.*sleep*(3000);

String seperate\_new\_window\_handle = driver.getWindowHandle();

Thread.*sleep*(5000);

driver.close();

driver.quit();

}

}

**Practice\_Draggable\_Droppable**

**package** com.Practice\_Draggable;

**import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.JavascriptExecutor;

**import** org.openqa.selenium.Keys;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.interactions.Actions;

**public** **class** Practice\_Draggable {

**public** **static** **void** main(String args[]) **throws** InterruptedException {

String path = "lib\\chromedriver.exe";

System.*setProperty*("webdriver.chrome.driver", path);

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("http://demoqa.com");

Thread.*sleep*(5000);

JavascriptExecutor js = (JavascriptExecutor) driver;

// This will scroll down the page by 1000 pixel vertical

js.executeScript("window.scrollBy(0,1000)");

Thread.*sleep*(5000);

js.executeScript("window.scrollBy(0,-1000)");

Thread.*sleep*(5000);

WebElement draggable = driver.findElement(By.*xpath*("//div/ul/li[1]/a[1][contains(text(),'Draggable')]"));

Thread.*sleep*(3000);

draggable.click();

Thread.*sleep*(3000);

// Element(BANK) which need to drag.

// Default Functionality

WebElement From = driver.findElement(By.*xpath*("//div[@id='draggable']"));

// Using Action class for drag and drop.

Actions act = **new** Actions(driver);

// Drag and Drop by Pixel.

**for** (**int** i = 10; i < 100; i = i + 10) {

act.dragAndDropBy(From, i, i).build().perform();

Thread.*sleep*(1000);

}

WebElement constrain\_movement = driver.findElement(By.*xpath*("//a[contains(text(),'Constrain movement')]"));

constrain\_movement.click();

Thread.*sleep*(1000);

// I can be dragged only vertically

WebElement from = driver.findElement(By.*xpath*("//p[contains(text(),'I can be dragged only vertically')]"));

// System.out.println(from.getLocation());

**for** (**int** i = 10; i < 60; i = i + 10) {

act.dragAndDropBy(from, 0, i).build().perform();

Thread.*sleep*(1000);

}

// I can be dragged only horizontally

WebElement from1 = driver.findElement(By.*xpath*("//p[contains(text(),'I can be dragged only horizontally')]"));

**for** (**int** i = 10; i < 60; i = i + 10) {

act.dragAndDropBy(from1, i, 0).build().perform();

Thread.*sleep*(1000);

}

// I’m contained within the box

WebElement from2 = driver.findElement(By.*xpath*("//p[contains(text(),'I’m contained within the box')]"));

**for** (**int** i = 0; i < 15; i++) {

act.dragAndDropBy(from2, i + 50, i).build().perform();

Thread.*sleep*(1000);

}

// I’m contained within my parent

WebElement from3 = driver.findElement(By.*xpath*("//p[contains(text(),'I’m contained within my parent')]"));

**for** (**int** i = 0; i < 15; i++) {

act.dragAndDropBy(from3, i, i).build().perform();

Thread.*sleep*(1000);

}

// Cursor style

WebElement cursor\_style = driver.findElement(By.*xpath*("//a[contains(text(),'Cursor style')]"));

cursor\_style.click();

Thread.*sleep*(1000);

// I will always stick to the center (relative to the mouse)

WebElement from4 = driver.findElement(

By.*xpath*("//p[contains(text(),'I will always stick to the center (relative to the mouse)')]"));

**for** (**int** i = 0; i < 5; i++) {

act.dragAndDropBy(from4, i + 10, i + 10).build().perform();

Thread.*sleep*(1000);

}

// My cursor is at left -5 and top -5

WebElement from5 = driver.findElement(By.*xpath*("//p[contains(text(),'My cursor is at left -5 and top -5')]"));

**for** (**int** i = 0; i < 5; i++) {

act.dragAndDropBy(from5, i + 10, i + 10).build().perform();

Thread.*sleep*(1000);

}

// My cursor position is only controlled for the ‘bottom’ value

WebElement from6 = driver.findElement(

By.*xpath*("//p[contains(text(),'My cursor position is only controlled for the ‘bottom’ value')]"));

**for** (**int** i = 0; i < 5; i++) {

act.dragAndDropBy(from6, i + 10, i + 10).build().perform();

Thread.*sleep*(1000);

}

// Events

WebElement events = driver.findElement(By.*xpath*("//a[contains(text(),'Events')]"));

events.click();

Thread.*sleep*(1000);

WebElement from7 = driver

.findElement(By.*xpath*("//p[contains(text(),'Drag me to trigger the chain of events.')]"));

**for** (**int** i = 0; i < 100; i = i + 10) {

act.dragAndDropBy(from7, i + 10, i + 10).build().perform();

Thread.*sleep*(1000);

}

// Draggable + Sortable

WebElement Draggable\_Sortable = driver.findElement(By.*xpath*("//a[contains(text(),'Draggable + Sortable')]"));

Draggable\_Sortable.click();

Thread.*sleep*(1000);

**int** k = 0;

**for** (**int** i = 1; i <= 5; i++) {

WebElement from8 = driver.findElement(By.*xpath*("//li[contains(text(),'Item " + i + "')]"));

Thread.*sleep*(2000);

act.dragAndDropBy(from8, 0, k = k + 20).build().perform();

Thread.*sleep*(3000);

}

// Droppable

// Default Functionality

Actions act1 = **new** Actions(driver);

WebElement Droppable = driver.findElement(By.*xpath*("//a[contains(text(),'Droppable')]"));

Droppable.click();

Thread.*sleep*(2000);

WebElement drag\_from = driver

.findElement(By.*xpath*(" //div/div[1]/div/div/p[contains(text(),'Drag me to my target')]"));

WebElement drop\_to = driver.findElement(By.*xpath*("//div[@id='droppableview']"));

act1.dragAndDrop(drag\_from, drop\_to).build().perform();

Thread.*sleep*(3000);

// Accept

WebElement accept = driver.findElement(By.*xpath*("//a[contains(text(),'Accept')]"));

accept.click();

Thread.*sleep*(2000);

WebElement to\_drag = driver.findElement(By.*xpath*("//div[@id='droppableaccept']"));

WebElement from\_candrop = driver

.findElement(By.*xpath*("//div/div[2]/div/div/p[contains(text(),'Drag me to my target')]"));

act1.dragAndDrop(from\_candrop, to\_drag).build().perform();

Thread.*sleep*(3000);

WebElement from\_cantdropped = driver

.findElement(By.*xpath*("//p[contains(text(),'I’m draggable but can’t be dropped')]"));

act1.dragAndDrop(from\_cantdropped, to\_drag).build().perform();

Thread.*sleep*(3000);

// Prevent propagation

WebElement prevent\_propagation = driver.findElement(By.*xpath*("//a[contains(text(),'Prevent propagation')]"));

prevent\_propagation.click();

Thread.*sleep*(2000);

WebElement from\_drag1 = driver

.findElement(By.*xpath*("//div/div[3]/div/div/p[contains(text(),'Drag me to my target')]"));

WebElement to\_drop1 = driver.findElement(By.*xpath*("//div[@id='droppable-inner']"));

act1.dragAndDrop(from\_drag1, to\_drop1).build().perform();

Thread.*sleep*(5000);

List<WebElement> dropped\_count = driver.findElements(By.*xpath*("//p[contains(text(),'Dropped!')]"));

System.***out***.println(dropped\_count.size());

**if** (dropped\_count.size() == 4) {

System.***out***.println("2 Dropped displayed");

}

driver.navigate().refresh();

Thread.*sleep*(3000);

WebElement prevent\_propagation1 = driver.findElement(By.*xpath*("//a[contains(text(),'Prevent propagation')]"));

prevent\_propagation1.click();

WebElement from\_drag2 = driver

.findElement(By.*xpath*("//div/div[3]/div/div/p[contains(text(),'Drag me to my target')]"));

WebElement to\_drop2 = driver.findElement(By.*xpath*("//div[@id='droppable2-inner']"));

act1.dragAndDrop(from\_drag2, to\_drop2).build().perform();

Thread.*sleep*(5000);

List<WebElement> dropped\_count1 = driver.findElements(By.*xpath*("//p[contains(text(),'Dropped!')]"));

System.***out***.println(dropped\_count1.size());

**if** (dropped\_count1.size() == 1) {

System.***out***.println("1 Dropped displayed");

}

// Shopping cart demo

WebElement Shopping\_cart\_demo = driver.findElement(By.*xpath*("//a[contains(text(),'Shopping cart demo')]"));

Shopping\_cart\_demo.click();

WebElement from\_Lolcat\_Shirt = driver.findElement(By.*xpath*("//li[contains(text(),'Lolcat Shirt')]"));

WebElement to\_cart = driver.findElement(By.*xpath*("//ol[@class='ui-droppable ui-sortable']"));

act1.dragAndDrop(from\_Lolcat\_Shirt, to\_cart).build().perform();

Thread.*sleep*(2000);

WebElement Cheezeburger\_Shirt = driver.findElement(By.*xpath*("//li[contains(text(),'Cheezeburger Shirt')]"));

act1.dragAndDrop(Cheezeburger\_Shirt, to\_cart).build().perform();

Thread.*sleep*(2000);

List<WebElement> from\_Lolcat\_Shirt\_count = driver

.findElements(By.*xpath*("//li[contains(text(),'Lolcat Shirt')]"));

List<WebElement> Cheezeburger\_Shirt\_count = driver

.findElements(By.*xpath*("//li[contains(text(),'Cheezeburger Shirt')]"));

**if** (from\_Lolcat\_Shirt\_count.size() == 2 && Cheezeburger\_Shirt\_count.size() == 2) {

System.***out***.println("Lolcat Shirt added to the cart");

System.***out***.println("Cheezeburger Shirt added to the cart");

}

Thread.*sleep*(5000);

// Resizable

// Default functionality

WebElement resizable = driver.findElement(By.*xpath*("//a[contains(text(),'Resizable')]"));

resizable.click();

Thread.*sleep*(3000);

WebElement resizable\_object = driver.findElement(By.*xpath*("//\*[@id='resizable']/div[3]"));

**if** (resizable\_object.isDisplayed()) {

Actions action = **new** Actions(driver);

action.clickAndHold(resizable\_object).moveByOffset(500, 550).release().build().perform();

System.***out***.println("Element was displayed and resized");

} **else** {

System.***out***.println("Element was not displayed to drag");

}

Thread.*sleep*(5000);

// Animate

WebElement animate = driver.findElement(By.*xpath*("//a[contains(text(),'Animate')]"));

animate.click();

WebElement animate\_object = driver.findElement(By.*xpath*("//\*[@id='resizableani']/div[3]"));

Actions act10 = **new** Actions(driver);

act10.clickAndHold(animate\_object).moveByOffset(200, 200).release().build().perform();

Thread.*sleep*(5000);

// Constrain resize area

WebElement constrain\_resize\_area = driver

.findElement(By.*xpath*("//a[contains(text(),'Constrain resize area')]"));

constrain\_resize\_area.click();

WebElement resizableconstrain\_object = driver.findElement(By.*xpath*("//\*[@id='resizableconstrain']/div[3]"));

Actions act2 = **new** Actions(driver);

act2.clickAndHold(resizableconstrain\_object).moveByOffset(100, 100).release().build().perform();

Thread.*sleep*(5000);

// Helper

WebElement helper = driver.findElement(By.*xpath*("//a[contains(text(),'Helper')]"));

helper.click();

WebElement helper\_object = driver.findElement(By.*xpath*("//\*[@id='resizable\_helper']/div[3]"));

Actions act3 = **new** Actions(driver);

act3.clickAndHold(helper\_object).moveByOffset(150, 150).release().build().perform();

Thread.*sleep*(5000);

// Selectable

WebElement selectable = driver.findElement(By.*xpath*("//a[contains(text(),'Selectable')]"));

selectable.click();

// Default functionality

// WebElement selectable =

// driver.findElement(By.xpath("//a[contains(text(),'Selectable')]"));

// selectable.click();

Actions ac = **new** Actions(driver);

List<WebElement> selectable\_objects = driver.findElements(By.*xpath*("//\*[@id='selectable']/li"));

**for** (WebElement select : selectable\_objects) {

ac.keyDown(Keys.***CONTROL***);

ac.click(select).build().perform();

ac.keyDown(Keys.***CONTROL***);

Thread.*sleep*(5000);

}

// Display as grid

WebElement display\_as\_grid = driver.findElement(By.*xpath*("//a[contains(text(),'Display as grid')]"));

display\_as\_grid.click();

Actions ac1 = **new** Actions(driver);

// WebElement selectable\_objects1 =

// driver.findElement(By.xpath("//\*[@id='selectable\_grid']/li["+1+"]"));

**for** (**int** i = 1; i <= 12; i++) {

WebElement selectable\_objects1 = driver.findElement(By.*xpath*("//\*[@id='selectable\_grid']/li[" + i + "]"));

ac1.keyDown(Keys.***CONTROL***);

ac1.click(selectable\_objects1).build().perform();

ac1.keyDown(Keys.***CONTROL***);

Thread.*sleep*(5000);

}

Thread.*sleep*(5000);

// Sortable

WebElement sortable = driver.findElement(By.*xpath*("//a[contains(text(),'Sortable')]"));

sortable.click();

Thread.*sleep*(3000);

// Default Functionality

WebElement i1 = driver.findElement(By.*xpath*("//div/div/div[1]/div/ul/li[contains(text(),'Item 1')]"));

Actions from\_i1 = **new** Actions(driver);

from\_i1.dragAndDropBy(i1, 100, 170).build().perform();

Thread.*sleep*(5000);

WebElement i2 = driver.findElement(By.*xpath*("//div/div/div[1]/div/ul/li[contains(text(),'Item 2')]"));

Actions from\_i2 = **new** Actions(driver);

from\_i1.dragAndDropBy(i2, 100, 170).build().perform();

Thread.*sleep*(5000);

WebElement i3 = driver.findElement(By.*xpath*("//div/div/div[1]/div/ul/li[contains(text(),'Item 3')]"));

Actions from\_i3 = **new** Actions(driver);

from\_i1.dragAndDropBy(i3, 100, 170).build().perform();

Thread.*sleep*(5000);

WebElement i4 = driver.findElement(By.*xpath*("//div/div/div[1]/div/ul/li[contains(text(),'Item 4')]"));

Actions from\_i4 = **new** Actions(driver);

from\_i1.dragAndDropBy(i4, 100, 170).build().perform();

Thread.*sleep*(5000);

// Display as grid

WebElement display\_as\_grid1 = driver.findElement(By.*xpath*("//a[contains(text(),'Display as grid')]"));

display\_as\_grid1.click();

Thread.*sleep*(3000);

Actions fgrid = **new** Actions(driver);

WebElement grid1 = driver.findElement(By.*xpath*("//\*[@id='sortable\_grid']/li[1]"));

System.***out***.println(grid1.getText());

fgrid.dragAndDropBy(grid1, 0, 100).build().perform();

Thread.*sleep*(5000);

WebElement grid2 = driver.findElement(By.*xpath*("//\*[@id='sortable\_grid']/li[2]"));

System.***out***.println(grid2.getText());

fgrid.dragAndDropBy(grid2, 0, 100).build().perform();

Thread.*sleep*(5000);

WebElement grid3 = driver.findElement(By.*xpath*("//\*[@id='sortable\_grid']/li[3]"));

System.***out***.println(grid3.getText());

fgrid.dragAndDropBy(grid3, 0, 100).build().perform();

Thread.*sleep*(5000);

WebElement grid4 = driver.findElement(By.*xpath*("//\*[@id='sortable\_grid']/li[4]"));

System.***out***.println(grid4.getText());

fgrid.dragAndDropBy(grid4, 0, 100).build().perform();

Thread.*sleep*(5000);

// Portlets

WebElement portlets = driver.findElement(By.*xpath*("//a[contains(text(),'Portlets')]"));

portlets.click();

Actions portlets\_act = **new** Actions(driver);

List<WebElement> portlets\_object = driver.findElements(

By.*xpath*("//div[@class='portlet-header ui-sortable-handle ui-widget-header ui-corner-all']"));

**for** (WebElement element : portlets\_object) {

portlets\_act.dragAndDropBy(element, 300, 0).build().perform();

System.***out***.println(element.getText());

Thread.*sleep*(7000);

}

// WIDGET

// -------

// Accordion

// Default functionality

WebElement accordion = driver.findElement(By.*xpath*("//a[contains(text(),'Accordion')]"));

accordion.click();

Thread.*sleep*(7000);

WebElement section2 = driver.findElement(By.*xpath*("//\*[@id='ui-id-6']/span"));

section2.click();

Thread.*sleep*(7000);

WebElement section\_scroll = driver.findElement(By.*xpath*("//\*[@id='ui-id-7']"));

Actions accordion\_default = **new** Actions(driver);

js.executeScript("arguments[0].scrollBy(0,1000);", section\_scroll);

Thread.*sleep*(7000);

// Autocomplete

// Default functionality

WebElement autocomplete = driver.findElement(By.*xpath*("//a[contains(text(),'Autocomplete')]"));

autocomplete.click();

WebElement tags = driver.findElement(By.*xpath*("//\*[@id='tagss']"));

tags.sendKeys("a");

Thread.*sleep*(3000);

WebElement applescript = driver.findElement(By.*xpath*("//li[contains(text(),'AppleScript')]"));

System.***out***.println(applescript.getText());

applescript.click();

// Multiple Values

Thread.*sleep*(5000);

WebElement multiple\_values = driver.findElement(By.*xpath*("//a[contains(text(),'Multiple Values')]"));

multiple\_values.click();

WebElement tag\_programming\_language = driver.findElement(By.*xpath*("//\*[@id='tagsss']"));

tag\_programming\_language.click();

tag\_programming\_language.sendKeys("c");

Thread.*sleep*(3000);

WebElement cobol = driver.findElement(By.*xpath*("//li[contains(text(),'COBOL')]"));

System.***out***.println(cobol.getText());

cobol.click();

Thread.*sleep*(5000);

tag\_programming\_language.sendKeys("l");

Thread.*sleep*(6000);

WebElement javascript = driver.findElement(By.*xpath*("//li[contains(text(),'Lisp')]"));

System.***out***.println(javascript.getText());

javascript.click();

Thread.*sleep*(7000);

// Thread.sleep(5000);

tag\_programming\_language.sendKeys("k");

Thread.*sleep*(1000);

/\*

\* WebElement basic =

\* driver.findElement(By.xpath("//li[contains(text(),'Haskell')]"));

\*

\* System.out.println(basic.getText());

\*

\* Thread.sleep(5000);

\*

\* basic.click();

\*/

Thread.*sleep*(5000);

// Categories

WebElement categories = driver.findElement(By.*xpath*("//a[contains(text(),'Categories')]"));

categories.click();

Thread.*sleep*(3000);

WebElement search\_txt = driver.findElement(By.*xpath*("//\*[@id='searcha']"));

search\_txt.click();

search\_txt.sendKeys("p");

Thread.*sleep*(3000);

WebElement annttop = driver.findElement(By.*xpath*(" //li[contains(text(),'annttop C13')]"));

annttop.click();

Thread.*sleep*(3000);

driver.close();

driver.quit();

/\*

\* driver.switchTo().window(null); driver.switchTo().alert();

\* driver.switchTo().frame(0);

\*/

}

}

**Widgets**

**package** com.Widgets;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.JavascriptExecutor;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.interactions.Actions;

**public** **class** Widgets {

**public** **static** **void** main(String args[]) **throws** InterruptedException{

String path = "lib\\chromedriver.exe";

System.*setProperty*("webdriver.chrome.driver", path);

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("http://jqueryui.com/");

Thread.*sleep*(5000);

String root\_handle = driver.getWindowHandle();

JavascriptExecutor js = (JavascriptExecutor) driver;

// This will scroll down the page by 1000 pixel vertical

js.executeScript("window.scrollBy(0,1000)");

Thread.*sleep*(5000);

js.executeScript("window.scrollBy(0,-1000)");

Thread.*sleep*(5000);

//CheckBoxRadio

WebElement checkboxradio = driver.findElement(By.*xpath*("//a[contains(text(),'Checkboxradio')]"));

checkboxradio.click();

Thread.*sleep*(10000);

//Move to iFrame

WebElement iframe = driver.findElement(By.*xpath*("//iframe"));

driver.switchTo().frame(iframe);

Thread.*sleep*(5000);

//NewYork

WebElement newyork = driver.findElement(By.*xpath*("//label[contains(text(),'New York')]/span[1]"));

//WebElement newyork = driver.findElement(By.xpath("//html/body/div/fieldset[1]/label[1]/span[1]"));

newyork.click();

System.***out***.println("Where did you click:"+newyork.getText());

Thread.*sleep*(5000);

//Hotel Rating

WebElement hotel\_rating = driver.findElement(By.*xpath*("//label[contains(text(),'2 Star')]/span[1]"));

hotel\_rating.click();

Thread.*sleep*(5000);

WebElement scroll = driver.findElement(By.*xpath*("/html[@lang='en']"));

Actions act = **new** Actions(driver);

act.moveToElement(scroll, 0, 1000);

Thread.*sleep*(4000);

WebElement doubl = driver.findElement(By.*xpath*("//label[contains(text(),'Double')]/span[1]"));

doubl.click();

Thread.*sleep*(5000);

//move to the root window

driver.switchTo().window(root\_handle);

//ControlGroup

WebElement control\_group = driver.findElement(By.*xpath*("//a[contains(text(),'Controlgroup')]"));

control\_group.click();

Thread.*sleep*(8000);

//Move to frame

WebElement frame1 = driver.findElement(By.*xpath*("//iframe"));

driver.switchTo().frame(frame1);

Thread.*sleep*(4000);

//Car type

WebElement fieldset = driver.findElement(By.*xpath*("//\*[@id='car-type-button']/span[1]"));

fieldset.click();

Thread.*sleep*(3000);

WebElement select\_car = driver.findElement(By.*xpath*("//div[2]/ul/li/div[contains(text(),'Luxury')]"));

select\_car.click();

Thread.*sleep*(5000);

//move to the root window

driver.switchTo().window(root\_handle);

//Dialog

WebElement dialog = driver.findElement(By.*xpath*("//a[contains(text(),'Dialog')]"));

dialog.click();

Thread.*sleep*(4000);

//move to iframe

WebElement iframe2 = driver.findElement(By.*xpath*("//iframe"));

driver.switchTo().frame(iframe2);

Thread.*sleep*(3000);

//Close Dialog

WebElement close\_dialog = driver.findElement(By.*xpath*("//button[contains(text(),'Close')]"));

close\_dialog.click();

Thread.*sleep*(4000);

//move to the root window

driver.switchTo().window(root\_handle);

//Menu

WebElement menu = driver.findElement(By.*xpath*("//a[contains(text(),'Menu')]"));

menu.click();

Thread.*sleep*(4000);

//move to iframe

WebElement iframe3 = driver.findElement(By.*xpath*("//iframe"));

driver.switchTo().frame(iframe3);

Thread.*sleep*(3000);

//Open Sub Menu

WebElement sub\_menu = driver.findElement(By.*xpath*("//div[contains(text(),'Music')]"));

Actions mouse\_hover = **new** Actions(driver);

mouse\_hover.moveToElement(sub\_menu).build().perform();

Thread.*sleep*(4000);

//sub\_menu.click();

//Open Sub Sub Menu

WebElement sub\_sub\_menu = driver.findElement(By.*xpath*("//div[contains(text(),'Rock')]"));

mouse\_hover.moveToElement(sub\_sub\_menu).build().perform();

Thread.*sleep*(10000);

// sub\_sub\_menu.click();

//Mouse hover on sub sub sub menu

WebElement sub\_sub\_sub\_menu= driver.findElement(By.*xpath*("//div[contains(text(),'Classic')]"));

mouse\_hover.moveToElement(sub\_sub\_sub\_menu).build().perform();

Thread.*sleep*(5000);

//sub\_sub\_sub\_menu.click();

Thread.*sleep*(5000);

//move to the root window

driver.switchTo().window(root\_handle);

//Progressabr

WebElement progressbar = driver.findElement(By.*xpath*("//a[contains(text(),'Progressbar')]"));

progressbar.click();

Thread.*sleep*(4000);

//download dialog

WebElement download\_dialog = driver.findElement(By.*xpath*("//a[contains(text(),'Download Dialog')]"));

download\_dialog.click();

Thread.*sleep*(3000);

//move to frame

WebElement iframe4= driver.findElement(By.*xpath*("//iframe"));

driver.switchTo().frame(iframe4);

Thread.*sleep*(4000);

//Start Download

WebElement start\_download = driver.findElement(By.*xpath*("//button[contains(text(),'Start Download')]"));

start\_download.click();

Thread.*sleep*(15000);

//Check the Download complete

WebElement download\_complete = driver.findElement(By.*xpath*("//div[contains(text(),'Complete!')]"));

**if**(download\_complete.isDisplayed())

System.***out***.println("Download complete");

Thread.*sleep*(5000);

//move to the root window

driver.switchTo().window(root\_handle);

driver.close();

driver.quit();

}

}

**CreateFormInGoogleDrive**

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.util.List;

**import** java.util.Set;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** CreateFormInGoogleDrive {

**public** **static** **void** main(String[] args)**throws** Exception

{

String path = "chromedriver.exe";

System.*setProperty*("webdriver.chrome.driver", path);

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("https://mail.google.com/");

Thread.*sleep*(8000);

//File file = new File("C:\\Users\\kiwiQA\\Desktop\\Vipul\\credentials.txt");

File file = **new** File("F:\\eclipse-workspace\\CreateFormInGoogleDrive\\credentials.txt");

BufferedReader br = **new** BufferedReader(**new** FileReader(file));

String username = **null**;

String password = **null**;

**int** i = 0;

String st;

**while** ((st = br.readLine()) != **null**)

{

**if**(i==0)

username = st.trim();

**else**

password = st.trim();

i++;

System.***out***.println(st);

}

WebElement user\_name = driver.findElement(By.*xpath*("//input[@aria-label='Email or phone']"));

user\_name.click();

user\_name.sendKeys(username);

WebElement next\_btn = driver.findElement(By.*xpath*("//span[text()='Next']"));

next\_btn.click();

Thread.*sleep*(5000);

WebElement psw = driver.findElement(By.*xpath*("//input[@aria-label='Enter your password']"));

psw.click();

psw.sendKeys(password);

WebElement next\_btn\_psw = driver.findElement(By.*xpath*("//span[text()='Next']"));

next\_btn\_psw.click();

Thread.*sleep*(10000);

WebElement google\_apps = driver.findElement(By.*xpath*("//a[@title='Google apps']"));

google\_apps.click();

Thread.*sleep*(4000);

WebElement drive = driver.findElement(By.*xpath*("//span[text()='Drive']"));

drive.click();

Thread.*sleep*(8000);

Set<String> handles = driver.getWindowHandles();

String handle = **null**;

**for**(String handle\_win : handles) {

handle = handle\_win;

}

driver.switchTo().window(handle);

Thread.*sleep*(8000);

WebElement new\_btn = driver.findElement(By.*xpath*("//div/button[1]/div[contains(text(),'New')]/..//div[2]"));

new\_btn.click();

Thread.*sleep*(3000);

WebElement more\_btn = driver.findElement(By.*xpath*("//div[contains(text(),'More')]"));

more\_btn.click();

Thread.*sleep*(3000);

WebElement google\_form\_btn = driver.findElement(By.*xpath*("//div[contains(text(),'Google Forms')]"));

google\_form\_btn.click();

Set<String> handles1 = driver.getWindowHandles();

String handle1 = **null**;

**for**(String handle\_win : handles1) {

handle1 = handle\_win;

}

driver.switchTo().window(handle1);

Thread.*sleep*(8000);

**for**(**int** k=1;k<=5;k++) {

**if**(k==1){

WebElement untitled\_que = driver.findElement(By.*xpath*("//textarea[contains(text(),'Untitled Question')]"));

untitled\_que.click();

untitled\_que.sendKeys("My first question");

}

**else** {

List<WebElement> untitled\_que\_list = driver.findElements(By.*xpath*("//div[@class='freebirdFormeditorViewItemTitleRow']//textarea"));

**int** queCount = untitled\_que\_list.size()-1;

WebElement untitled\_que = untitled\_que\_list.get(queCount);

untitled\_que.click();

**switch**(k) {

**case** 1:

**break**;

**case** 2:

untitled\_que.sendKeys("My second question");

**break**;

**case** 3:

untitled\_que.sendKeys("My third question");

List<WebElement> multi\_option = driver.findElements(By.*xpath*("//div[@class='quantumWizMenuPaperselectDropDown exportDropDown']"));

Thread.*sleep*(8000);

**int** multioption = multi\_option.size()-1;

multi\_option.get(multioption).click();

Thread.*sleep*(8000);

List<WebElement> multi\_choice\_clk = driver.findElements(By.*xpath*("//textarea[@data-initial-value='My third question']/../../../../../../..//div//div[contains(@class,'-radio-')]/../../../content[contains(text(),'Multiple choice')]"));

multi\_choice\_clk.get(1).click();

/\*String str1 = multi\_option.get(multioption).getText();

System.out.println(str1);

String str = "Multiple choice";

if (!str1.equals(str)) {

}\*/

**break**;

**case** 4:

untitled\_que.sendKeys("My fourth question");

**break**;

**default**:

untitled\_que.sendKeys("My fifth question");

**break**;

}

}

Thread.*sleep*(3000);

List<WebElement> add\_option = driver.findElements(By.*xpath*("//input[@aria-label='Add option']"));

**int** p = add\_option.size();

System.***out***.println(p);

Thread.*sleep*(8000);

add\_option.get(p-1).click();

Thread.*sleep*(2000);

add\_option.get(p-1).click();

Thread.*sleep*(2000);

add\_option.get(p-1).click();

Thread.*sleep*(2000);

add\_option.get(p-1).sendKeys("Done");

Thread.*sleep*(2000);

**for** (**int** j=1; j<5;j++) {

String s = Integer.*toString*(j);

//WebElement answer = driver.findElement(By.xpath("//input[@value='Option "+j+"']"));

WebElement answer = driver.findElement(By.*xpath*("//input[@data-initial-value='Option "+j+"']"));

answer.click();

answer.sendKeys(s);

Thread.*sleep*(3000);

}

//WebElement answer = driver.findElement(By.xpath("//input[@value='Option "+1+"']"));

//input[@value='Option 1']

WebElement add\_que = driver.findElement(By.*xpath*("//div/div/div/div/div/div/div/div[@aria-label='Add question']"));

add\_que.click();

}

Thread.*sleep*(15000);

driver.quit();

}

}

**Table**

**package** demo2;

**import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **interface** demo2 {

**public** **static** **void** main(String args[]) **throws** InterruptedException {

String path = "chromedriver.exe";

System.*setProperty*("webdriver.chrome.driver", path);

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("http://toolsqa.com/");

Thread.*sleep*(6000);

WebElement demo\_link = driver.findElement(By.*xpath*("//span/span/span[contains(text(),'DEMO SITES')]"));

demo\_link.click();

Thread.*sleep*(4000);

WebElement practice\_link = driver.findElement(By.*xpath*("//span/span/span[contains(text(),'Automation Practice Table')]"));

practice\_link.click();

Thread.*sleep*(5000);

List<WebElement> no\_tr = driver.findElements(By.*xpath*("//tbody/tr"));

**int** count = no\_tr.size();

WebElement total = driver.findElement(By.*xpath*("//td [contains(text(),'buildings')]"));

System.***out***.println("Total number of result::" + count);

String count\_result = total.getText().replace("buildings", "");

count\_result = count\_result.trim();

Integer result = Integer.*valueOf*(count\_result);

**if**(result == count) {

System.***out***.println("Total number of result is match with Toatl::" + result);

}

/\*int t=0;

for(int i=0;i<count\_result.length();i++) {

char ch = count\_result.charAt(i);

if (ch == ' ') {

t = i;

count\_result = count\_result.substring(0,t);

break;

}

}

Integer result = Integer.valueOf(count\_result);

if(result == (count-2)) {

System.out.println("Total number of result is match with Toatl::" + result);

}

\*/

WebElement taiwan\_detail = driver.findElement(By.*xpath*("//td[text()='Taiwan']/..//a"));

**if**(taiwan\_detail.isDisplayed()) {

System.***out***.println("Taiwan Details field Displayed");

}

**if**(taiwan\_detail.isSelected()) {

System.***out***.println("Taiwan Details field selected");

}

List<WebElement> all\_records = driver.findElements(By.*xpath*("//td[text()='Saudi Arabia']/../\*[position()<last()]"));

**for**(**int** i=0;i<all\_records.size();i++) {

String rec = all\_records.get(i).getText();

System.***out***.println("Saudi Arabia Records::"+all\_records.get(i).getText());

}

List<WebElement> all\_records\_height = driver.findElements(By.*xpath*("//tr/td[3]"));

**int** l;

**for**(**int** i=0;i<all\_records\_height.size();i++) {

String rec = all\_records\_height.get(i).getText();

rec = rec.replace("m", "");

rec = rec.trim();

result = Integer.*valueOf*(rec);

**if**(result > 600) {

System.***out***.println("Total Height greater than 600m ::" + all\_records\_height.get(i).getText());

}

}

**for**(**int** i=0;i<all\_records\_height.size();i++) {

String rec = all\_records\_height.get(i).getText();

rec = rec.replace("m", "");

rec = rec.trim();

result = Integer.*valueOf*(rec);

result = Integer.*valueOf*(rec);

**double** result1 = result \* 0.10 ;

**if**(result1 > 50.00) {

System.***out***.println("Record having 10% of height is more than 50::" + all\_records\_height.get(i).getText());

}

}

List<WebElement> years = driver.findElements(By.*xpath*("//tr/td[4]"));

**for**(**int** i=0;i<years.size();i++) {

result = Integer.*valueOf*(years.get(i).getText());

**if**(result > 2009) {

System.***out***.println("Build after 2009::" + years.get(i).getText());

}

}

List<WebElement> all\_records\_height1 = driver.findElements(By.*xpath*("//tr/td[3]"));

List<WebElement> years1 = driver.findElements(By.*xpath*("//tr/td[4]"));

**for**(**int** i=0;i<all\_records\_height1.size();i++) {

String rec = all\_records\_height1.get(i).getText();

rec = rec.replace("m", "");

rec = rec.trim();

result = Integer.*valueOf*(rec);

**if**(result < 600) {

result = Integer.*valueOf*(years1.get(i).getText());

**if**(result > 2005) {

System.***out***.println("Records with height < 600m and build after 2005::" + all\_records\_height1.get(i).getText() + " " +years1.get(i).getText());

}

}

}

taiwan\_detail.click();

Thread.*sleep*(4000);

driver.quit();

}

}

**Open Google Account**

**package** test1;

**import** java.nio.charset.Charset;

**import** java.util.List;

**import** java.util.Random;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** demo {

/\*\*

\* **@param** args

\* **@throws** InterruptedException

\*/

**public** **static** **void** main(String args[]) **throws** InterruptedException {

String path = "chromedriver.exe";

System.*setProperty*("webdriver.chrome.driver", path);

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("https://accounts.google.com");

System.***out***.println("Step-1 :: Open GOOGLE MAIL account.");

Thread.*sleep*(13000);

/\*String url = driver.getCurrentUrl();

System.out.println(url);\*/

WebElement email\_id = driver.findElement(By.*xpath*("//input[@type='email']"));

**if**(email\_id.isDisplayed()) {

System.***out***.println(" ->>>> Step-2 Element Email textfield Displayed");

}

String email\_id\_var = "vipulkiwiqa";

email\_id.sendKeys(email\_id\_var);

WebElement next\_btn = driver.findElement(By.*xpath*("//span[text()=\"Next\"]"));

next\_btn.click();

System.***out***.println("Email Id: " +email\_id\_var);

Thread.*sleep*(5000);

WebElement password = driver.findElement(By.*xpath*("//input[@type=\"password\"]"));

**if**(password.isDisplayed()) {

System.***out***.println(" ->>> > Step-3 Element Password textfield Displayed");

}

String psw\_var = "hetalmylove";

password.sendKeys(psw\_var);

WebElement pass\_next\_btn = driver.findElement(By.*xpath*("//span[text()=\"Next\"]"));

pass\_next\_btn.click();

Thread.*sleep*(10000);

WebElement gmail\_logo = driver.findElement(By.*xpath*("//a[@href=\"https://mail.google.com\"]"));

**if**(gmail\_logo.isDisplayed()) {

System.***out***.println("->>> Step-4 Element Gmail logo Display");

}

gmail\_logo.click();

Thread.*sleep*(5000);

WebElement compose\_btn = driver.findElement(By.*xpath*("//div[text()=\"COMPOSE\"]"));

WebElement gmail\_link = driver.findElement(By.*xpath*("//div/span[text()='Gmail']"));

// WebElement acc\_logo = driver.findElement(By.xpath("//span[@class=\"gb\_db gbii\"]"));

//&& acc\_logo.isDisplayed()

**if**(compose\_btn.isDisplayed() && gmail\_link.isDisplayed()) {

System.***out***.println("->>> Step-5 You are in Gmail ");

System.***out***.println("->>> Step-6 Gmail link displayed");

System.***out***.println("->>> step-7 account logo displayed ");

}

compose\_btn.click();

Thread.*sleep*(10000);

WebElement to\_txt\_fld = driver.findElement(By.*xpath*("//textarea[@name='to']"));

**if**(to\_txt\_fld.isDisplayed()) {

System.***out***.println("->>> Step-8 Enter Email Id ");

}

**int** leftLimit = 97; // letter 'a'

**int** rightLimit = 122; // letter 'z'

**int** targetStringLength = 3;

Random random = **new** Random();

StringBuilder buffer = **new** StringBuilder(targetStringLength);

**for** (**int** i = 0; i < targetStringLength; i++) {

**int** randomLimitedInt = leftLimit + (**int**)

(random.nextFloat() \* (rightLimit - leftLimit + 1));

buffer.append((**char**) randomLimitedInt);

}

String generatedString\_email = buffer.toString();

System.***out***.println(generatedString\_email);

String Rec\_emailId\_var = generatedString\_email+"@mailinator.com";

to\_txt\_fld.sendKeys(Rec\_emailId\_var );

System.***out***.println("->>>Step 9 Entered Email Id:- "+Rec\_emailId\_var);

WebElement sub\_txt\_fld = driver.findElement(By.*xpath*("//input[@name=\"subjectbox\"]"));

**int** leftLimit1 = 97; // letter 'a'

**int** rightLimit1 = 122; // letter 'z'

**int** targetStringLength1 = 10;

Random random1 = **new** Random();

StringBuilder buffer1 = **new** StringBuilder(targetStringLength1);

**for** (**int** i = 0; i < targetStringLength1; i++) {

**int** randomLimitedInt = leftLimit1 + (**int**)

(random1.nextFloat() \* (rightLimit1 - leftLimit1 + 1));

buffer1.append((**char**) randomLimitedInt);

}

String generatedString\_subject = buffer1.toString();

String sub\_var = generatedString\_subject;

sub\_txt\_fld.sendKeys(sub\_var);

System.***out***.println("->>>Step 10 Subject is: " +sub\_var);

WebElement msg\_txt\_fld = driver.findElement(By.*xpath*("//div[@aria-label='Message Body']"));

**int** leftLimit11 = 97; // letter 'a'

**int** rightLimit11 = 122; // letter 'z'

**int** targetStringLength11 = 50;

Random random11 = **new** Random();

StringBuilder buffer11 = **new** StringBuilder(targetStringLength11);

**for** (**int** i = 0; i < targetStringLength11; i++) {

**int** randomLimitedInt = leftLimit11 + (**int**)

(random11.nextFloat() \* (rightLimit11 - leftLimit11 + 1));

buffer11.append((**char**) randomLimitedInt);

}

String generatedString\_message = buffer11.toString();

System.***out***.println(generatedString\_message);

String msg\_var = generatedString\_message;

msg\_txt\_fld.sendKeys(msg\_var);

System.***out***.println("->>>Step 11 Message is: " +msg\_var);

WebElement send\_btn = driver.findElement(By.*xpath*("//div[@role='button'][contains(text(),'Send')]"));

send\_btn.click();

Thread.*sleep*(35000);

WebElement sent\_lnk = driver.findElement(By.*xpath*("//a[@title='Sent Mail']"));

sent\_lnk.click();

System.***out***.println("->>>Step 12 Sent emails open");

Thread.*sleep*(2000);

/\*//String xpath = "//div/span[1][@class='bog'][contains(text(),'"+sub\_var+"')]";

String xpath = "//span[@class='bog']";

List<WebElement> email = driver.findElements(By.xpath(xpath));

//System.out.println(email);

for(WebElement emailsub : email){

if(emailsub.getText().trim().equals(sub\_var)){

System.out.println("Subject : " + emailsub.getText());

emailsub.click();

System.out.println("->>> Step 13 Sent message with subject:My First Selenium Script!!! Thanks Tarpan is opened ");

break;

}

else {

System.out.println("No email sent with this subject");

}

}

WebElement sub\_txt = driver.findElement(By.xpath("//div[@class='yW'][@id=':2t']"));

//String subject\_var = sub\_txt.getText();

if(subject\_var.equalsIgnoreCase("My First Selenium Script!!! Thanks Tarpan")) {

sub\_txt.click(); div.xT>div.y6>span>b

}

\*/

driver.navigate().to("https://www.mailinator.com/");

Thread.*sleep*(10000);

WebElement mailinator\_emailid\_txt = driver.findElement(By.*xpath*("//input[@id='inboxfield']"));

mailinator\_emailid\_txt.click();

mailinator\_emailid\_txt.sendKeys(Rec\_emailId\_var);

WebElement go\_btn = driver.findElement(By.*xpath*("//span/button[@type='button']"));

go\_btn.click();

Thread.*sleep*(10000);

List<WebElement> email\_ids = driver.findElements(By.*xpath*("//div[@class='lb\_all\_sub-item\_text ng-binding']"));

**for**(WebElement sub\_email : email\_ids) {

String email\_chk = sub\_email.getText();

**if**(generatedString\_email.equals(email\_chk)) {

sub\_email.click();

System.***out***.println("Click on the email that I have just sent!!!");

}

}

Thread.*sleep*(5000);

//List<WebElement> mailinator\_email\_sub = driver.findElements(By.xpath("//img[contains(@class,'image-225')]"));

List<WebElement> mailinator\_email\_sub = driver.findElements(By.*xpath*("//div[contains(@class,'all\_message-min\_text all\_message-min\_text')]"));

**for**(WebElement sub : mailinator\_email\_sub) {

**if**(sub\_var.equals(sub.getText())) {

sub.click();

Thread.*sleep*(10000);

WebElement iframe = driver.findElement(By.*xpath*("//iframe[@id='msg\_body']"));

driver.switchTo().frame(iframe);

WebElement sig\_logo = driver.findElement(By.*xpath*("//a/img[contains(@src,'https://www.kiwiqa.com/')]"));

Boolean status = sig\_logo.isDisplayed();

**if**(status == **true**) {

System.***out***.println("Congratulation !!! Signature Displayed that you are looking for");

}

}

}

driver.switchTo().defaultContent();

Thread.*sleep*(15000);

driver.quit();

}

}

//driver.switchTo().frame(iframe);

//driver.switchTo().defaultContent();

**Commands**

**package** Commands;

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.support.ui.Select;

**public** **class** Commands {

**private** **static** **final** **int** ***List*** = 0;

**private** **static** **final** **boolean** ***flase*** = **false**;

**public** **static** **void** main(String args[]) **throws** InterruptedException, IOException {

System.***out***.println("Hello how are you today");

String path = "chromedriver.exe";

System.*setProperty*("webdriver.chrome.driver", path);

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

/\*driver.get("https://google.com");

System.out.println("Title of the page :->" +driver.getTitle());

System.out.println("Length of Title of the page :->" +driver.getTitle().length());

System.out.println("Current url is:-> " +driver.getCurrentUrl());

//System.out.println("Page source" + driver.getPageSource());

System.out.println("Page source length :->" + driver.getPageSource().length());

Thread.sleep(13000);

WebElement search\_txt = driver.findElement(By.xpath("//input[@title='Search']"));

search\_txt.click();

search\_txt.sendKeys("kiwiqa");

Thread.sleep(3000);

search\_txt.clear();

Thread.sleep(3000);

search\_txt.sendKeys("how are you?");

Thread.sleep(3000);

WebElement search\_btn = driver.findElement(By.xpath("//input[@value=\"Google Search\"]"));

search\_btn.click();

Thread.sleep(5000);

driver.navigate().to("http://www.DemoQA.com");

Thread.sleep(5000);\*/

//driver.navigate().back();

//Thread.sleep(5000);

//driver.navigate().forward();

//Thread.sleep(5000);

//driver.navigate().refresh();

//Thread.sleep(5000);

/\*List<WebElement> img\_list = driver.findElements(By.xpath("//img[@class='alignnone size-medium wp-image-225']"));

System.out.println(img\_list);

for(WebElement img\_cl : img\_list) {

boolean status = img\_cl.isDisplayed();

if(status == true) {

System.out.println("Element you are looking for is Displayed and clicked");

img\_cl.click();

}

Thread.sleep(5000);

driver.navigate().back();

Thread.sleep(10000);

}\*/

/\*

List<WebElement> imgs = driver.findElements(By.xpath("//div[@class='entry-content']/div//a/img[contains(@class,'image-225')]"));

for(int i=1;i<=imgs.size();i++) {

String xpath = "//div[@class='entry-content']/div["+i+"]//a/img[contains(@class,'image-225')]";

WebElement imgsEle = driver.findElement(By.xpath(xpath));

//System.out.println("->> TagName:--" +imgsEle.getTagName());

//System.out.println("->> GetText:--" +imgsEle.getText());

//System.out.println("->> GetAttribute:--" +imgsEle.getAttribute(xpath));

//System.out.println("->> GetSize:--" +imgsEle.getSize());

//System.out.println("->> GetRect:--" +imgsEle.getRect());

//System.out.println("->> GetLocation:-- " +imgsEle.getLocation());

//System.out.println("->> Getclass:--" +imgsEle.getClass());

//System.out.println("->> GetCssValue:--" +imgsEle.getCssValue(xpath));

boolean status = imgsEle.isDisplayed();

if(status == true) {

System.out.println("Element you are looking for is Displayed and clicked");

imgsEle.click();

//driver.findElement(By.xpath(xpath)).click();

}

driver.navigate().back();

Thread.sleep(10000);

}

\*/

driver.navigate().to("http://toolsqa.com/Automation-practice-form/");

Thread.*sleep*(5000);

/\*WebElement link = driver.findElement(By.linkText("Link Test"));

System.out.println(link.getText());

link.click();

Thread.sleep(4000);

driver.navigate().back();

Thread.sleep(4000);

WebElement partial\_link = driver.findElement(By.partialLinkText("Partial Link Test"));

System.out.println(partial\_link.getText());

partial\_link.click();

Thread.sleep(3000);\*/

WebElement firstname\_txt = driver.findElement(By.*xpath*("//input[@name='firstname']"));

firstname\_txt.click();

firstname\_txt.sendKeys("kiwiqa");

WebElement lastname\_txt = driver.findElement(By.*xpath*("//input[@name='lastname']"));

lastname\_txt.click();

lastname\_txt.sendKeys("ahmedabad");

List<WebElement> sex\_btn = driver.findElements(By.*name*("sex"));

Boolean select\_status = **false**;

String value;

**if**(select\_status == sex\_btn.get(0).isSelected()) {

sex\_btn.get(0).click();

System.***out***.println(sex\_btn.get(0).getText());

System.***out***.println(sex\_btn.get(0).getTagName());

value = sex\_btn.get(0).getAttribute("value");

System.***out***.println("Gender::"+value);

**if**(value.equals("Male")){

System.***out***.println("You have selected the Male gender");

}

}

Thread.*sleep*(4000);

**if**(select\_status == sex\_btn.get(1).isSelected()) {

sex\_btn.get(1).click();

System.***out***.println(sex\_btn.get(1).getText());

System.***out***.println(sex\_btn.get(1).getTagName());

value = sex\_btn.get(1).getAttribute("value");

System.***out***.println(value);

**if**(value.equals("Female")){

System.***out***.println("You have selected the Male gender");

}

}

List<WebElement> exp = driver.findElements(By.*name*("exp"));

Boolean state = **true**;

**int** count = 0;

**for**(**int** i=0;i<exp.size();i++) {

**if**(exp.get(i).isSelected()== state) {

count++;

}

}

**if**(count == 0) {

System.***out***.println("None Check Box is Selected");

exp.get(0).click();

**if**(exp.get(0).isSelected()) {

System.***out***.println("Radio button 1 is selected");

}

}

Thread.*sleep*(10000);

exp.get(1).click();

**for**(**int** i=0;i<exp.size();i++) {

**if**(exp.get(i).isSelected()== state) {

count++;

}

}

System.***out***.println("How many radio button has been selected? ::" +count);

Thread.*sleep*(4000);

WebElement date = driver.findElement(By.*id*("datepicker"));

date.sendKeys("01/01/2019");

Thread.*sleep*(3000);

List<WebElement> profession = driver.findElements(By.*cssSelector*("input[name='profession']"));

**for**(**int** i=0;i<profession.size();i++) {

**if**(profession.get(i).isSelected() == ***flase***) {

profession.get(i).click();

}

}

**for**(**int** i=0;i<profession.size();i++) {

**if**(profession.get(i).isSelected() == **true**) {

System.***out***.println("Check Box Selected :" +profession.get(i).getAttribute("value"));

}

}

Thread.*sleep*(4000);

/\* WebElement fileupload = driver.findElement(By.xpath("//input [@id='photo']"));

fileupload.sendKeys("C:\\Users\\kiwiQA\\Desktop\\Vipul\\tie.png");

System.out.println(fileupload.getAttribute("name"));

System.out.println(fileupload.getAttribute("file"));

System.out.println(fileupload.getText());

String filename = fileupload.getAttribute("value");

System.out.println(filename);

int l = filename.length();

int f = l - 7;

String testfile = filename.substring(f, l);

System.out.println(testfile);

if(testfile.equals("tie.png")) {

System.out.println("File uploaded successfully");

}\*/

WebElement fileupload = driver.findElement(By.*xpath*("//input [@id='photo']"));

File file = **new** File("tie.png");

//file = file.getAbsoluteFile();

String absolutePath = file.getAbsolutePath();

fileupload.sendKeys(absolutePath);

List <WebElement> tool = driver.findElements(By.*xpath*("//input[@name='tool']"));

**for**(**int** i=0;i<tool.size();i++) {

tool.get(i).click();

**if**(tool.get(i).isSelected()) {

System.***out***.println("Check Box Selected :: "+tool.get(i).getAttribute("value"));

}

}

WebElement web = driver.findElement(By.*xpath*("//select[@id='continents']"));

Select continents = **new** Select(web);

List<WebElement> cont = driver.findElements(By.*xpath*("//select[@id = 'continents']/option"));

**for** (**int** i=0;i<cont.size();i++) {

continents.selectByIndex(i);

System.***out***.println("Continents ::"+cont.get(i).getText());

Thread.*sleep*(2000);

}

WebElement web1 = driver.findElement(By.*xpath*("//select[@id='selenium\_commands']"));

Select selenium\_cmd = **new** Select(web1);

List<WebElement> cont\_cmd = driver.findElements(By.*xpath*("//select[@id='selenium\_commands']/option"));

**for** (**int** i=0;i<cont\_cmd.size();i++) {

**if**(i>0)

selenium\_cmd.deselectByIndex(i-1);

selenium\_cmd.selectByIndex(i);

//System.out.println("All Selected Displayed::" +selenium\_cmd.getAllSelectedOptions());

System.***out***.println("Slenium commands::"+cont\_cmd.get(i).getText());

Thread.*sleep*(2000);

}

WebElement btn = driver.findElement(By.*xpath*("//button[@id='submit']"));

btn.submit();

//select[@id = 'selenium\_commands']/option

/\*continents.selectByValue("Asia");

Thread.sleep(2000);

continents.selectByValue("Europe");

Thread.sleep(2000);

continents.selectByValue("Africa");

Thread.sleep(2000);

continents.selectByValue("Australia");

Thread.sleep(2000);

continents.selectByValue("South America");

Thread.sleep(2000);

continents.selectByValue("North America");

Thread.sleep(2000);

continents.selectByValue("Antartia");

\*/

Thread.*sleep*(7000);

driver.quit();

}

}

**TestNGBeforeAfter**

**package** TestNGBeforeAfter;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.AfterSuite;

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Test;

**public** **class** TestNGBeforeAfter {

@Test

**public** **void** testCase1() {

System.***out***.println("This is Test Case 1 in TestBeforeAfter Class");

}

@Test

**public** **void** testCase2() {

System.***out***.println("This is Test Case 2 in TestBeforeAfter Class");

}

@Test

**public** **void** testCase3() {

System.***out***.println("This is Test Case 3 in TestBeforeAfter Class");

}

@BeforeSuite

**public** **void** beforeSuite() {

System.***out***.println("Before Suite method");

}

@AfterSuite

**public** **void** afterSuite() {

System.***out***.println("After Suite method");

}

@BeforeTest

**public** **void** beforeTest() {

System.***out***.println("Before Test method");

}

@AfterTest

**public** **void** afterTest() {

System.***out***.println("After Test method");

}

@BeforeClass

**public** **void** beforeClass() {

System.***out***.println("Before Class method");

}

@AfterClass

**public** **void** afterClass() {

System.***out***.println("After Class method");

}

@BeforeMethod

**public** **void** beforeMethod() {

System.***out***.println("Before Method");

}

@AfterMethod

**public** **void** afterMethod() {

System.***out***.println("After Method");

}

}

<suite name=*"Parameter test Suite"* verbose=*"1"*>

<test name=*"Parameter Test one"*>

<classes>

<class name=*"TestNGBeforeAfter.TestNGBeforeAfter"*>

<methods>

<include name=*"testCase1"* />

</methods>

</class>

</classes>

</test>

<test name=*"Parameter Test one 1"*>

<classes>

<class name=*"TestNGBeforeAfter.TestNGBeforeAfter"*>

<methods>

<include name=*"testCase2"* />

</methods>

</class>

</classes>

</test>

<test name=*"Parameter Test one 3"*>

<classes>

<class name=*"TestNGBeforeAfter.TestNGBeforeAfter"*>

<methods>

<include name=*"testCase3"* />

</methods>

</class>

</classes>

</test>

</suite>

**TestNGdataprovider**

**package** com.TestNGdataprovider;

**import** org.testng.annotations.DataProvider;

**import** org.testng.annotations.Test;

**public** **class** TestNGdataprovider {

@DataProvider(name = "data1")

**public** Object[][] dataProviderMethod() {

**return** **new** Object[][] { { "data one" }, { "data two" },{ "data three" },{ "data four" } };

}

@Test(dataProvider = "data1")

**public** **void** testMethod(String data) {

System.***out***.println("Data is: " + data);

}

}

**ackage** com.TestNGdataprovider;

**import** org.testng.annotations.DataProvider;

**public** **class** TestNGdataproviderbaseclass {

@DataProvider(name = "data-provider")

**public** **static** Object[][] dataProviderMethod()

{

**return** **new** Object[][] { { "Base" }, { "Derived" } };

}

}

**package** com.TestNGdataprovider;

**import** org.testng.annotations.Test;

**public** **class** TestNGdataproviderderivedclass {

@Test(dataProvider = "data-provider", dataProviderClass = TestNGdataproviderbaseclass.**class**)

**public** **void** testMethod(String data)

{

System.***out***.println("Data is: " + data);

}

}

<suite name=*"Parameter test Suite"* verbose=*"1"*>

<test name=*"Parameter Test one"*>

<classes>

<class name=*"com.TestNGdataprovider.TestNGdataprovider"*>

<methods>

<include name=*"testMethod"* />

</methods>

</class>

</classes>

</test>

<test name=*"Base and Derived Class"*>

<classes>

<class name=*"com.TestNGdataprovider.TestNGdataproviderderivedclass"*>

<methods>

<include name=*"testMethod"* />

</methods>

</class>

</classes>

</test>

</suite>

**TestNGDemo**

**package** com.testng.demo;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.AfterSuite;

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Test;

**public** **class** TestNGAnno {

@Test

**public** **void** testCase1() {

System.***out***.println("Test Execution 1");

}

@Test

**public** **void** testCase2() {

System.***out***.println("Test Execution 2");

}

@BeforeSuite

**public** **void** beforeSuite() {

System.***out***.println("Before Suite method");

}

@AfterSuite

**public** **void** afterSuite() {

System.***out***.println("After Suite method");

}

@BeforeTest

**public** **void** beforeTest() {

System.***out***.println("Before Test method");

}

@AfterTest

**public** **void** afterTest() {

System.***out***.println("After Test method");

}

@BeforeClass

**public** **void** beforeClass() {

System.***out***.println("Before Class method");

}

@AfterClass

**public** **void** afterClass() {

System.***out***.println("After Class method");

}

@BeforeMethod

**public** **void** beforeMethod() {

System.***out***.println("Before Method");

}

@AfterMethod

**public** **void** afterMethod() {

System.***out***.println("After Method");

}

}

<suite name=*"Parameter test Suite"* verbose=*"1"*>

<!-- This parameter will be passed to every test in this suite -->

<!-- <parameter name="suite-param" value="suite level parameter" /> -->

<test name=*"TC\_ID1: TestNg Annotation Before And After"*>

<classes>

<class name=*"com.testng.demo.TestNGAnno"*>

<methods>

<include name=*"testCase1"* />

</methods>

</class>

</classes>

</test>

<test name=*"TC\_ID2: TestNg Annotation Before And After"*>

<classes>

<class name=*"com.testng.demo.TestNGAnno"*>

<methods>

<include name=*"testCase2"* />

</methods>

</class>

</classes>

</test>

</suite>

<suite name=*"Parameter test Suite"* verbose=*"1"*>

<!-- This parameter will be passed to every test in this suite -->

<!-- <parameter name="suite-param" value="suite level parameter" /> -->

<test name=*"TC\_ID1: TestNg Annotation Before And After"*>

<classes>

<class name=*"com.testng.demo.TestNGAnno"*>

<methods>

<include name=*"testCase"* />

</methods>

</class>

</classes>

</test>

</suite>

**package** com.TestNGDemo2;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.AfterSuite;

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Test;

**public** **class** TestNGDemo2 {

@Test

**public** **void** testCase() {

}

@BeforeSuite

**public** **void** beforeSuite() {

System.***out***.println("Before Suite method Demo 2");

}

@AfterSuite

**public** **void** afterSuite() {

System.***out***.println("After Suite method Demo 2");

}

@BeforeTest

**public** **void** beforeTest() {

System.***out***.println("Before Test method Demo 2");

}

@AfterTest

**public** **void** afterTest() {

System.***out***.println("After Test method Demo 2");

}

@BeforeClass

**public** **void** beforeClass() {

System.***out***.println("Before Class method Demo 2");

}

@AfterClass

**public** **void** afterClass() {

System.***out***.println("After Class method Demo 2");

}

@BeforeMethod

**public** **void** beforeMethod() {

System.***out***.println("Before Method Demo 2");

}

@AfterMethod

**public** **void** afterMethod() {

System.***out***.println("After Method Demo 2");

}

}

<suite name=*"Parameter test Suite"* verbose=*"1"*>

<test name=*"Parameter Test one"*>

<classes>

<class name=*"com.TestNGDemo2.TestNGDemo2"*>

<methods>

<include name=*"testCase"* />

</methods>

</class>

</classes>

</test>

</suite>

**package** TestNGDemo3;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.AfterSuite;

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Test;

**public** **class** TestNGDemo3 {

@Test

**public** **void** testCase1() {

System.***out***.println("TestCase 1 is Executed");

}

@Test

**public** **void** testCase2() {

System.***out***.println("TestCase 2 is Executed");

}

@BeforeSuite

**public** **void** beforeSuite() {

System.***out***.println("Before Suite method Demo 3");

}

@AfterSuite

**public** **void** afterSuite() {

System.***out***.println("After Suite method Demo 3");

}

@BeforeTest

**public** **void** beforeTest() {

System.***out***.println("Before Test method Demo 3");

}

@AfterTest

**public** **void** afterTest() {

System.***out***.println("After Test method Demo 3");

}

@BeforeClass

**public** **void** beforeClass() {

System.***out***.println("Before Class method Demo 3");

}

@AfterClass

**public** **void** afterClass() {

System.***out***.println("After Class method Demo 3");

}

@BeforeMethod

**public** **void** beforeMethod() {

System.***out***.println("Before Method Demo 3");

}

@AfterMethod

**public** **void** afterMethod() {

System.***out***.println("After Method Demo 3");

}

}

<suite name=*"Parameter test Suite"* verbose=*"1"*>

<test name=*"Parameter Test one"*>

<classes>

<class name=*"TestNGDemo3.TestNGDemo3"*>

<methods>

<include name=*"testCase1"* />

</methods>

</class>

</classes>

</test>

<test name=*"Parameter Test one 1"*>

<classes>

<class name=*"TestNGDemo3.TestNGDemo3"*>

<methods>

<include name=*"testCase2"* />

</methods>

</class>

</classes>

</test>

</suite>

**TestNGDependent**

**package** com.TestNGdependentTests;

**import** org.testng.annotations.Test;

**public** **class** DependentTestExamples **extends** TestNGdependentTests {

@Test(dependsOnMethods = { "testOne" })

**public** **void** testThree() {

System.***out***.println("Test three method in Inherited test");

}

@Test

**public** **void** testFour() {

System.***out***.println("Test four method in Inherited test");

}

}

**package** com.TestNGdependentTests;

**import** org.testng.annotations.Test;

**import** org.testng.annotations.Test;

**import** org.testng.annotations.Test;

**public** **class** TestNGdependentTests {

@Test(dependsOnMethods = { "testTwo" })

**public** **void** testOne() {

System.***out***.println("Test method depends on method Two and Four");

}

@Test /\*(dependsOnMethods = {"testFour"})\*/

**public** **void** testTwo() {

System.***out***.println("Test method Two Execute second");

}

@Test

**public** **void** testThree() {

System.***out***.println("Test method three");

}

@Test

**public** **void** testFour() {

System.***out***.println("Test method Four Execute first");

}

@Test(dependsOnMethods = {"testSix" , "testSeven"})

**public** **void** testFive() {

System.***out***.println("Test method five Execute last");

}

@Test

**public** **void** testSix() {

System.***out***.println("Test method six Execute first");

}

@Test

**public** **void** testSeven() {

System.***out***.println("Test method seven Execute first");

}

@Test(dependsOnGroups = { "test-group" })

**public** **void** groupTestOne() {

System.***out***.println("Group Test method one");

}

@Test(groups = { "test-group" })

**public** **void** groupTestTwo() {

System.***out***.println("Group test method two");

}

@Test(groups = { "test-group" })

**public** **void** groupTestThree() {

System.***out***.println("Group Test method three");

}

}

<suite name=*"Parameter test Suite"* verbose=*"1"*>

<test name=*"One Method depends on hierarchy"*>

<classes>

<class name=*"com.TestNGdependentTests.TestNGdependentTests"*>

<methods>

<include name=*"testOne"* />

<include name=*"testTwo"* />

<include name=*"testFour"* />

</methods>

</class>

</classes>

</test>

<test name=*"Method depends on hierarchy"*>

<classes>

<class name=*"com.TestNGdependentTests.TestNGdependentTests"*>

<methods>

<include name=*"testOne"* />

<include name=*"testTwo"* />

<include name=*"testFour"* />

<include name=*"testThree"* />

</methods>

</class>

</classes>

</test>

<test name=*"Parameter Test one 2"*>

<classes>

<class name=*"com.TestNGdependentTests.TestNGdependentTests"*>

<methods>

<include name=*"testFive"* />

<include name=*"testSix"* />

<include name=*"testSeven"* />

</methods>

</class>

</classes>

</test>

<test name=*"Parameter Test one 3"*>

<classes>

<class name=*"com.TestNGdependentTests.DependentTestExamples"*>

<methods>

<include name=*"testThree"* />

<include name=*"testFour"* />

</methods>

</class>

<class name=*"com.TestNGdependentTests.TestNGdependentTests"*>

<methods>

<include name=*"testOne"* />

<include name=*"testTwo"* />

<include name=*"testFour"* />

</methods>

</class>

</classes>

</test>

<test name=*"Group Test Dependencies"*>

<groups>

<run>

<include name=*"test-group"* />

</run>

</groups>

<classes>

<class name=*"com.TestNGdependentTests.TestNGdependentTests"* />

<methods>

<include name=*"groupTestOne"* />

<include name=*"groupTestTwo"* />

<include name=*"groupTestThree"* />

</methods>

</classes>

</test>

</suite>

**TestNGenabledisable**

**package** com.TestNGenabledisable;

**import** org.testng.annotations.Test;

**public** **class** TestNGenabledisable {

@Test(enabled = **true**)

**public** **void** testMethodOne() {

System.***out***.println("Test method one.");

}

@Test(enabled = **false**)

**public** **void** testMethodTwo() {

System.***out***.println("Test method two.");

}

@Test

**public** **void** testMethodThree() {

System.***out***.println("Test method three.");

}

}

<suite name=*"Parameter test Suite"* verbose=*"1"*>

<test name=*"Parameter Test one"*>

<classes>

<class name=*"com.TestNGenabledisable.TestNGenabledisable"*>

<!-- <methods>

<include name="testMethodTwo" />

</methods> -->

</class>

</classes>

</test>

</suite>

**TestNGgroups**

**package** com.testNGgroups;

**import** org.testng.annotations.Test;

@Test(groups = {"Default Group"})

**public** **class** TestNGgroups {

@Test(groups = { "test-group car" })

**public** **void** testMethodOne() {

System.***out***.println("Test method one belonging to group car.");

}

@Test

**public** **void** testMethodTwo() {

System.***out***.println("Test method two not belonging to group.");

}

@Test(groups = { "test-group car" })

**public** **void** testMethodThree() {

System.***out***.println("Test method three belonging to group car.");

}

@Test(groups = { "test-group bus" })

**public** **void** testMethodFour() {

System.***out***.println("Test method four belonging to group bus.");

}

@Test

**public** **void** testMethodFive() {

System.***out***.println("Test method five not belonging to group.");

}

@Test(groups = { "test-group bus" })

**public** **void** testMethodSix() {

System.***out***.println("Test method six belonging to group bus.");

}

@Test(groups = { "test-group bus","test-group car" })

**public** **void** testMethodSeven() {

System.***out***.println("Test method Seven belonging to group car and bus.");

}

@Test(groups = { "testgroupmotor" })

**public** **void** testMethodEight() {

System.***out***.println("Test method eight belonging to group motor.");

}

@Test(groups = { "testgrouptrain" })

**public** **void** testMethoodNine() {

System.***out***.println("Test method nine belonging to group train.");

}

}

<suite name=*"Time test Suite"* verbose=*"1"*>

<test name=*"Group Test"*>

<groups>

<run>

<include name=*"Default Group"* />

</run>

</groups>

<classes>

<class name=*"com.testNGgroups.TestNGgroups"* />

</classes>

</test>

</suite>

<suite name=*"Time test Suite"* verbose=*"1"*>

<test name=*"Group Test"*>

<groups>

<run>

<include name=*"test-group bus"* />

<exclude name=*"test-group car"* />

</run>

</groups>

<classes>

<class name=*"com.testNGgroups.TestNGgroups"* />

</classes>

</test>

<test name=*"Group Test 1"*>

<groups>

<run>

<exclude name=*"test-group bus"* />

<include name=*"test-group car"* />

</run>

</groups>

<classes>

<class name=*"com.testNGgroups.TestNGgroups"* />

</classes>

</test>

</suite>

<suite name=*"Time test Suite"* verbose=*"1"*>

<test name=*"Group Test"*>

<groups>

<run>

<include name=*"test-group car"* />

</run>

</groups>

<classes>

<class name=*"com.testNGgroups.TestNGgroups"* />

</classes>

</test>

<test name=*"Group Test 1"*>

<classes>

<class name=*"com.testNGgroups.TestNGgroups"* />

</classes>

</test>

</suite>

<suite name=*"Time test Suite"* verbose=*"1"*>

<test name=*"Group Test"*>

<groups>

<run>

<include name=*"testgroupm.\*"* />

<exclude name=*".\*rain"* />

</run>

</groups>

<classes>

<class name=*"com.testNGgroups.TestNGgroups"* />

</classes>

</test>

<test name=*"Group Test 1"*>

<groups>

<run>

<exclude name=*"testgroupm.\*"* />

<include name=*"test.\*"* />

</run>

</groups>

<classes>

<class name=*"com.testNGgroups.TestNGgroups"* />

</classes>

</test>

</suite>

<suite name=*"Group of group Suite"* verbose=*"1"*>

<test name=*"Group of group Test"*>

<groups>

<define name=*"include-group"*>

<include name=*"testgroupmotor"* />

<include name=*"testgrouptrain"* />

</define>

<define name=*"exclude-group"*>

<include name=*"test-group car"* />

<include name=*"test-group bus"* />

</define>

<run>

<include name=*"include-group"* />

<exclude name=*"exclude-group"* />

<include name=*"Default Group"* />

</run>

</groups>

<classes>

<class name=*"com.testNGgroups.TestNGgroups"* />

</classes>

</test>

</suite>

<suite name=*"Time test Suite"* verbose=*"1"*>

<test name=*"Group Test"*>

<groups>

<run>

<include name=*"test-group bus"* />

</run>

</groups>

<classes>

<class name=*"com.testNGgroups.TestNGgroups"* />

</classes>

</test>

</suite>

**TestNGparameter**

**package** com.TestNGparameter;

**import** org.testng.annotations.Optional;

**import** org.testng.annotations.Parameters;

**import** org.testng.annotations.Test;

**public** **class** TestNGparameter {

@Parameters({ "suite-param" })

@Test

**public** **void** prameterTestOne(String param) {

System.***out***.println("Test one suite param is: " + param);

}

@Parameters({ "test-two-param" })

@Test

**public** **void** prameterTestTwo(@Optional("test-two-param")String param) {

System.***out***.println("Test two param is: " + param);

}

@Parameters({ "suite-param", "test-two-param" })

@Test

**public** **void** prameterTestThree(String param, String paramTwo) {

System.***out***.println("Test three suite param is: " + param);

System.***out***.println("Test three param is: " + paramTwo);

}

}

<suite name=*"Parameter test Suite"* verbose=*"1"*>

<!-- This parameter will be passed to every test in this suite -->

<test name=*"Parameter Test one"*>

<parameter name=*"suite-param"* value=*"Yor are passing parameter value"* />

<classes>

<class name=*"com.TestNGparameter.TestNGparameter"*>

<methods>

<include name=*"prameterTestOne"* />

</methods>

</class>

</classes>

</test>

<!-- This parameter will be passed to every test in this suite -->

<test name=*"Parameter Test two"*>

<parameter name=*"test-two-param"* value=*"Yor are passing parameter value in method two."* />

<classes>

<class name=*"com.TestNGparameter.TestNGparameter"*>

<methods>

<include name=*"prameterTestTwo"* />

</methods>

</class>

</classes>

</test>

<!-- This parameter will be passed to every test in this suite -->

<parameter name=*"suite-param"* value=*"Yor are passing parameter value together!!!"* />

<parameter name=*"test-two-param"* value=*"Yor are passing parameter value together!!!"* />

<test name=*"Parameter Test mix"*>

<classes>

<class name=*"com.TestNGparameter.TestNGparameter"*>

<methods>

<include name=*"prameterTestOne"* />

<include name=*"prameterTestTwo"* />

<include name=*"prameterTestThree"* />

</methods>

</class>

</classes>

</test>

</suite>

**TestNGparellelTest**

**package** com.TestNGparellelTest;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.Test;

**public** **class** TestNGparellelTest {

@BeforeMethod

**public** **void** beforeMethod() {

**long** id = Thread.*currentThread*().getId();

System.***out***.println("Before test-method. Thread id is: " + id);

}

@Test

**public** **void** testMethodsOne() {

**long** id = Thread.*currentThread*().getId();

System.***out***.println("Simple test-method One. Thread id is: " + id);

}

@Test

**public** **void** testMethodsTwo() {

**long** id = Thread.*currentThread*().getId();

System.***out***.println("Simple test-method Two. Thread id is: " + id);

}

@Test

**public** **void** testMethodsThree() {

**long** id = Thread.*currentThread*().getId();

System.***out***.println("Simple test-method Three. Thread id is: " + id);

}

@AfterMethod

**public** **void** afterMethod() {

**long** id = Thread.*currentThread*().getId();

System.***out***.println("After test-method. Thread id is: " + id);

}

}

<suite name=*"Test-method Suite"* parallel=*"methods"* thread-count=*"3"* >

<test name=*"Test-method test"* group-by-instances=*"true"*>

<classes>

<class name=*"com.TestNGparellelTest.TestNGparellelTest"* />

</classes>

</test>

<test name=*"Test-method test 1"* group-by-instances=*"true"*>

<classes>

<class name=*"com.TestNGparellelTest.TestNGparellelTest"* />

</classes>

</test>

</suite>

**TestNGSimple**

**package** com.TestNGsimple;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.Test;

**public** **class** TestNGSimple {

@Test

**public** **void** display() {

System.***out***.println("You have executed the disply method");

}

@Test

**public** **void** calculation() {

**int** i=30;

System.***out***.println("Addition of 10 and 20 = " +i);

}

@BeforeMethod

**public** **void** beforeMethod() {

System.***out***.println("Before method executed");

}

@AfterMethod

**public** **void** afterMethod() {

System.***out***.println("After method executed");

}

@BeforeClass

**public** **void** beforeClass() {

System.***out***.println("Before class method executed");

}

@AfterClass

**public** **void** afterClass() {

System.***out***.println("After class method executed");

}

}

<suite name=*"TestNG Simple"* verbose=*"1"* >

<test name=*"Regression one"*>

<classes>

<class name=*"com.TestNGsimple.TestNGSimple"*/>

<methods>

<include name = *"display"* />

</methods>

</classes>

</test>

<test name=*"Regression two"*>

<classes>

<class name=*"com.TestNGsimple.TestNGSimple"*/>

<methods>

<include name = *"calculation"* />

</methods>

</classes>

</test>

</suite>

**waits\_commands\_demo**

**package** waits\_commands\_demo;

**import** java.util.Set;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.Alert;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.WebDriverWait;

**public** **class** waits\_commands\_demo {

**public** **static** **void** main(String args[])**throws** InterruptedException {

String path = "chromedriver.exe";

System.*setProperty*("webdriver.chrome.driver", path);

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***);

driver.get("http://toolsqa.wpengine.com/automation-practice-switch-windows/");

Thread.*sleep*(4000);

String handle= driver.getWindowHandle();

System.***out***.println("Current Window Handle::" +handle);

WebDriverWait wait = **new** WebDriverWait(driver, 10);

WebElement New\_brow\_win = wait.until(ExpectedConditions.*elementToBeClickable*(By.*xpath*("//button[text()='New Browser Window']")));

New\_brow\_win.click();

WebElement New\_msg\_win = wait.until(ExpectedConditions.*elementToBeClickable*(By.*xpath*("//button[text()='New Message Window']")));

New\_msg\_win.click();

WebElement new\_brow\_tab = wait.until(ExpectedConditions.*elementToBeClickable*(By.*xpath*("//button[text()='New Browser Tab']")));

new\_brow\_tab.click();

Set<String> handles = driver.getWindowHandles();

System.***out***.println(handles);

// Pass a window handle to the other window

**int** i = 0;

String parent\_window="";

**for** (String handle1 : handles) {

System.***out***.println(handle1);

**if**(i==0) {

parent\_window = handle1;

i++;

}

driver.switchTo().window(handle1);

driver.manage().window().maximize();

Thread.*sleep*(15000);

}

driver.switchTo().window(parent\_window);

Thread.*sleep*(5000);

WebElement alert = driver.findElement(By.*xpath*("//button[text()='Alert Box']"));

alert.click();

Thread.*sleep*(5000);

Alert myalert = driver.switchTo().alert();

myalert.accept();

Thread.*sleep*(2000);

WebElement alert\_timing = driver.findElement(By.*xpath*("//button[text()='Timing Alert']"));

alert\_timing.click();

Thread.*sleep*(5000);

Alert mytiming = driver.switchTo().alert();

mytiming.accept();

Thread.*sleep*(15000);

driver.close(); // closing the pop up Window

System.***out***.println("All Windows Closed");

driver.quit(); // Closing the original window

}

}

**Pom file**

<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>projectvp</groupId>

<artifactId>projectvp</artifactId>

<version>0.0.1-SNAPSHOT</version>

<build>

<!-- <sourceDirectory>src</sourceDirectory> -->

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.3</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.19.1</version>

<configuration>

<suiteXmlFiles>

<!-- TestNG suite XML files -->

<!-- <suiteXmlFile>Build/index.xml</suiteXmlFile>

<suiteXmlFile>Build/index1.xml</suiteXmlFile> -->

<suiteXmlFile>Build/index2.xml</suiteXmlFile>

</suiteXmlFiles>

<reportsDirectory>test-output</reportsDirectory>

<properties>

<property>

<name>usedefaultlisteners</name>

<value>true</value>

</property>

<property>

<name>reporter</name>

<value>listenReport.Reporter</value>

</property>

</properties>

</configuration>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.13.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.8.8</version>

<!-- <scope>test</scope> -->

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-io -->

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-io</artifactId>

<version>1.3.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.7</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.13</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>3.13</version>

</dependency>

</dependencies>

</project>

**POM file**

<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>esignature</groupId>

<artifactId>esignature</artifactId>

<version>0.0.1-SNAPSHOT</version>

<build>

<!-- <sourceDirectory>src</sourceDirectory> -->

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.3</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.19.1</version>

<configuration>

<suiteXmlFiles>

<!-- TestNG suite XML files -->

<suiteXmlFile>Build/esignindex.xml</suiteXmlFile>

<!-- <suiteXmlFile>Build/index.xml</suiteXmlFile>

<suiteXmlFile>Build/index1.xml</suiteXmlFile>

<suiteXmlFile>Build/index2.xml</suiteXmlFile> -->

</suiteXmlFiles>

<reportsDirectory>test-output</reportsDirectory>

<properties>

<property>

<name>usedefaultlisteners</name>

<value>true</value>

</property>

<property>

<name>reporter</name>

<value>listenReport.Reporter</value>

</property>

</properties>

</configuration>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.13.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.8.8</version>

<!-- <scope>test</scope> -->

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-io -->

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-io</artifactId>

<version>1.3.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.7</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.13</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>3.13</version>

</dependency>

</dependencies>

</project>