**The features are as follows:**

**#1) Capture screenshot of specific web element:**

|  |
| --- |
| WebElement logo=driver.Findelement (By.xpath(“//div[@id=’divLogo’]//img”));  File file=logo.getScreenshotAs(OutputType.FILE);  File destFile =new File(“logo.png”);  FileUtils.copyFile(file,destfile); |

**#2) Open the new tab on the browser:**

|  |
| --- |
| driver.get(https://www.google.com/);  driver.switchTo().newWindow(WindowType.TAB);  driver.navigate().to(https://www.crmpro.com/); |

**#4) Object Location:**

|  |
| --- |
| WebElement logo1=driver.Findelement(By.xpath(“//div[@id=’divLogo’]//img”));  System.out.println(“Height:” +logo.getRect().getDimension().getHeight());  System.out.println(“Height:” +logo.getRect().getDimension().getWidth());  System.out.println(“X Location: “ +Logo.getRect().getX());  System.out.println(“Y Location: “ +Logo.getRect().getY()); |

**#5) Relative Locators:**

These are also known as Friendly Locators, and this functionality is being added to find out the element which is present nearby to other web element or, we can say that it can find the web elements based on GUI location.

**There are five locators added in Selenium 4:**

* **below():** Web element located below for the specified element.
* **toLeftOf() :** Target web element which is present to the left of specified element.
* **toRightOf():** Target web element which is presented to the right of a specified element.
* **above():** Web element located above for the specified element.
* **near() :** Target web element which is away(approx. 50 pixels) from the specified element.

**Note:** All the above relative locators method support “withTagName” method.

**The below example is for the toLeftof() and below() locators:**

|  |
| --- |
| WebElement book;  book = driver.Findelement(RelativeLocators.withTagName(“li”).toLeftOf(By.id(“pid1”))  .below(By.id(“pid2”)));  String id1=book.getAttribute (“id1”); |

**The below example is for the toRightOf() and above() locators:**

|  |
| --- |
| WebElement book1;  book1 = driver.Findelement(RelativeLocators.withTagName(“li”).toRightOf(By.id(“pid1”))  .above(By.id(“pid2”)));  String id2=book1.getAttribute (“id2”); |

**#6) Chrome Dev tools:**

In the new version of Selenium, they have made some internal changes in the API. Earlier in Selenium 3, the Chrome driver extends directly to the Remote Web Driver class. But now in Selenium 4, Chrome driver class extends to Chromium Driver. Chromium Driver class has some predefined methods to access the dev tool.

**Note:** Chromium Driver extends the Remote Web driver class.

**By using the API, we can perform the following operations:**

* Enable Network Offline
* Enable Network Online
* Get Console Logs
* Load Insure Web Site

String str = "beautiful beach";

char[] chararr = str.toCharArray();

for (int i = 0; i < str.length(); i++) {

for (int j = i + 1; j < str.length(); j++) {

if (chararr[i] == chararr[j]) {

System.out.print(chararr[j] + " ");

break;

char[] characters = origString.toCharArray();

for( int i = characters.length - 1 ; i >= 0 ; i-- ) {

reverseString = reverseString + characters[i];

}

//Check palindrome string

if (origString.equals(reverseString)) {

System.out.println("String is a palindrome.");

} else {

System.out.println("String is not a palindrome.");

}

String string = "Once in a blue moon";

char ch = '-';

//Replace space with specific character ch

string = string.replace(' ', ch);