

# Selenium Keyboard events

In Selenium with Java, the **Keys** class (part of the `org.openqa.selenium.Keys` package) is used to simulate **keyboard events** such as pressing **Enter**, **Tab**, **Arrow keys**, **Control (Ctrl)**, etc. These are especially useful when:

- Submitting forms with **Enter**
- Navigating dropdowns with **Arrow** keys
- Copying/Pasting text using **Ctrl + C** / **Ctrl + V**
- Performing keyboard shortcuts (like **Ctrl + A**, **Ctrl + X**)

We usually use **Keys** with the `sendKeys()` method.

## Setup:

Make sure you import:

```
java

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
```

## Example 1: Submit a form using Enter key

```
java

public class KeyboardEventExample {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com");

        // Locate search box and type query
        WebElement searchBox = driver.findElement(By.name("q"));
        searchBox.sendKeys("Selenium WebDriver");

        // Press Enter to search
        searchBox.sendKeys(Keys.RETURN); // Or Keys.ENTER

        driver.quit();
    }
}
```

## Explanation:

- `Keys.RETURN` simulates the pressing of the **Enter key**.
- The browser navigates to the search results page automatically.

## ✓ Example 2: Copy-Paste using Ctrl + A, Ctrl + C, Ctrl + V

```
java

public class KeyboardCopyPaste {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com");

        WebElement searchBox = driver.findElement(By.name("q"));

        // Type text
        searchBox.sendKeys("Selenium Automation");

        // Select all text: CTRL + A
        searchBox.sendKeys(Keys.chord(Keys.CONTROL, "a"));

        // Copy text: CTRL + C
        searchBox.sendKeys(Keys.chord(Keys.CONTROL, "c"));

        // Clear box and paste: CTRL + V
        searchBox.clear();
        searchBox.sendKeys(Keys.chord(Keys.CONTROL, "v"));

        driver.quit();
    }
}
```

### 💡 Explanation:

- `Keys.chord(Keys.CONTROL, "a")` sends Ctrl + A.
- You can simulate any keyboard combo using `Keys.chord()`.

## ✓ Example 3: Tab navigation and submit form

```
java

public class KeyboardTabNavigation {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://example.com/login"); // Replace with actual login page

        WebElement username = driver.findElement(By.id("username"));
        username.sendKeys("admin");

        // Press TAB to switch to password field
        username.sendKeys(Keys.TAB);

        // Now send password (focus is on password field)
        WebElement body = driver.findElement(By.tagName("body"));
        body.sendKeys("password123");

        // Press TAB to go to Login button, then press ENTER
        body.sendKeys(Keys.TAB);
        body.sendKeys(Keys.ENTER);

        driver.quit();
    }
}
```

## Commonly Used Keys in Selenium:

Key	Description
<code>Keys . ENTER</code>	Enter key
<code>Keys . RETURN</code>	Same as Enter
<code>Keys . TAB</code>	Move focus to next
<code>Keys . ESCAPE</code>	Escape key
<code>Keys . ARROW_DOWN</code>	Down arrow key
<code>Keys . ARROW_UP</code>	Up arrow key
<code>Keys . CONTROL</code> + other key	Combo (Ctrl+C, Ctrl+V etc.)
<code>Keys . SHIFT</code>	Shift key
<code>Keys . BACK_SPACE</code>	Backspace key
<code>Keys . DELETE</code>	Delete key

### Example 4: Keyboard Shortcut in Search Field

java

```
searchBox.sendKeys("test");
searchBox.sendKeys(Keys.chord(Keys.CONTROL, "a")); // Select all
searchBox.sendKeys(Keys.chord(Keys.CONTROL, "x")); // Cut
searchBox.sendKeys(Keys.chord(Keys.CONTROL, "v")); // Paste
```

### Summary

Concept	How It's Done
Basic key press	<code>sendKeys(Keys . ENTER)</code>
Combination keys	<code>sendKeys(Keys . chord(Keys . CONTROL, "c"))</code>
Navigation via keyboard	<code>Keys . TAB</code> , <code>Keys . ARROW_DOWN</code> , etc.
Clear and send new text	<code>element.clear(); element.sendKeys("...")</code>

Would you like a **real-time mini-project** like login form automation using keyboard events?