🚀 From Clicks to Inputs: Automating Real-World Web Elements with Selenium (Buttons & Text Boxes)

Hello Selenium Explorers!

Welcome back to the third post in my beginner-friendly Selenium series where we go from zero to test hero — one element at a time! 🚀

In the last post, we learned how to find web elements and work with links. Now, **Buttons and Text Boxes.**

Whether you're automating a login page, a checkout form, or a feedback module, you'll meet these two *everywhere*. So, let's learn how to tame them with Java and Selenium as our weapons, and Leaf Ground Practice Site as our training ground.

BUTTONS — Small Element, Big Power

Buttons do things. They **submit**, they **trigger**, they **navigate** — and they're everywhere.

Here's how to master them 👇

1. **Clicking a Button — The Most Basic, Yet Most Powerful**

```
WebElement clickBtn = driver.findElement(By.id("j_idt88:j_idt90"));
clickBtn.click();
```

That's it. You've just simulated a user action!

2. What Color Is That Button?

UI matters. If you’re testing design or dark mode themes, you’ll want to check this:

```
WebElement colorBtn = driver.findElement(By.id("j_idt88:j_idt96"));
String color = colorBtn.getCssValue("background-color");
System.out.println("Button color: " + color);
```

Helpful in testing visual consistency across environments or themes.

3. How Big Is It? (Size Matters in UI)

```
WebElement sizeBtn = driver.findElement(By.id("j_idt88:j_idt98"));

Dimension size = sizeBtn.getSize();

System.out.println("Width: " + size.getWidth());

System.out.println("Height: " + size.getHeight());
```

Use this for responsive testing or verifying consistent spacing.

4. Where’s That Button Positioned?

Sometimes, it’s not just about presence. It’s about placement.

```
Point location = sizeBtn.getLocation();
System.out.println("X: " + location.getX());
System.out.println("Y: " + location.getY());
```

Useful when checking UI alignment or pixel-perfect designs.

TEXT BOXES — Your Way to Enter the Web

If you’ve ever typed your name in a form — you’ve used a text box. As testers, you need to interact with them *just like real users*.

Let’s go! 💪

1.Entering Text — Your First Automation Superpower

```
WebElement nameField = driver.findElement(By.id("j_idt88:name"));

nameField.sendKeys("QA Ninja");
```

2.Is That Field Disabled?

Sometimes, a field may be *visible* but not *usable*. Let’s check:

```
WebElement disabledField = driver.findElement(By.id("j_idt88:j_idt93"));

System.out.println("Is enabled? " + disabledField.isEnabled());
```

Good for validating conditional form behavior (e.g., disabled fields for guests).

2.Retrieve the Value You Entered

Want to double-check what you typed or what's auto-filled?

```
WebElement emailField = driver.findElement(By.id("j_idt88:j_idt99"));

emailField.sendKeys("test@qa.com");

String enteredText = emailField.getAttribute("value");

System.out.println("Entered value: " + enteredText);
```

Mini Summary: What You Just Mastered

Component	Skills You Gained
Button	Click, get color, get size, get position
Text Box	Enter text, clear, check enabled state, get value

You've now unlocked two essential Selenium skills — ones you'll reuse in nearly *every* UI automation script you write.

What's Coming Next?

In the next chapter of this journey, we'll be working with:

- 🔽 Dropdowns
- ✅ Checkboxes
- 🔘 Radio Buttons

You’ll learn to interact with user choices — the true decision-makers in any form!