

# How can we interact with different kind of elements in Playwright

Playwright is a powerful tool that allows us to automate browser actions for testing web applications. In this guide, I covered how to interact with different types of elements in the DOM (Document Object Model), such as **input fields**, **radio buttons**, **checkboxes**, dropdowns, dialog boxes, frames, and more.

## What is an Action?

An **action** in Playwright is an operation you perform on a web element. Playwright provides several functions to interact with elements on a web page. These actions include filling input fields, checking radio buttons, selecting checkboxes, interacting with dropdowns, handling pop-ups, and more.

---

## Interacting with Different Web Elements

### 1. Input Box Field

The action we perform on an **input text field** is to **fill** it. In Playwright, the function to handle this is `fill()`.

#### DOM:

```
<input type="text" id="firstname">
```

#### Playwright Code Example:

```
import { test, expect } from '@playwright/test';

test('Input field', async ({ page }) => {
  await page.goto('https://example.com');

  // Add assertions to validate the input text field
  await expect(page.locator('//input[@id="firstname"]')).toBeVisible();
  await expect(page.locator('//input[@id="firstname"]')).toBeEmpty();
  await expect(page.locator('//input[@id="firstname"]')).toBeEditable();
  await expect(page.locator('//input[@id="firstname"]')).toBeEnabled();

  // Fill the input text field
  await page.fill('//input[@id="firstname"]', 'Yogesh');

});
```

### 2. Radio Button

The action we perform on a **radio button** is to **check** it. In Playwright, the function used to check a radio button is `check()`.

## DOM:

```
<input type="radio" id="male" value="option1">
<input type="radio" id="female" value="option2">
```

### Playwright Code Example:

```
import { test, expect } from '@playwright/test';

test('Radio button', async ({ page }) => {
  await page.goto('https://example.com');

  // Check the 'male' radio button
  await page.locator("//input[@value='option1']").check();

  // Verify the 'male' radio button is checked
  await expect(page.locator("//input[@value='option1']")).toBeChecked();
  await expect(page.locator("//input[@value='option1']")).isChecked();

  // Verify the 'female' radio button is not checked
  await expect(page.locator("//input[@value='option2']")).isChecked().toBeFalsy();
});
```

## 3. Checkboxes

Checkboxes can be handled either individually or as a group (multiple checkboxes).

### Single Checkbox

The action we perform on a **checkbox** is to **check** or **uncheck** it. In Playwright, you can use `check()` and `uncheck()` to handle checkboxes.

## DOM:

```
<input type="checkbox" id="monday">
<input type="checkbox" id="tuesday">
<!-- ... -->
<input type="checkbox" id="sunday">
```

### Playwright Code Example:

```
import { test, expect } from '@playwright/test';

test('Handle single checkbox', async ({ page }) => {
  await page.goto('https://example.com');

  // 1. Check the Monday checkbox
  await page.locator("//input[@id='monday' and type='checkbox']").check();

  // 2. Verify the Monday checkbox is checked
  await expect(page.locator("//input[@id='monday' and type='checkbox']")).toBeChecked();
```

```

await expect(page.locator("//input[@id='monday' and
type='checkbox']")).isChecked().toBeTruthy();

// 3. Check that Sunday is not checked
await expect(page.locator("//input[@id='monday' and
type='checkbox']")).isChecked().toBeFalsy();

// 4. Uncheck the Monday checkbox
await page.locator("//input[@id='monday' and
type='checkbox']").uncheck();
});

```

## Multiple Checkboxes

In case there are multiple checkboxes (e.g., Monday to Sunday), you can perform actions like selecting multiple checkboxes and then unselecting them.

### Playwright Code Example for Multiple Checkboxes:

```

import { test, expect } from '@playwright/test';

test('Handle multiple checkboxes', async ({ page }) => {
  await page.goto('https://example.com');

  // 1. Check Monday, Wednesday, and Sunday checkboxes
  const checkboxes = [
    page.locator("//input[@id='monday' and type='checkbox']"),
    page.locator("//input[@id='wednesday' and type='checkbox']"),
    page.locator("//input[@id='sunday' and type='checkbox']")
  ];

  for (const checkbox of checkboxes) {
    await checkbox.check();
  }

  // 2. Unselect Monday, Wednesday, and Sunday checkboxes
  for (const checkbox of checkboxes) {
    if (await checkbox.isChecked()) {
      await checkbox.uncheck();
    }
  }
});

```