

PART 4: ADVANCED INTERACTIONS & VALIDATIONS

# Playwright Mastery Interview Deep Dive

Questions 31–40: Handling complex UI elements and user behavior with modern automation practices.

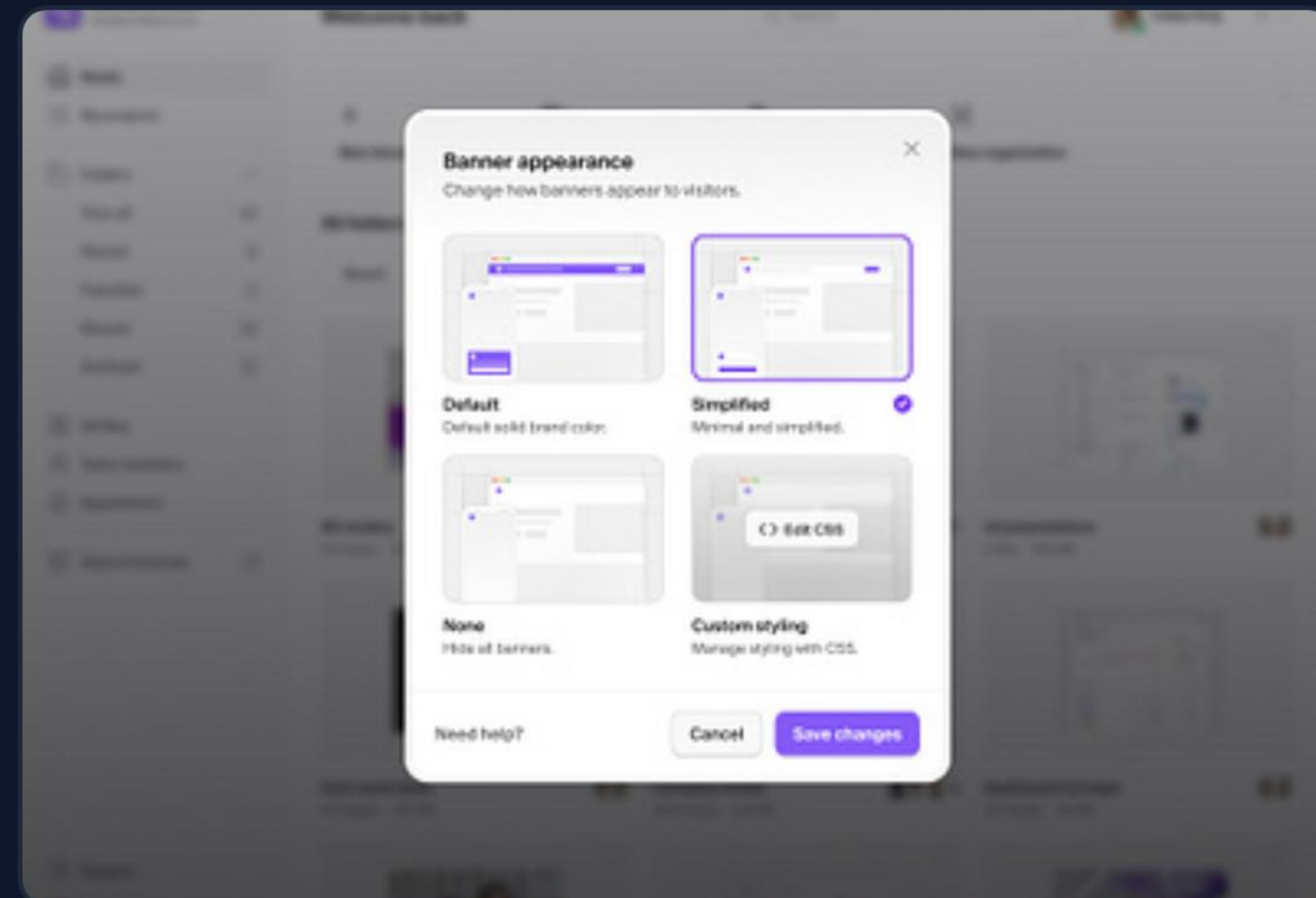
 Abhishesh Mishra |  6387782073



## 31. HANDLING CHECKBOXES

Playwright provides a native `check()` and `unchecked()` method that automatically waits for actionability.

```
// Check an element  
await page.locator('#terms').check();  
  
// Uncheck and verify state  
await page.locator('#newsletter').unchecked();  
await expect(locator).not.toBeChecked();
```





## 32. HOW TO UPLOAD A FILE?



### Single File

Upload one file directly to a file input.

```
await locator.setInputFiles('doc.pdf');
```



### Multiple Files

Upload an array of files in one go.

```
await locator.setInputFiles(['a.png',  
  'b.png']);
```



### Remove Files

Clear current file selections.

```
await locator.setInputFiles([]);
```

```
// Modern Playwright upload  
await page.getByLabel('Upload').setInputFiles('path/to/file.png');
```



## 33. HOW TO HOVER ON ELEMENTS?

Hovering is crucial for testing tooltips and hidden menus. Playwright simulates a real mouse move.

```
// Simple hover  
await page.locator('.menu-item').hover();  
  
// Hover with options  
await page.locator('#tooltip').hover({  
  force: true,  
  position: { x: 10, y: 10 }  
});
```





## 34. HOW TO DRAG AND DROP?

Playwright offers two ways: a high-level helper and a manual mouse simulation for complex scenarios.



### The Helper Way

Uses `dragTo()` to automate the whole move.

```
await source.dragTo(target);
```



### The Manual Way

Hover, Down, Move, Up sequence for precise control.

```
await page.mouse.move(...);
```

```
// Most common approach  
await page.locator('#draggable').dragTo(page.locator('#droppable'));
```



## 35. SCROLLING IN PLAYWRIGHT



### Auto-Scroll

Playwright scrolls to elements automatically before actions.



### Manual Scroll

Scroll an element into the viewport manually.

```
await loc.scrollIntoViewIfNeeded();
```



### Mouse Wheel

Scroll using coordinates via the mouse API.

```
await page.mouse.wheel(0, 500);
```

*Pro-tip: `scrollIntoViewIfNeeded()` is the most stable way to ensure visibility before taking screenshots.*

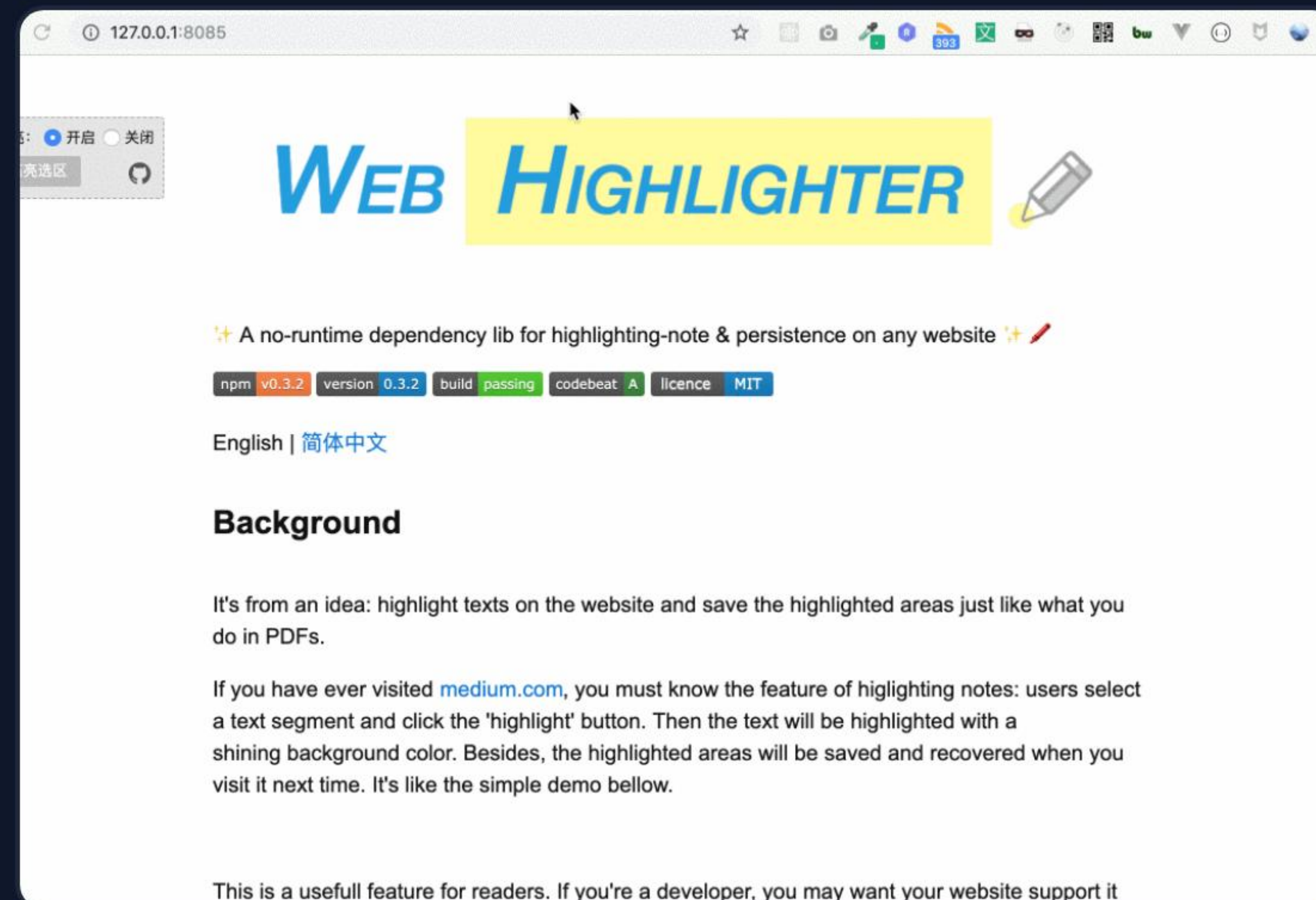


## 36. GETTING INNER TEXT

There is a difference between `innerText()` and `textContent()` in Playwright.

- ✓ **`innerText()`**: Respects CSS (only visible text).
- ✓ **`textContent()`**: Includes hidden text (faster).

```
const val = await
page.locator('h1').innerText();
console.log(val);
```





## 37. ASSERTING URL

Navigation Check

# URL

Use Web-First assertions to wait for navigation to complete. This is safer than checking `page.url()` manually.

```
// Exact URL  
await  
expect(page).toHaveURL('https://site.com/home');  
  
// Regex pattern  
await expect(page).toHaveURL(/.*dashboard/);
```



## 38. PRESSING KEYBOARD KEYS



### Single Key

Simulate single key presses like Enter or Escape.

```
await page.keyboard.press('Enter');
```



### Shortcuts

Handle modifier keys like Ctrl or Shift.

```
await page.keyboard.press('Control+A');
```



### Down/Up

Hold a key down while performing other actions.

```
await page.keyboard.down('Shift');
```

Playwright supports all standard key names: F1-F12, Backspace, Delete, etc.

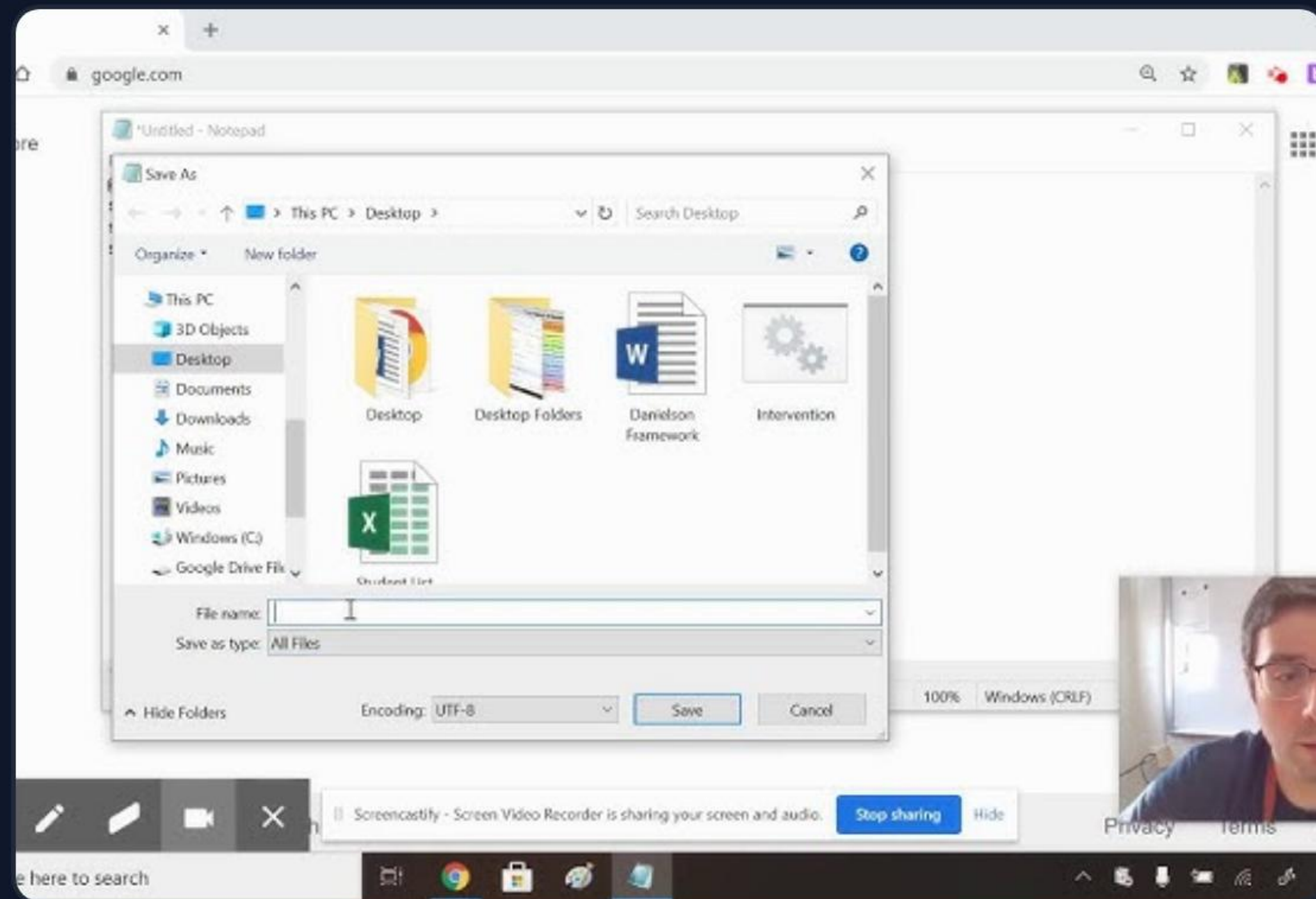


## 39. MULTI-TAB SCENARIOS

Handle popups or new tabs by waiting for the 'page' event on the BrowserContext.

```
const [newPage] = await Promise.all([
  context.waitForEvent('page'),
  page.getByText('Open Tab').click()
]);

await newPage.waitForLoadState();
console.log(await newPage.title());
```





## 40. EXPANDABLE MENUS

- ▶ **Identify Trigger:** Locate the button or link that expands the menu.
- ▶ **Verify Expansion:** Assert the menu is visible before interacting with child links.
- ▶ **Wait for Stability:** Menus often have animations; wait for them to finish.

```
// Standard pattern  
await page.getByLabel('Menu').click();  
await expect(page.locator('.dropdown-content')).toBeVisible();  
await page.getByText('Settings').click();
```





# Advanced Actions **Mastered!**

Practice these 10 actions to handle 99% of web automation challenges. Ready for Post 5?



**6387782073**

TRAINING INQUIRIES



**Abhishesh Mishra**

AUTOMATION MENTOR



# IMAGE SOURCES



<https://cdn.dribbble.com/userupload/15728581/file/original-b8c2cb89f129f40b9447be787e463f5b.jpg?resize=400x0>

Source: [dribbble.com](https://dribbble.com)

---



[https://png.pngtree.com/png-vector/20220923/ourmid/pngtree-illustration-of-a-vector-mouse-hand-cursor-hovering-over-a-button-labeled-free-download-vector-png-image\\_39288052.png](https://png.pngtree.com/png-vector/20220923/ourmid/pngtree-illustration-of-a-vector-mouse-hand-cursor-hovering-over-a-button-labeled-free-download-vector-png-image_39288052.png)

Source: [pngtree.com](https://pngtree.com)

---



<https://raw.githubusercontent.com/alienzhou/web-highlighter/master/docs/img/sample.gif>

Source: [github.com](https://github.com)

---



<https://i.ytimg.com/vi/FIJ25NQBd1Y/hq720.jpg?sqp=-oaymwEhCK4FEIIDSFryq4qpAxMIARUAAAAAGAEIAADIQj0AgKJD&rs=AOOn4CLDYAI1aTAv0BxBtvwltma-VgSGsrA>

Source: [www.youtube.com](https://www.youtube.com)