

# Handling Auto-Suggest Dropdown in Playwright

## What is auto-suggestion dropdown?

An **auto-suggestion dropdown** is a feature in input fields that shows a list of suggested options based on what the user is typing. It helps users by providing relevant suggestions in real-time, making it easier and faster to find what they are looking for.

---

### 1. Find the Input Field and Fill with Partial Value

To interact with the auto-suggest dropdown, we first need to locate the input field and enter the partial value that triggers the dropdown suggestions.

**DOM:**

```
<input type='text' id='src'>
```

**Code:**

```
await page.locator('#src').fill('friends sea');
```

**Explanation:**

The `fill()` method will input the text 'friends sea' into the input field with the ID `src`.

### 2. Wait for the Auto-Suggestion Dropdown to Appear

Auto-suggestions take some time to appear, so we need to wait for the dropdown to be visible. Playwright provides a method called `waitForSelector()` to wait for specific elements to be rendered.

**DOM:**

```
<li class='src_list'>
  <div class='sr'>
    <text class='text'>friends season1</text>
  </div>
</li>
<li class='src_list'>
  <div class='sr'>
    <text class='text'>friends season2</text>
  </div>
</li>
<li class='src_list'>
  <div class='sr'>
    <text class='text'>friends season3</text>
  </div>
</li>
```

**Code:**

```
await page.waitForSelector('//li[contains(@class, "src_list")]/div/text');
```

**Explanation:**

The `waitForSelector()` method ensures that the auto-suggestions have loaded and are visible on the page.

The XPath selector waits for the list item (`li`) elements with class `src_list` to appear, containing the `div` and `text` elements.

### 3. Store All Auto-Suggestion Options and Iterate Through Them

Next, we will store all the auto-suggested options in an array and iterate through them to print the options or perform further actions.

**Code:**

```
const suggestedOptions =  
await page.$$('//li[contains(@class, "src_list")]/div/text');  
  
for (let suggestedOption of suggestedOptions) {  
  const value = await suggestedOption.textContent();  
  console.log(value);  
}
```

**Explanation:**

The `$$(...)` method returns an array of elements that match the given XPath.

We loop through each option, retrieve its text content, and log it to the console.

### 4. Click on a Suggested Option Based on Text Content

If you want to click on a specific option from the suggestions, you can check the text content of each option and click the one that matches your condition.

**Code:**

```
const suggestedOptions = await page.$$('//li[contains(@class,  
"src_list")]/div/text');  
  
for (let suggestedOption of suggestedOptions) {  
  const value = await suggestedOption.textContent();  
  if (value.includes('friends season 2')) {  
    await suggestedOption.click();  
    break;  
  }  
}
```

**Explanation:**

We search through the options and click on the one that contains the string `'friends season 2'`.

The `break` statement ensures we only click the first matching option and exit the loop.

---