Understanding Playwright Selector Errors and Best Practices

🔍Selector Engines in Playwright: Why Syntax Can Fail Even If It Looks Correct

Playwright supports **two different selector engines**:

Selector Type	Used by	Supports `:has-text()`?
CSS selectors	`page.wait_for_selector()`	❌**No**
Playwright selectors	`page.locator()`	✅**Yes**

This distinction is important because even if a selector **looks correct**, using it with the wrong function can lead to errors.

---

1. Error When Using ``

Cause: The function `wait_for_selector()` uses **CSS selectors**, and `:has_text` is **not** valid in CSS.

Incorrect:

```
await page.wait_for_selector("h2:has_text('Results')")
```

Why it fails:

- `has_text` is not a valid CSS pseudo-class.
- Playwright raises a `SyntaxError` because it tries to use this as a native CSS selector.

Correct approaches:

- Use Playwright's **locator API**:

```
await page.locator("h2").filter(has_text="Results").wait_for()
```

- Or use the proper Playwright pseudo selector:

```
await page.locator("h2:has-text('Results')").wait_for()
```

---

2. Error When Using ``

Cause: Incorrect use of attribute selector syntax.

Incorrect:

```python
await page.wait_for_selector("[role:'listitem']")
```

Why it fails:

- CSS attribute selectors must use `=` to match values, not `:`.
- The value must be quoted **inside the brackets** using `'` or `"`.

Correct:

```python
await page.wait_for_selector('[role="listitem"]')
```

---

3. Why `` Sometimes Fails on Sites Like Amazon

Issue: Some websites (like Amazon) keep making background requests (ads, recommendations, tracking, etc.).

Problematic Code:

```python
await page.wait_for_load_state("networkidle")
```

Error:

```
TimeoutError: Page did not become idle
```

Solution: Wait for a specific content selector (e.g., product grid or results title):

```python
await page.wait_for_selector("div.s-main-slot")
```

Or use a safer load state:

```python
await page.wait_for_load_state("load")
```

---

4. Screenshot Saving Mistake

Incorrect:

```
await page.screenshot("form_interation_img/page.png")
```

Why it fails:

- Playwright requires the `path` keyword argument.

Correct:

```
await page.screenshot(path="form_interation_img/page.png")
```

---

5. Best Practice When Moving to a New Page

When a new page loads after an interaction (e.g., search or pagination), don't rely solely on `wait_for_load_state()`.

Why:

- The browser may finish loading but the content you're interested in hasn't rendered yet.

Best Practice:

- Wait for specific selectors instead:

```
await page.wait_for_selector("div.s-main-slot")
```

Or for an element you expect:

```
await page.locator("h2:has-text('Results')").wait_for()
```

By following these patterns, you'll avoid common Playwright errors related to selectors and page timing issues, and your automation scripts will be more reliable and robust.