Common Errors in Playwright When Scraping nytimes.com and Their Causes

---

1. TimeoutError: Page.goto: Timeout XXXXms exceeded.

Cause:

- This error happens when the page doesn't finish loading within the specified time.

Example:

```
await page.goto("https://www.nytimes.com/", wait_until="networkidle",
timeout=90000)
```

Why it fails:

- The New York Times continuously loads resources (ads, analytics, recommendations).
- The `networkidle` state waits for **no network activity for 500ms**, which rarely happens.

Fix:

- Use `wait_until="domcontentloaded"` for faster and more reliable navigation:

```
await page.goto("https://www.nytimes.com/", wait_until="domcontentloaded",
timeout=60000)
```

Or use `try/except` to retry:

```
try:
    await page.goto("https://www.nytimes.com/", wait_until="networkidle",
timeout=60000)
except:
    await page.goto("https://www.nytimes.com/", wait_until="domcontentloaded",
timeout=60000)
```

---

2. TimeoutError: Page.wait_for_selector('a p')

Cause:

- Selector `a p` might be present but not visible or still loading.

Example:

```
await page.wait_for_selector("a p")
```

Why it fails:

- Although many `<p>` tags are nested inside `<a>`, some may not be immediately rendered or visible.
- `wait_for_selector()` waits for **visibility** by default.

Fix: Use `page.locator("a p")` directly or wait a bit before scraping:

```
await page.wait_for_timeout(2000)  # small delay
headlines = await page.locator("a p").all_text_contents()
```

---

3. Incorrect Selector Syntax

Cause:

- Using invalid CSS selectors.

Example:

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

Why it fails:

- CSS attribute selectors must use `=` not `:`

Fix:

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

---

4. Duplicate Headlines Not Removed

Issue:

- When collecting headlines, some are repeated.

Solution: Use `set` to eliminate duplicates:

```
cleaned = list({each.strip() for each in headlines if len(each.strip()) > 25})
```

Or keep order (Python 3.7+):

```
cleaned = list(dict.fromkeys([each.strip() for each in headlines if
len(each.strip()) > 25]))
```

---

5. enumerate() in Printing

Usage:

```
for i, h in enumerate(cleaned, 1):
    print(f"{i}. {h}")
```

Why use `** instead of **`?

- It starts numbering from **1** instead of **0** (more human-friendly).

By understanding these errors and how to resolve them, you'll build more stable and effective Playwright scraping scripts.