Playwright Workshop

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

2 - Basics

Basic Rules in E2E Testing

- Let your test fail

- Use resilient selectors
  - data-*
  - text
  - role
- Multiple Matchers are allowed
- Stay with locator, not do direct fetching

test: useful methods

- test.**describe**
  - Suite: Defines a collection of tests
  - Can apply generic commands on test cases
- test.**only**
  - Runs only this test (relative to the suite)
- test.**skip**
  - Doesn't run this test
- **beforeEach, beforeAll**
  - setup Code
- test.**use**
  - override some configuration values

Playwright's workhorse: `page.locator`

- Returns a promise of a DOM query
  - Is not the DOM element
- Chainable for actions
- Argument for assertions (expect)
- Different selectors (not just CSS)
  - Text selectors
  - Data-testid
  - Location-based
  - XPath
  - React/Vue experimental
- Nestable
- Shadow DOM "piercing"

Text selectors

- ```
  await page.locator("text=Welcome to Eternal").waitFor();  // partial match
  ```

- ```
  await page.locator("text='Welcome to Eternal'").waitFor(); // exact match
  ```

- ```
  await page.locator("'Welcome to Eternal'").waitFor(); // exact match
  ```

- ```
  await page.locator("*", { hasText: "Welcome to Eternal" }).waitFor();  // fails on strict mode
  ```

- ```
  await page.locator("h2", { hasText: "Welcome to Eternal" }).waitFor();  // right approach
  ```

- ```
  await page.locator("h2:has-text('Welcome to Eternal')").waitFor(); // 👍
  ```

- ```
  await page.locator(":has-text('Welcome to Eternal')").waitFor(); // 👎
  ```

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Selectors via data-testid

```
await page.locator("data-testid=greeting").waitFor();


await page.locator("[data-testid=greeting]").waitFor();


await page.locator("css=[data-testid=greeting]").waitFor();


await page.getByTestId("greeting").waitFor();
```

Engines, combinations & nested selection

```
page.locator("data-testid=holiday-card");
```

```
page.locator("data-testid=holiday-card >> text=Vienna");
```

```
page.locator("*data-testid=holiday-card >> text=Vienna");
```

```
page.locator("data-testid=holiday-card", {hasText: "Vienna"});
```


Vienna / Wien

Dive into the capital of the Habsburg empire

ON SALE

With a population of almost 2 million, Vienna is the second largest German-speaking city and breathes history in every

♡   Get a Brochure

London

Explore one of the world's most famous cities

Being a tourist magnet, this city needs no further introduction.

♡   Get a Brochure

Detroit

Top 10 of the cosmopolitan cities of the world

This US city is mainly known for the autombile industry. It hosts the GM's headquarters.

♡   Get a Brochure

```
page.locator("data-testid=holiday-card", { has: page.locator("text=Vienna") })
```

Going down - v1

```
page
    .locator("data-testid=holiday-card", { has: page.locator("text=Vienna") })
    .locator("data-testid=btn-brochure")
```


Vienna / Wien
Dive into the capital of the Habsburg empire

ON SALE

With a population of almost 2 million, Vienna is the second largest German-speaking city and breathes history in every

Get a Brochure

London
Explore one of the world's most famous cities

Being a tourist magnet, this city needs no further introduction.

Get a Brochure

Detroit
Top 10 of the cosmopolitan cities of the world

This US city is mainly known for the autombile industry. It hosts the GM's headquarters.

Get a Brochure

Going down - v2

```
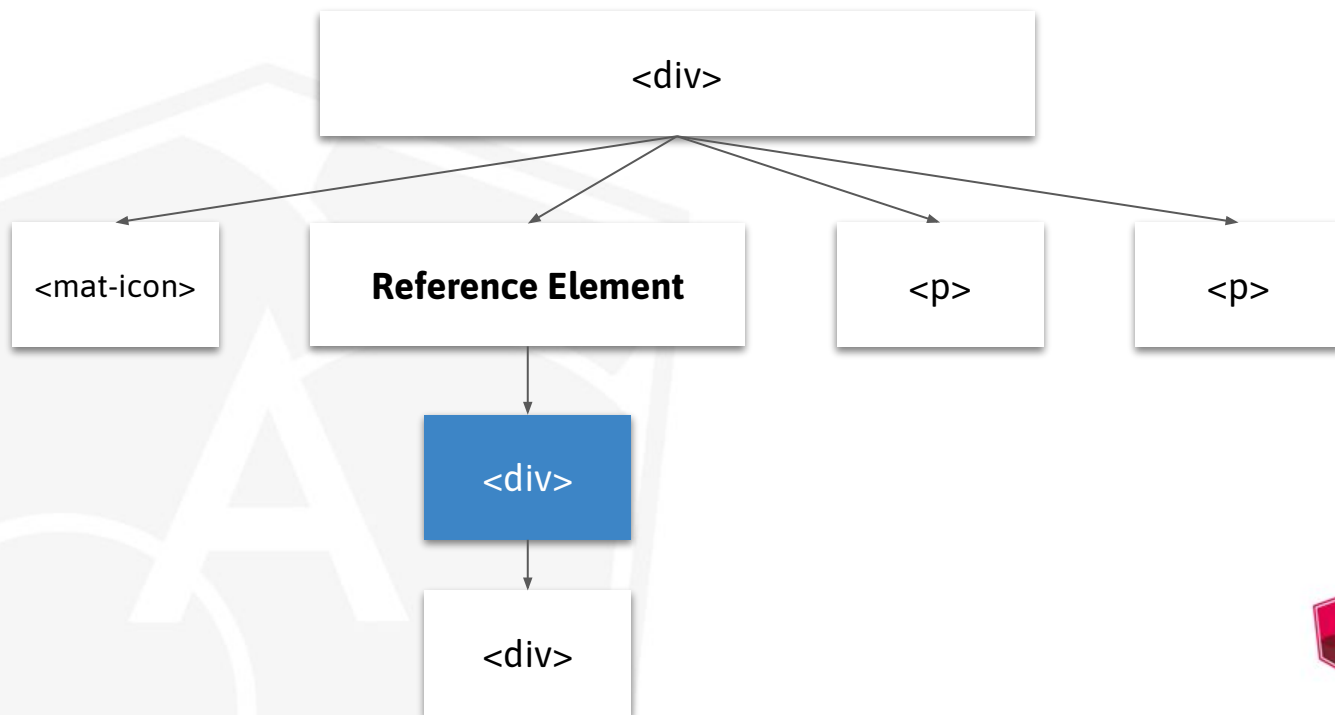page.locator("data-testid=btn-brochure >> nth=0")
```

Going down - v3 🏆

```
page
  .locator(`data-testid=holiday-card`, {
    has: page.locator("mat-card-title", { hasText: "Vienna" })
  })
  .locator("data-testid=btn-brochure")
```

Relative Navigation

Relative Navigation: Direct Child

```
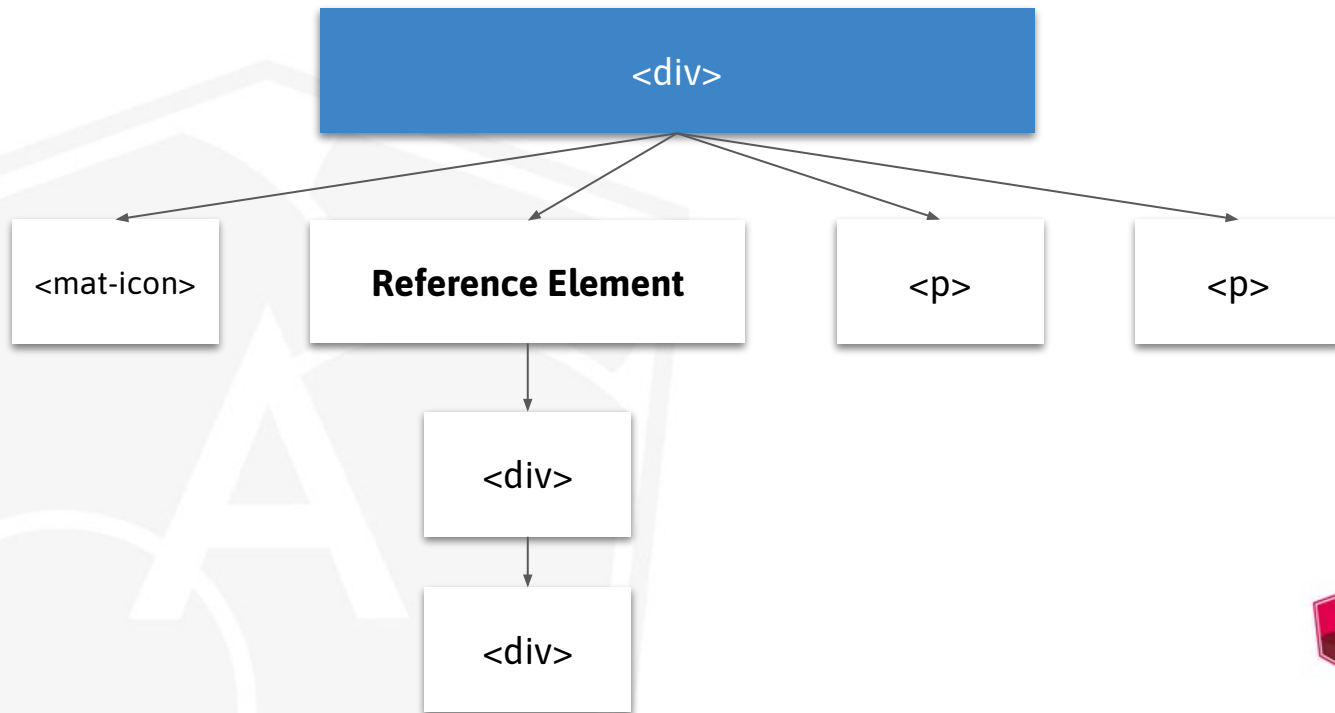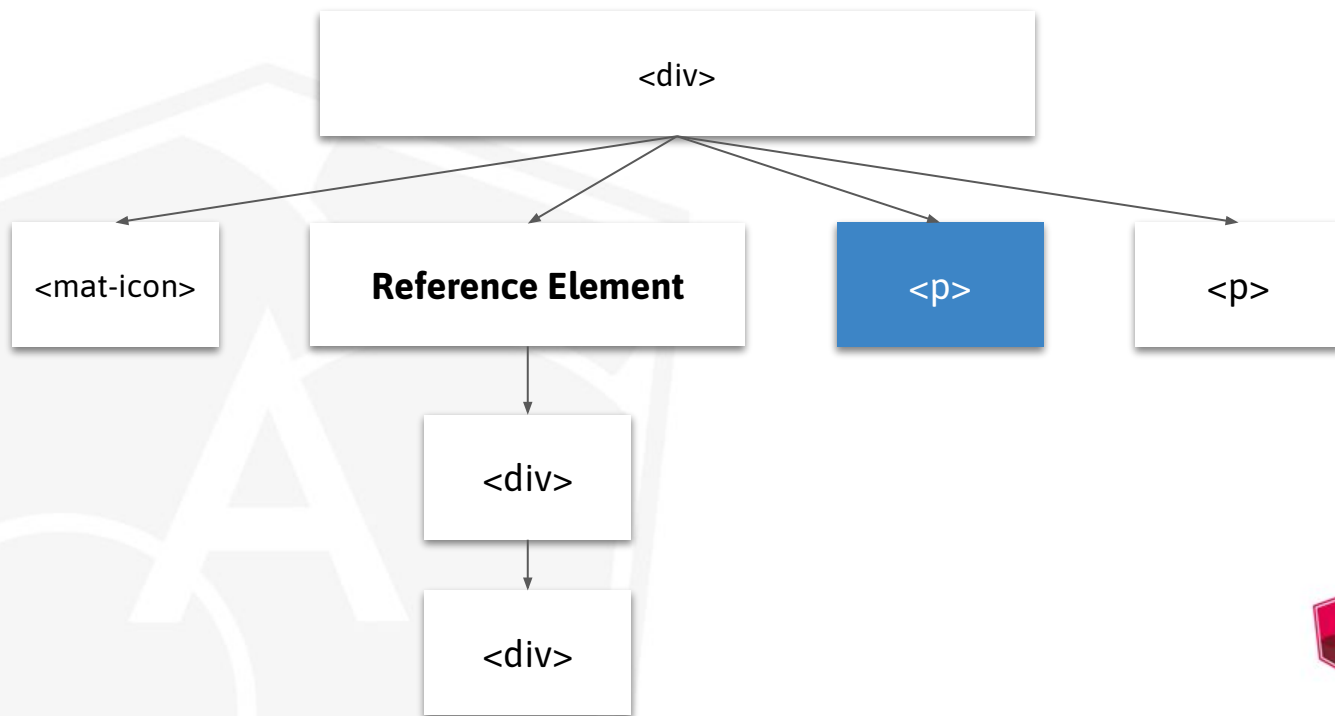data-testid=reference >> >div
```

Relative Navigation: Parent

`data-testid=reference >> ..`

Relative Navigation: Adjacent (Right) Sibling

`[data-testid=reference]` **+ p**

Relative Navigation: Sibling

`[data-testid=reference] ~ p`

Relative Navigation: Preceding Sibling

`data-testid=reference >> .. > ` **`mat-icon`**

Selector Strategies

DOM-Based

- id Attribute
  - Must be unique
  - Browser-internal meaning
  - Consensus not to use it
- CSS Classes
  - Should be used for Layout
  - Don't deal with functionality
- Text
  - `page.locator(text='')`
  - Better to go with User-Facing Selectors
- data-testid
  - Best Practices
  - Recommended by Cypress

User-Facing

- Use text and/or roles
- Includes partially a11y
- "Testing Library Style"
  - No dependencies in Playwright
- Careful when text changes often
  - Fallback to data-testid

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

User-Facing Selectors

1. **page.getByRole**

2. **page.getByLabel**

3. page.getByPlaceholder

4. page.getByText

5. page.getByAltText

6. page.getByTitle

7. page.getByTestId

A11y Tooling

Examples

- `page.getByRole('link', { name: 'Holidays', exact: true })`

  `.click();`

- `page.getByRole('button', { name: /Save/ })`

  `.click();`

- `page.getByLabel(/Firenze/i);`

- `page.getByLabel(/Latitia/)`

  `.getByRole('link', { name: 'Edit Customer', exact: true });`

Layout-specific tests

```
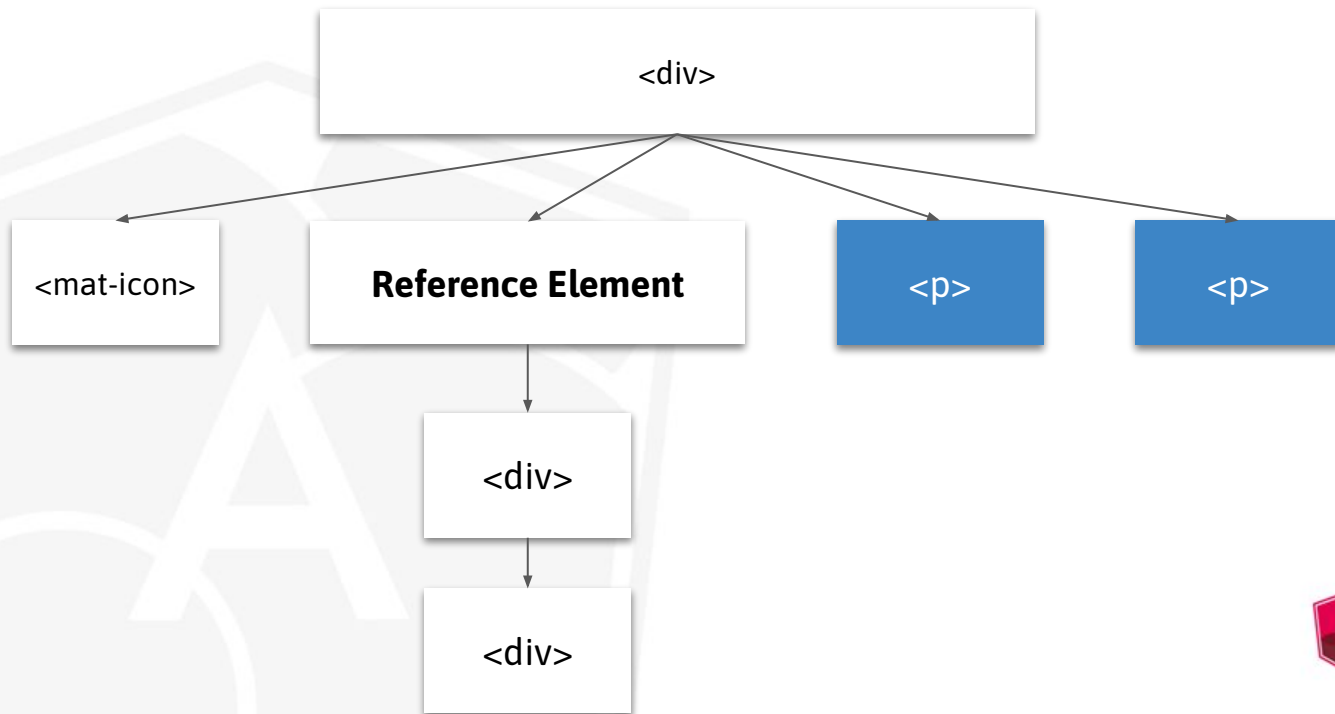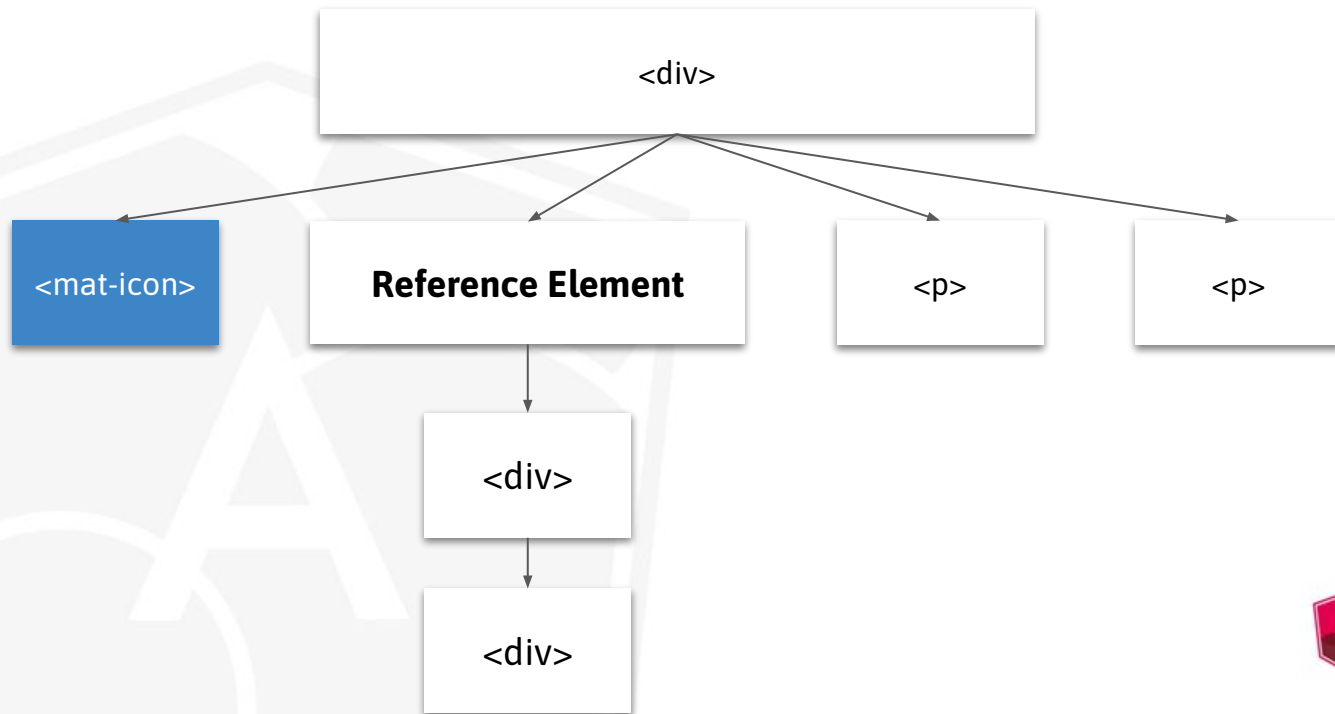test("layout selectors", async ({ page }) => {
  await expect(
    page.locator("p:right-of(:has-text('Logout'))")
  ).not.toBeVisible();
  await expect(
    page.locator("p:left-of(:has-text('Logout'))")
  ).not.toBeVisible();


  await expect(page.locator("p:below(:has-text('Logout'))")).toContainText(
    "Welcome"
  );
});
```

Actions

- click

- fill

- waitFor

  - can be used instead of matcher

- first

  - Required for strict mode

Actionability built-in

- Based on action, Playwright runs checks before the action

- Checks
  - Attached
  - Visible
  - Stable
  - Receive Events (no overlay)
  - Enabled
  - Editable

Assertions

```
await expect([locator]).[matcher]
```

Assertions 1/2

- Assertions are build-in (Part of Playwright)

- Apply matchers to element

  - Always use `locator`

  - Don't forget about the `await`

- Verify that element(s) can be found

  - use `waitFor()` instead expect

Assertions 2/2 - Common Matchers

- `toBeVisible`

- `toHaveCount`

- `toHaveText`

- `toContainText`

- `not.[otherMatcher]`

Lab Time