Python Playwright Selector Cheat Sheet

Table of Contents

Basic Selectors

CSS Selectors

```python
# By tag name
page.locator("button")
page.locator("div")
page.locator("input")

# By ID
page.locator("#submit-btn")
page.locator("#user-profile")

# By class
page.locator(".btn-primary")
page.locator(".navbar-item")

# By attribute
page.locator("[data-testid='login-form']")
page.locator("[type='email']")
```

HTML Example:

```html
<button id="submit-btn" class="btn-primary" data-testid="login-form">
  Submit
</button>
```

XPath Selectors

```python
# Basic XPath
page.locator("xpath=//button")
page.locator("xpath=//div[@class='container']")
page.locator("xpath=//input[@id='username']")

# XPath with text
page.locator("xpath=//button[text()='Submit']")
page.locator("xpath=//span[contains(text(), 'Welcome')]")
```

Text-Based Selectors

By Exact Text

```python
# Button with exact text
page.locator("button", has_text="Submit")
page.get_by_text("Submit")

# Link with exact text
page.get_by_text("Learn More")
page.locator("a", has_text="Learn More")
```

By Partial Text

```python
# Contains text
page.locator("text=Submit")  # Exact match
page.locator("text*=Sub")    # Starts with
page.locator("text$=mit")    # Ends with
page.locator("text/.*Sub.*/")  # Regex

# Using has_text with regex
page.locator("button", has_text=re.compile(r"Submit|Save"))
```

HTML Example:

```html
<button>Submit Form</button>
<button>Save Changes</button>
<a href="/learn">Learn More About Us</a>
```

JavaScript Equivalent:

```javascript
// Text-based selectors in JavaScript
await page.locator('button:has-text("Submit")').click();
await page.getByText('Submit').click();
await page.locator('text=Submit').click();
```

Attribute Selectors

Standard Attributes

python

```python
# By data attributes
page.locator("[data-testid='user-menu']")
page.locator("[data-cy='submit-button']")

# By custom attributes
page.locator("[aria-label='Close dialog']")
page.locator("[role='button']")

# Multiple attributes
page.locator("[type='text'][name='username']")
page.locator("[class*='btn'][disabled]")
```

Attribute Value Matching

python

```python
# Exact match
page.locator("[title='Save document']")

# Contains
page.locator("[class*='btn-']")  # Contains 'btn-'
page.locator("[id*='user']")     # Contains 'user'

# Starts with
page.locator("[class^='btn-']")  # Starts with 'btn-'

# Ends with
page.locator("[class$='-primary']")  # Ends with '-primary'
```

HTML Example:

html

```html
<button
  class="btn btn-primary"
  data-testid="save-btn"
  aria-label="Save document"
  title="Save your work">
  Save
</button>
```

Hierarchy and Relationship Selectors

Parent-Child Relationships

python

```python
# Direct child
page.locator("form > input")
page.locator(".container > .card")

# Descendant
page.locator("form input")
page.locator(".navbar .nav-item")

# Adjacent sibling
page.locator("label + input")

# General sibling
page.locator("h2 ~ p")
```

Playwright-Specific Hierarchy

python

```python
# Locator chaining
page.locator(".container").locator("button")
page.locator("form").locator("[type='submit']")

# Filter by child elements
page.locator("div").filter(has=page.locator("button"))
page.locator("li").filter(has_text="Active")
```

HTML Example:

html

```html
<div class="container">
  <form id="login-form">
    <label for="username">Username</label>
    <input id="username" type="text" name="username">
    <button type="submit">Login</button>
  </form>
</div>
```

State-Based Selectors

Visibility States

python

```python
# Visible elements
page.locator("button:visible")
page.locator(".modal").filter(visible=True)

# Hidden elements
page.locator("input:hidden")
page.locator(".tooltip").filter(visible=False)

# Enabled/Disabled
page.locator("input:enabled")
page.locator("button:disabled")
```

Form States

python

```python
# Checked checkboxes/radios
page.locator("input:checked")
page.locator("[type='checkbox']:checked")

# Selected options
page.locator("option:selected")

# Empty/filled inputs
page.locator("input:empty")
page.locator("textarea:not(:empty)")
```

HTML Example:

html

```html
<form>
  <input type="checkbox" id="terms" checked>
  <input type="text" id="email" disabled>
  <button type="submit" style="display: none;">Submit</button>
</form>
```

Advanced Selectors

Nth-Child Selectors

python

```python
# First/last child
page.locator("li:first-child")
page.locator("tr:last-child")

# Nth child
page.locator("li:nth-child(3)")
page.locator("tr:nth-child(odd)")
page.locator("td:nth-child(even)")

# Nth of type
page.locator("input:nth-of-type(2)")
page.locator("button:last-of-type")
```

Negation Selectors

python

```python
# Not selector
page.locator("button:not(.disabled)")
page.locator("input:not([type='hidden'])")
page.locator("div:not(.hidden)")

# Multiple negations
page.locator("li:not(:first-child):not(:last-child)")
```

HTML Example:

html

```html
<ul>
  <li class="active">First Item</li>
  <li>Second Item</li>
  <li class="disabled">Third Item</li>
  <li>Fourth Item</li>
</ul>
```

Form Element Selectors

Input Types

python

```python
# By input type
page.locator("input[type='email']")
page.locator("input[type='password']")
page.locator("input[type='checkbox']")
page.locator("input[type='radio']")

# Playwright getBy methods
page.get_by_label("Email Address")
page.get_by_placeholder("Enter your email")
page.get_by_role("textbox", name="username")
```

Form Controls

python

```python
# Select elements
page.locator("select[name='country']")
page.get_by_label("Country")

# Textarea
page.locator("textarea[name='message']")
page.get_by_label("Message")

# Buttons
page.locator("button[type='submit']")
page.get_by_role("button", name="Submit")
```

HTML Example:

html

```html
<form>
  <label for="email">Email Address</label>
  <input type="email" id="email" name="email" placeholder="Enter your email">

  <label for="country">Country</label>
  <select id="country" name="country">
    <option value="us">United States</option>
    <option value="ca">Canada</option>
  </select>

  <label for="message">Message</label>
  <textarea id="message" name="message"></textarea>

  <button type="submit">Submit</button>
</form>
```

JavaScript Equivalent:

javascript

```javascript
// Form selectors in JavaScript
await page.locator('input[type="email"]').fill('user@example.com');
await page.getByLabel('Email Address').fill('user@example.com');
await page.getByPlaceholder('Enter your email').fill('user@example.com');
```

Playwright-Specific Selectors

GetBy Methods

python

```python
# By role
page.get_by_role("button")
page.get_by_role("link", name="Home")
page.get_by_role("textbox", name="Search")

# By label
page.get_by_label("Username")
page.get_by_label("Password")

# By placeholder
page.get_by_placeholder("Search...")
page.get_by_placeholder("Enter email")

# By text
page.get_by_text("Click here")
page.get_by_text("Welcome", exact=True)

# By title
page.get_by_title("Close window")

# By test ID
page.get_by_test_id("submit-button")
page.get_by_test_id("user-profile")
```

Frame Selectors

python

```python
# Frame by name
page.frame("frame-name").locator("button")

# Frame by URL
page.frame_locator("iframe[src*='payment']").locator("input")

# Nested frames
page.frame_locator("iframe").frame_locator("iframe").locator("button")
```

Combining Selectors

Multiple Conditions

python

```python
# AND conditions
page.locator("button.btn-primary:not(:disabled)")
page.locator("input[type='text'][required]")

# OR conditions using CSS
page.locator("button, input[type='submit']")
page.locator(".btn-primary, .btn-secondary")

# Complex combinations
page.locator("form").locator("input[type='email']:visible")
```

Filter Combinations

python

```python
# Filter by text and state
page.locator("button").filter(has_text="Submit").filter(visible=True)

# Filter by child elements
page.locator("div").filter(has=page.locator("button.primary"))

# Filter by not having elements
page.locator("li").filter(has_not=page.locator(".disabled"))
```

HTML Example:

html

```html
<div class="form-container">
  <button class="btn btn-primary" disabled>Primary Disabled</button>
  <button class="btn btn-primary">Primary Active</button>
  <button class="btn btn-secondary">Secondary</button>
  <input type="submit" value="Submit Input">
</div>
```

Best Practices

Selector Priority (Recommended Order)

1. **Test IDs** - Most reliable

python

```python
page.get_by_test_id("user-menu")
page.locator("[data-testid='submit-btn']")
```

2. **Semantic Roles** - Accessible and meaningful

python

```python
page.get_by_role("button", name="Save")
page.get_by_role("textbox", name="Email")
```

3. **Labels** - User-facing and stable

python

```python
page.get_by_label("Username")
page.get_by_label("Password")
```

4. **Placeholder Text** - When labels aren't available

python

```python
page.get_by_placeholder("Enter your email")
```

5. **Text Content** - Visible to users

python

```python
page.get_by_text("Sign In")
```

Avoid These Selectors

python

```python
# Fragile - dependent on structure
page.locator("div > div > button:nth-child(3)")

# Brittle - CSS classes may change
page.locator(".css-1234567")

# Unreliable - dependent on order
page.locator("button:nth-of-type(2)")
```

Common Patterns

Dynamic Content

```python
# Wait for element to appear
page.wait_for_selector(".loading-spinner", state="hidden")
page.wait_for_selector("[data-testid='content']", state="visible")

# Handle dynamic IDs
page.locator("[id*='user-']")  # ID contains 'user-'
page.locator("[class^='dynamic-']")  # Class starts with 'dynamic-'
```

Tables and Lists

```python
# Table cells
page.locator("table tr:nth-child(2) td:nth-child(3)")
page.locator("table").locator("tr").filter(has_text="John").locator("td").nth(2)

# List items
page.locator("ul li").filter(has_text="Active")
page.locator("ul").locator("li").nth(0)  # First item
```

Modals and Popups

```python
# Modal content
page.locator("[role='dialog']").locator("button", has_text="Close")
page.locator(".modal").locator("[data-testid='confirm-btn']")

# Popup triggers
page.locator("[aria-haspopup='true']")
page.locator("[data-toggle='dropdown']")
```

Navigation Elements

python

```python
# Menu items
page.locator("nav").get_by_role("link", name="Home")
page.locator(".navbar").locator("a", has_text="Products")

# Breadcrumbs
page.locator(".breadcrumb").locator("a").last
page.locator("[aria-label='breadcrumb']").locator("li").nth(-1)
```

HTML Example:

html

```html
<nav class="navbar">
  <a href="/" role="link">Home</a>
  <a href="/products" role="link">Products</a>
  <div class="dropdown">
    <button aria-haspopup="true" data-toggle="dropdown">
      User Menu
    </button>
  </div>
</nav>

<div role="dialog" class="modal">
  <div class="modal-content">
    <button data-testid="confirm-btn">Confirm</button>
    <button aria-label="Close">×</button>
  </div>
</div>
```

Performance Tips

Efficient Selectors

python

```python
# Good - specific and fast
page.get_by_test_id("submit-btn")
page.locator("#user-form input[name='email']")

# Better than - broad searches
page.locator("input").filter(has_text="email")
page.locator("*").filter(has_text="Submit")
```

Locator Reuse

python

```python
# Store commonly used locators
login_form = page.locator("#login-form")
username_input = login_form.locator("input[name='username']")
password_input = login_form.locator("input[name='password']")
submit_button = login_form.locator("button[type='submit']")

# Use them multiple times
username_input.fill("user@example.com")
password_input.fill("password123")
submit_button.click()
```

Debugging Selectors

Highlight Elements

python

```python
# Highlight matched elements (for debugging)
page.locator("button").highlight()

# Take screenshot with highlights
page.screenshot(path="debug.png")
```

Count Elements

python

```python
# Count matching elements
button_count = page.locator("button").count()
print(f"Found {button_count} buttons")

# Check if element exists
if page.locator("[data-testid='submit']").count() > 0:
    print("Submit button found")
```

Selector Evaluation

python

```python
# Get all matching elements
buttons = page.locator("button").all()
for button in buttons:
    print(button.text_content())

# Get element attributes
element = page.locator("#user-menu")
print(element.get_attribute("class"))
print(element.get_attribute("data-testid"))
```

This comprehensive cheat sheet covers all the essential Playwright selectors for Python, with practical examples and best practices. Keep it handy for quick reference during your automation projects!