

XPath & CSS Selector

1. XPath Cheat Sheet

1.1 XPath Types

Absolute XPath

Starts with /

Example:

```
/html/body/div[1]/form/input
```

- Not recommended (breaks easily with UI changes)

Relative XPath

Starts with //

Example:

```
//input[@id='username']
```

- Recommended and stable

1.2 Basic XPath Syntax

Purpose	XPath Example
By tag	//input
By attribute	//input[@type='text']
By multiple attributes	//input[@type='text' and @name='user']
By text	//button[text()='Login']
Partial text	//button[contains(text(), 'Log')]

1.3 XPath Functions (Very Important)

Function	Usage	Example
<code>contains()</code>	Partial match	<code>//input[contains(@id, 'user')]</code>
<code>starts-with()</code>	Prefix match	<code>//input[starts-with(@id, 'user')]</code>
<code>normalize-space()</code>	Ignore extra spaces	<code>//label[normalize-space()='Username']</code>
<code>text()</code>	Exact text	<code>//a[text()='Home']</code>

1.4 XPath Axes (Advanced & Powerful)

Axis	Meaning	Example
<code>parent::</code>	Parent node	<code>//input/parent::div</code>
<code>child::</code>	Direct child	<code>//div/child::input</code>
<code>following-sibling::</code>	Next sibling	<code>//label/following-sibling::input</code>
<code>preceding-sibling::</code>	Previous sibling	<code>//input/preceding-sibling::label</code>
<code>ancestor::</code>	Any parent above	<code>//input/ancestor::form</code>
<code>descendant::</code>	Any child below	<code>//form/descendant::input</code>

1.5 Indexing & Position

Purpose	Example
First element	<code>(//input)[1]</code>
Last element	<code>(//input)[last()]</code>
Specific position	<code>(//input)[3]</code>

1.6 XPath Best Practices

- Prefer **relative XPath**
- Use **attributes or text**, not indexes
- Avoid absolute XPath
- Avoid dynamic IDs unless using `contains()`

2. CSS Selector Cheat Sheet

2.1 Basic CSS Syntax

Purpose	CSS Selector
By ID	<code>#loginBtn</code>
By class	<code>.submit</code>
By tag	<code>input</code>
Tag + class	<code>input.submit</code>
Tag + ID	<code>input#username</code>

2.2 Attribute Selectors

Selector	Meaning	Example
<code>[attr=value]</code>	Exact match	<code>input[type="text"]</code>
<code>[attr^=value]</code>	Starts with	<code>input[id^="user"]</code>
<code>[attr\$=value]</code>	Ends with	<code>input[id\$="name"]</code>
<code>[attr*=value]</code>	Contains	<code>input[id*="login"]</code>

2.3 CSS Relationships (Combinators)

Symbol	Meaning	Example
(space)	Descendant	<code>div input</code>
>	Direct child	<code>form > input</code>
+	Adjacent sibling	<code>label + input</code>
~	All siblings	<code>label ~ input</code>

2.4 Pseudo-Classes (Common in Automation)

Pseudo-class	Example
<code>:first-child</code>	<code>li:first-child</code>
<code>:last-child</code>	<code>li:last-child</code>
<code>:nth-child(n)</code>	<code>li:nth-child(2)</code>
<code>:checked</code>	<code>input:checked</code>
<code>:enabled</code>	<code>input:enabled</code>

2.5 CSS Selector Limitations

- ✗ Cannot move upward (no parent selector)
- ✗ Cannot locate by visible text
- ✗ Less powerful than XPath for complex DOMs