Assume you are expert in css selector paths for automation like xpath and a good guide to explain for beginners from beginner to advance like id starts with, ends with, contains text etc. where it also includes the html codes for each path

Absolutely! Here's a **complete guide to CSS Selector Paths** for automation testing (like in Cypress, Selenium, or Playwright), ranging from beginner to advanced. Each selector example comes with an associated HTML snippet to visualize how it works.

✅ What is a CSS Selector?

CSS selectors allow you to target HTML elements based on attributes like `id`, `class`, tag name, and more. They're widely used in automation tools to find elements on a page.

🧩 Table of Contents:

---

1. 🟢 Basic Selectors

A. By Tag Name

input

<input type="text" />

B. By ID

#username

<input id="username" />

C. By Class

.form-input

<input class="form-input" />

D. By Multiple Classes

.btn.primary

<button class="btn primary">Click</button>

---

2. 🟡 Attribute Selectors

A. Exact Match

input[type="email"]

<input type="email" />

B. Exists Only (Any value)

input[placeholder]

<input placeholder="Enter your name" />

---

3. 🟠 Partial Attribute Match

A. Starts With ^=

input[id^="user"]

<input id="user-name" />

B. Ends With $=

input[id$="name"]

<input id="first-name" />

C. Contains *=

input[id*="ser"]

<input id="user-id" />

---

4. 🔵 Combinators

A. Descendant (any level inside)

form input

<form>
  <input />
</form>

B. Direct Child >

form > input

<form>
  <input />
</form>

C. Adjacent Sibling `+`

label + input

```
<label>Username</label>
<input />
```

D. General Sibling `~`

h2 ~ p

```
<h2>Title</h2>
<p>Paragraph 1</p>
<p>Paragraph 2</p>
```

---

5. 🟣 Pseudo-classes

A. `:first-child`

li:first-child

```
<ul>
  <li>One</li>
  <li>Two</li>
</ul>
```

B. `:last-child`

li:last-child

C. `:nth-child(n)`

li:nth-child(2)

D. `:nth-of-type(n)`

p:nth-of-type(2)

```
<div>
  <p>First</p>
```

```
  <p>Second</p>
</div>
```

E. `:not(selector)`

```
button:not(.disabled)
```

```
<button class="disabled">Can't Click</button>
<button>Click Me</button>
```

---

6. 🟤 Advanced Real-world Examples

A. Select button with text (CSS cannot do `contains(text)` like XPath, but we can get creative)

```
button[class*="submit"]
```

```
<button class="btn-submit">Submit</button>
```

> **Note**: For exact text, use XPath. CSS cannot match inner text directly.

B. Form with `data-*` attributes

```
input[data-test-id="login-input"]
```

```
<input data-test-id="login-input" />
```

C. Nested elements

```
form#loginForm input[type="password"]
```

```
<form id="loginForm">
  <input type="password" />
</form>
```

---

7. ⚙️ Tips for Test Automation

Use Case	CSS Selector Example	Notes
Button by class	`.btn.submit`	Combine multiple classes
Input by placeholder	`input[placeholder="Email"]`	Reliable for dynamic UIs
Div by custom attribute	`div[data-role="tab"]`	Use `data-*` attributes whenever possible
Avoid IDs if dynamic	Prefer classes or data attributes	Dynamic IDs can break tests
Avoid deep nested paths	Use unique classes or attributes	Keep selectors stable and short

---

💡 Summary Table

Pattern	Selector	Meaning
`#id`	`#login`	Element with `id="login"`
`.class`	`.active`	Element with class "active"
`[attr]`	`[disabled]`	Element with `disabled` attr
`[attr="val"]`	`[type="email"]`	Exact match of attribute
`[attr^="val"]`	`[id^="user"]`	Attribute starts with value
`[attr$="val"]`	`[id$="name"]`	Attribute ends with value
`[attr*="val"]`	`[id*="log"]`	Attribute contains value
`A B`	`div p`	`<p>` inside `<div>`
`A > B`	`ul > li`	Direct child

`A + B`	`label + input`	Next sibling
`:nth-child(n)`	`li:nth-child(2)`	nth element in list

Would you like a **PDF version** or **project folder with HTML + example selectors** to practice with Cypress or Playwright?

Let me know if you want XPath comparison as well.