Assume you are good guide/explainer to the beginner and expert in finding elements in web pages using css selector then provide all possible ways to use css selector from beginner to advanced

Absolutely, Routh Kiran! If you're aiming to **master CSS selectors for finding elements on web pages**—especially useful in automation testing (e.g., **Selenium**, **Cypress**, **Playwright**)—here's a **complete guide from beginner to expert level** with syntax, examples, and best practices.

✅ Beginner Level – Basic CSS Selectors

1. Tag Selector

Selects all elements of a given tag.

input

✔ Matches all `<input>` elements.

---

2. ID Selector

Selects an element with a specific `id`.

#username

✔ Matches: `<input id="username" />`

---

3. Class Selector

Selects all elements with a given `class`.

.login-button

✔ Matches: `<button class="login-button">Login</button>`

---

4. Universal Selector

Selects all elements.

*

✔ Matches everything on the page.

---

5. Group Selector

Combines multiple selectors using commas.

input, select, textarea

✔ Selects all three input-related fields.

✅ Intermediate Level – Attribute Selectors

6. Element with Attribute

input[type]

✔ Matches any input element with a `type` attribute.

---

7. Attribute Equals

input[type="text"]

✔ Matches: `<input type="text" />`

---

8. Attribute Starts With

input[placeholder^="Enter"]

✔ Matches: `<input placeholder="Enter your name" />`

---

9. Attribute Ends With

input[placeholder$="name"]`

✔ Matches: `<input placeholder="Your name" />`

---

10. Attribute Contains

input[placeholder*="user"]

✔ Matches: `<input placeholder="Enter username" />`

✅ Advanced Level – Combinators and Relationships

11. Descendant Selector

form input

✔ Matches all `<input>` inside a `<form>` (any level deep).

---

12. Child Selector

form > input

✔ Matches **only direct** child `<input>` under `<form>`.

---

13. Adjacent Sibling Selector

label + input

✔ Matches the first `<input>` **immediately** after `<label>`.

---

14. General Sibling Selector

label ~ input

✔ Matches all `<input>` elements that follow a `<label>` (not necessarily immediate).

✅ Expert Level – Pseudo-classes and Functions

15. First Child

ul li:first-child

✔ Matches the first `<li>` in every `<ul>`.

---

16. Last Child

ul li:last-child

---

17. Nth Child

table tr:nth-child(2)

✔ Matches the second row of each table.

---

18. Nth of Type

div p:nth-of-type(2)

✔ Second `<p>` under each `<div>` (ignores other tags).

---

19. Not Selector

input:not([type="submit"])

✔ Matches all `<input>` elements **except** type="submit".

---

20. Empty Selector

div:empty

✔ Matches `<div></div>` with no child nodes or text.

✅ Expert++ Level – Multiple Attributes, Complex Chains

21. Multiple Attribute Match

input[type="text"][name="username"]

---

22. Nested & Hierarchical Selectors

div.container form#loginForm input.login-input[type="text"]

✔ Matches very specific element inside a structure.

---

23. Class Selector Chaining

.btn.primary.large

✔ Matches element with **all** three classes.

---

24. Combining with `:not()`

button:not(.disabled):not([type="reset"])

✔ All buttons except those disabled or reset type.

---

25. Complex Example (Real-world usage)

div#main .card:nth-child(2) > h2.title[role="heading"]

✔ A very specific heading inside the second card of a main div.

✅ CSS Selector Best Practices in Automation

Tip	Why
Prefer ID over Class	IDs are unique and faster
Avoid overly complex selectors	They break easily with UI changes
Use `data-*` attributes if available	Made for testing, e.g., `data-test="submit"`
Avoid using indexes like `nth-child()` too much	They are brittle on dynamic pages
Use chaining for precision but avoid too much nesting	Balance between precision and maintainability

---

✅ Tools to Test CSS Selectors

Chrome DevTools Console:

```
document.querySelector('input[name="email"]')
```

1.

Cypress:

```
cy.get('input[type="text"]')
```

2.

Selenium (Java):

```
driver.findElement(By.cssSelector("input[type='text']"));
```

3.

Playwright:

```
page.locator('button.login')
```

4.

✅ Bonus: CSS Selector vs XPath

Feature	CSS Selector	XPath
Syntax	Simpler	More powerful
Readability	High	Low (for beginners)
Performance	Faster	Slower (usually)
Backward Navigation	❌ No	✅ Yes (`..` parent)
Starts With / Contains	✅ Yes	✅ Yes
Use in Cypress	✅ Preferred	❌ Not supported

If you'd like, I can provide **practice questions**, **real-world examples**, or even a **mini-project** where you practice finding elements using all these selectors in a dummy page.

Would you like that?