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 along with how the html looks like

Absolutely! Let's go step-by-step—from **beginner to expert level**—on how to **use CSS selectors to find elements in web pages**, with clear **examples of HTML** and **CSS selectors**.



BEGINNER LEVEL

1. Tag Selector

Selects all elements of a specific HTML tag.

HTML:

Hello World
Another paragraph

CSS Selector:

р

2. ID Selector

Targets an element with a specific id.

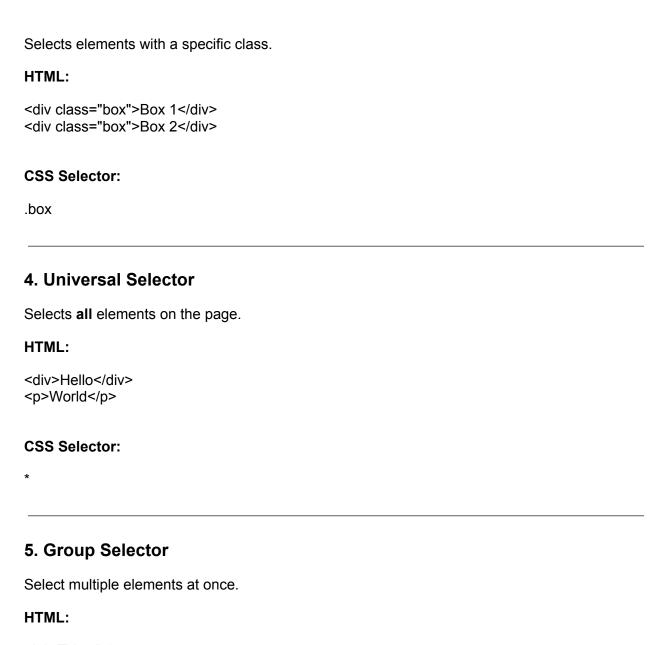
HTML:

<div id="header">Welcome</div>

CSS Selector:

#header

3. Class Selector



<h1>Title</h1>
p>Paragraph

CSS Selector:

h1, p

Ø INTERMEDIATE LEVEL

6. Descendant Selector (space)

Selects elements that are **nested inside** another.

```
HTML:
```

```
<div class="container">
  This is inside container
</div>
```

CSS Selector:

.container p

7. Child Selector (>)

Selects only direct children.

HTML:

```
<div class="container">
  Direct Child
  <div>
  Nested Paragraph
  </div>
</div>
```

CSS Selector:

.container > p

8. Adjacent Sibling Selector (+)

Selects the element that is **immediately next** to a specified one.

HTML:

```
<h2>Heading</h2>
Paragraph right after h2
```

CSS Selector:

h2 + p

9. General Sibling Selector (~)

Selects all siblings after a certain element.

HTML:

```
<h2>Heading</h2>
Paragraph 1
Paragraph 2
```

CSS Selector:

h2 ~ p

10. Attribute Selector

Targets elements by attributes.

a. Exact Match

```
<input type="text">
input[type="text"]
```

b. Contains Word (~=)

```
<div class="btn primary">Click</div>div[class~="primary"]
```

c. Starts With (^=)

```
<a href="https://example.com">Link</a>
a[href^="https"]
```

d. Ends With (\$=)

```
<img src="logo.png">
img[src$=".png"]
```

e. *Contains Substring (=)

```
<div data-id="user-123">User</div>
div[data-id*="user"]
```



ADVANCED LEVEL

11. Nth Child

Selects based on index (starting from 1).

HTML:

```
Item 1
Item 2
li:nth-child(2) /* Selects "Item 2" */
```

12. Nth-of-Type

```
Similar to nth-child, but only counts same tag type.
```

```
<div><span>1</span>23</div>
p:nth-of-type(2) /* Selects 2nd  */
```

13. First-child / Last-child

```
First
 Middle
 Last
li:first-child /* Selects "First" */
li:last-child /* Selects "Last" */
```

14. Not Selector

Exclude elements matching a selector.

```
<div class="box">Box</div>
<div class="circle">Circle</div>
div:not(.circle)
```

15. Empty Selector

Matches elements with no children.

<div class="empty"></div>

div:empty

16. Multiple Class or ID

a. With multiple classes

<div class="box red big">Content</div>

.box.red.big

b. With ID and class

<div id="unique" class="highlight">Text</div>

#unique.highlight

17. Chaining with Attributes, Classes, and Tags

<input type="email" class="input email" name="user-email">

input.email[type="email"][name="user-email"]

Example Use Case in Automation (e.g. Cypress, Selenium)

// Cypress
cy.get('input[type="text"].username')

// Selenium with Java driver.findElement(By.cssSelector("div.container > ul li:first-child"))



Example
div
#header
.box
.container p
.parent > .child
h2 + p
h2 ~ p
[type="text"]
[href^="https"]
[src\$=".png"]
[data-id*="user"]
li:nth-child(2)
li:first-child
<pre>div:not(.active)</pre>
<pre>div.card[data-id^= "12"]</pre>

Would you like this turned into a **PDF reference sheet**, or want a **project or exercise** to practice these?