Pseudo-classes
Pseudo-elements
CSS Positions (static, relative, absolute, fixed, sticky)
z-index

1.What is a Pseudo-Class?
A pseudo-class is used in CSS to define the style of an element when it is in a special state.
For example:
When you hover your mouse over a button
When a user clicks on a link
When an input box is focused
These are not extra elements. They are just states of existing elements.
Example:

```
button:hover {
  background-color: blue;
}
```

This means: when the mouse hovers over the button, its background turns blue.
Other common pseudo-classes:
:hover – when the mouse is over an element
:focus – when an element (like an input) is clicked or selected
:active – when an element is being clicked
:first-child – targets the first child inside a parent

2.What is a Pseudo-Element?
A pseudo-element lets you style a specific part of an element.
For example:
Style only the first letter of a paragraph
Add something before or after the content of an element
These act like they create new virtual elements, but they don’t really exist in the HTML.
Example:

```
p::first-letter {
  font-size: 200%;
}
```

This means: make the first letter of every paragraph twice as big.
Another example:

```
div::before {
  content: "Hello ";
}
```

This adds the word “Hello” before the content inside every div.
Common pseudo-elements:
::before – adds content before an element
::after – adds content after an element
::first-letter – styles the first letter
::first-line – styles the first line of text

3.Difference Between Pseudo-Class and Pseudo-Element
Pseudo-class: Styles based on the state of an element
Pseudo-element: Styles a part of an element or adds virtual content

Example of pseudo-class:

```
a:hover {
  color: red;
}
```

Example of pseudo-element:

```
h1::before {
  content: "* ";
}
```

4.When Do We Use Pseudo-Classes and Pseudo-Elements?
Use pseudo-classes when you want to style something based on how the user interacts with the page.
Use pseudo-elements when you want to style parts of an element or add something extra without editing the HTML.

5.What is CSS Position?
CSS position property is used to control where an element appears on the page.
There are 5 main values:
static
relative
absolute
fixed
sticky

1. static
This is the default position. Every element is placed normally, one after another from top to bottom.
You don’t need to write position: static because it is used automatically.
Elements cannot be moved using top, left, right, or bottom.
2. relative
When you use position: relative, the element stays in its normal spot, but you can move it a little using top, left, right, or bottom.
Example:

```
div {
  position: relative;
  top: 20px;
  left: 10px;
}
```

This moves the element 20px down and 10px to the right from where it normally was.
3. absolute
When you use position: absolute, the element is removed from the normal flow of the page. It is placed relative to the nearest parent that has a position (not static).
If no parent has a position, it uses the entire page (body) as a reference.
Example:

```
div {
  position: absolute;
  top: 50px;
  left: 100px;
}
```

This places the element exactly 50px from the top and 100px from the left of its

parent or page.
4. fixed
With position: fixed, the element is placed relative to the browser window, not the page. It does not move when you scroll.
Good for sticky headers or floating buttons.
Example:

```
div {
  position: fixed;
  top: 0;
  right: 0;
}
```

This places the element at the top-right corner of the screen, and it stays there even if you scroll.
5. sticky
With position: sticky, the element acts like relative at first, but when you scroll, it becomes stuck to a position (like the top) until the end of its container.
It’s used for things like sticky headers inside scrollable areas.
Example:

```
div {
  position: sticky;
  top: 0;
}
```

This element scrolls normally, but when it reaches the top of the page, it sticks there.

6.Differences Between All Position Types
static: normal, default position
relative: moves an element slightly from where it is
absolute: places element exactly, based on parent or page
fixed: always stays in one place on the screen
sticky: sticks to a position while scrolling, but only inside its container

7.What is z-index?
z-index is used to control which elements appear on top when they overlap.
It only works on elements that have a position (relative, absolute, fixed, or sticky).
Higher number = on top
Lower number = behind
Example:

```
div {
  position: absolute;
  z-index: 10;
}
```

This element will appear in front of other elements with a lower z-index.
Use z-index when you want to:
Put a modal or popup on top
Keep a dropdown above other content
Make sure a sticky header stays on top

Summary Quick Revision :

Pseudo-class: adds styles based on element state (:hover, :focus)
Pseudo-element: styles part of an element (::before, ::after)
Position:
static: default, no special positioning
relative: move slightly from its place
absolute: exact placement in parent or page
fixed: stays at one spot on screen
sticky: scrolls and sticks at a point
z-index: controls which element is in front when they overlap

Web Pages That Use Them:

You can use these in any webpage that needs:
Interactive buttons
Special headings
Styled paragraphs
Tooltip effects
Input focus styles