---

1. Define the use of the ruleset

A **CSS ruleset** is the basic building block of CSS that defines **how HTML elements should be styled**. A ruleset consists of two main parts: a **selector** and a **declaration block**.

Structure of a ruleset:

```
selector {
  property: value;
}
```

- **Selector** – Specifies which HTML element(s) the rule applies to.
- **Property** – Defines the style attribute (e.g., color, font-size).
- **Value** – Specifies the value of the property.
- **Declaration block** – Contains one or more property–value pairs enclosed in curly braces.

Example:

```
p {
  color: blue;
  font-size: 16px;
}
```

Use of ruleset:

- Applies consistent styling to elements
- Separates content from presentation
- Improves readability and maintainability
- Allows reuse of styles across multiple pages

Without rulesets, CSS styling would not be structured or manageable.

---

2. Can we include multiple CSS3 files in one HTML document? Explain.

Yes, **multiple CSS3 files can be included in a single HTML document**. This is commonly done in real-world projects to organize styles efficiently.

Example:

```
<link rel="stylesheet" href="layout.css">
<link rel="stylesheet" href="colors.css">
<link rel="stylesheet" href="responsive.css">
```

Explanation:

- Each CSS file can handle a specific purpose (layout, theme, responsiveness).
- Browsers load CSS files **in the order they appear**.
- If multiple files define the same rule, the **last loaded file takes precedence**.

Advantages:

- Better code organization
- Easier maintenance
- Reusable styles
- Faster teamwork in large projects

This approach is widely used in modern web development.

---

3. What is the difference between the usage of an ID and a Class?

ID and **Class** selectors are used to apply styles to HTML elements, but they differ in purpose and usage.

ID:

- Unique (used only once per page)
- Higher specificity
- Used for unique elements

Example:

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

```
#header {
  background-color: blue;
}
```

Class:

- Reusable (used on multiple elements)
- Lower specificity than ID
- Used for grouping similar elements

Example:

```
<p class="text"></p>
<div class="text"></div>
```

```
.text {
  color: green;
}
```

Key Differences:

- ID → unique, #
- Class → reusable, .
- ID overrides Class in conflicts
- Classes are preferred for styling

---

4. Elaborate the function of inline CSS with an example

Inline CSS is applied directly to an HTML element using the style attribute.

Function of inline CSS:

- Styles a single element
- Overrides internal and external CSS
- Useful for quick testing or one-time styling

Example:

```
<h2 style="color: red; text-align: center;">
  Inline CSS Example
</h2>
```

Advantages:

- Immediate effect
- No need for external files

Disadvantages:

- Poor maintainability
- Repetition of code
- Not suitable for large projects

Inline CSS should be avoided in professional web development.

---

5. What are pseudo-elements in CSS? Explain with code.

Pseudo-elements are used to style **specific parts of an element**, rather than the entire element.

They are written using double colon :: syntax.

Common pseudo-elements:

- ::before
- ::after
- ::first-letter
- ::first-line

- ::selection

Example:

```
p::first-letter {
  font-size: 30px;
  color: red;
}
h1::before {
  content: "★ ";
  color: gold;
}
```

Use:

- Decorative content
- Styling first letters or lines
- Adding icons or symbols

Pseudo-elements improve design without extra HTML.

---

6. How would you define the pseudo-classes in CSS3?

Pseudo-classes define the **special state of an element**, such as hover, focus, or visited state.

They use a single colon : syntax.

Common pseudo-classes:

- :hover
- :active
- :focus
- :visited
- :nth-child()

Example:

```
a:hover {
  color: red;
}
li:nth-child(2) {
  background-color: yellow;
}
```

Purpose:

- Enhance interactivity
- Style elements based on user actions
- Improve user experience

---

7. How many ways can CSS be integrated into a web page?

CSS can be integrated into a webpage in **three ways**:

1. **Inline CSS**

Text

2. **Internal CSS**

<style>

p { color: green; }

</style>

3. **External CSS**

<link rel="stylesheet" href="style.css">

Best practice:

External CSS is the most recommended method.

---

8. Define the property used for image scroll controlling

The CSS property used for **image scroll controlling** is:

background-attachment

It controls whether a background image scrolls with the page or remains fixed.

Values:

- scroll (default)
- fixed
- local

Example:

```
body {
  background-image: url("bg.jpg");
  background-attachment: fixed;
}
```

This property is often used for **parallax effects**.

9. What are the differences between relative and absolute in CSS? Explain.

position: relative and position: absolute control how elements are positioned.

Relative Position:

- Positioned relative to its **normal position**
- Does not remove element from document flow

Example:

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

Absolute Position:

- Positioned relative to the **nearest positioned ancestor**
- Removed from normal document flow

Example:

```
div {
  position: absolute;
  top: 20px;
}
```

Key Differences:

Relative	Absolute
Keeps space	Removes space
Relative to itself	Relative to parent
Used for minor adjustments	Used for precise layout