Zed-King

GROUP OF INSTITUTIONS

ISO 9001 : 2015 Certified

M2-R5.1

CHAPTER-4

CSS(Cascading Style Sheet)

WEB DESIGNING & PUBLISHING

" Think • Code • Beautify "

Topics Covered

Introduction to CSS

Types of CSS

CSS Selectors

CSS Properties



Web Designing Instructor

Ms. Doli

Presented By

Zed-King Group of Institutions, Kaithal

"CSS is the art that gives life to HTML — It transforms code into beauty."

ZED-KING INSTITUTE

Index – CSS

Sr. No.	Topics	Page No.
4.1	Introduction to CSS	
4.1.1	CSS Features	
4.1.2	Advantages of CSS	
4.1.3	Major Differences between CSS, CSS2, CSS3	
4.1.4	Define CSS Comments	
4.2	Types of CSS	
4.3	CSS Selectors	
4.3.1	CSS Syntax	
4.3.2	Types of Selectors	
4.3.2.1	Universal Selector	
4.3.2.2	ID Selector	
4.3.2.3	Tag Selector	
4.3.2.4	Class Selector	
4.3.2.5	Sub Selector	
4.3.2.6	Attribute Selector	
4.3.2.7	Group Selector	
4.4	CSS Properties	
4.4.1	Background Properties	
4.4.2	Block Properties	
4.4.3	Box Properties	
4.4.4	List Properties	
4.4.5	Border Properties	
4.4.6	Positioning Properties	
4.4.7	Text Properties	
4.4.8	Font Properties	
4.4.9	Table Properties	
4.4.10	Menu Design	
4.4.11	Image Gallery	

ZED-KING INSTITUTE

Chapter 4

CSS (Cascading Style Sheet)

4.1 Introduction to CSS

Definition:-

- **CSS = Cascading Style Sheets**
- Developed by **Håkon Wium Lie in 1996**.
- Supported by all modern browsers.
- **It** is a stylesheet language used to **control the presentation** (look) of HTML documents.
- It defines **how elements are displayed** on screen, paper, or other media.
- CSS brings **style, layout, and responsiveness** to web pages.
- It styles HTML elements → decides **color, size, spacing, layout, background, fonts**.
- "Cascading" means priority order:

Inline CSS > Internal CSS > External CSS

Example Program in CSS:-

ZED-KING INSTITUTE

4.1.1 CSS Features

Major Features of CSS

1. Separation of Structure and Style

- **HTML** → structure (content).
- **CSS** → style (design).
- Example: HTML makes a paragraph, CSS changes its color & size.

2. Less Code & Reusability

- One CSS file can be linked with **hundreds of web pages**.
- Reduces code duplication → saves time.

3. Consistency Across Pages

- Ensures **uniform look** on all pages (fonts, colors, layouts).

4. Cascading (Priority Order)

- CSS applies styles based on priority:
    a. Inline CSS (Highest)
    b. Internal CSS
    c. External CSS
    d. Browser default (Lowest)

5. Responsive Web Design

- Websites adjust automatically on **mobile, tablet, desktop**.
- Done using **media queries, flexbox, grid**.

6. Improved Performance

- External CSS files are **cached by browsers**.
- Websites load faster.

7. Advanced Styling & Visual Effects

- Supports **colors, gradients, shadows, filters, transitions, animations**.
- Makes websites interactive and attractive.

8. Layout Management

ZED-KING INSTITUTE

- **CSS provides powerful tools:**
  - **Box Model** → margin, border, padding, content.
  - **Flexbox** → one-dimensional layouts.
  - **Grid** → two-dimensional layouts.
  - **Positioning & Float** → element placement.

9. Cross-Browser Compatibility

- Works on **all modern browsers**.

10. Media Support

- Can design for **different media types**:
  - Screen (websites)
  - Print (print-friendly CSS)
  - Speech (screen readers for blind users)

11. Easy Maintenance

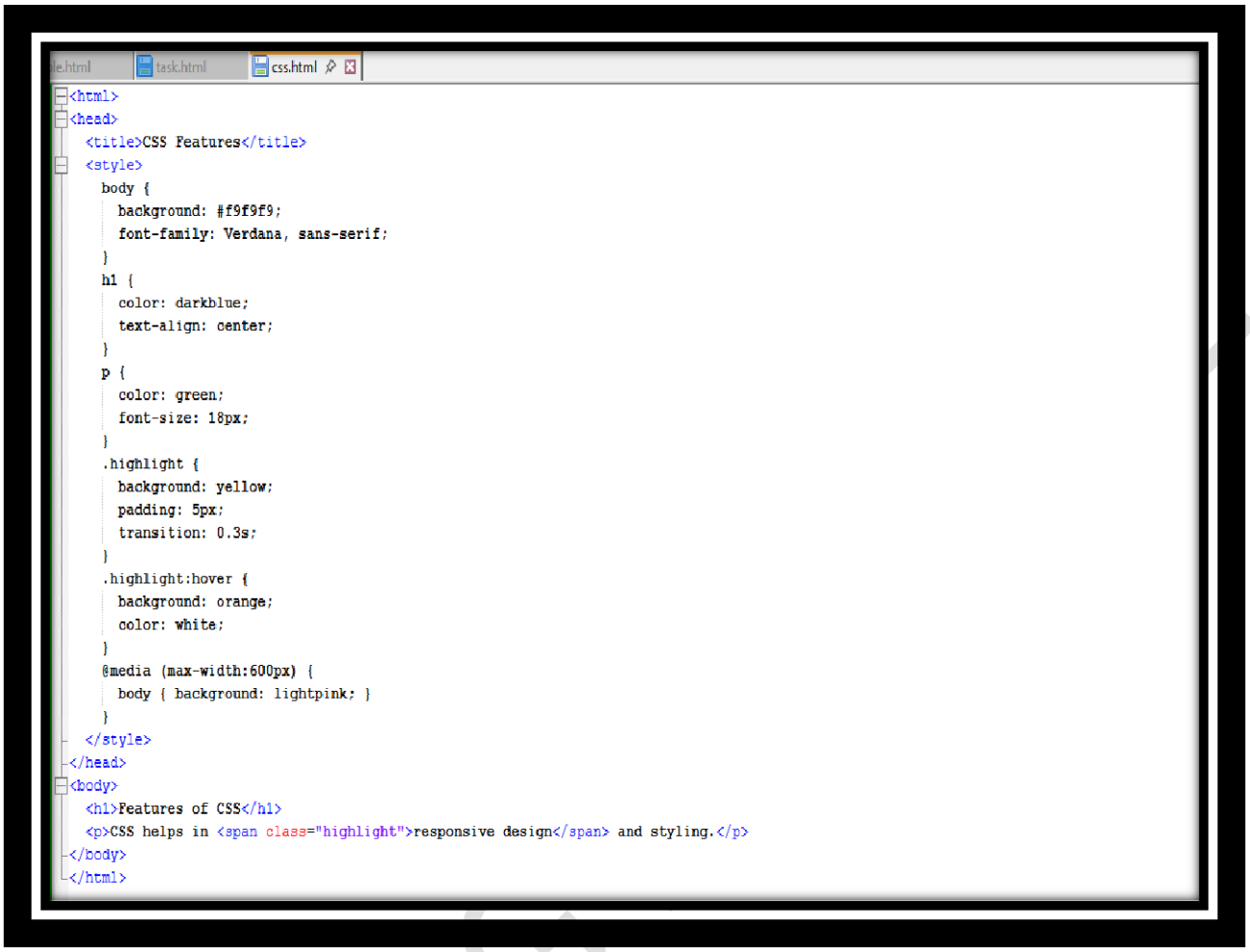
- Change once in CSS → automatically updates all linked pages.

12. Platform Independence

- Works on any OS (Windows, Linux, Mac, Android, iOS).
- Open standard (no license required).

ZED-KING INSTITUTE

Example Code Demonstrating Features



```
<html>
<head>
  <title>CSS Features</title>
  <style>
    body {
      background: #f9f9f9;
      font-family: Verdana, sans-serif;
    }
    h1 {
      color: darkblue;
      text-align: center;
    }
    p {
      color: green;
      font-size: 18px;
    }
    .highlight {
      background: yellow;
      padding: 5px;
      transition: 0.3s;
    }
    .highlight:hover {
      background: orange;
      color: white;
    }
    @media (max-width:600px) {
      body { background: lightpink; }
    }
  </style>
</head>
<body>
  <h1>Features of CSS</h1>
  <p>CSS helps in <span class="highlight">responsive design</span> and styling.</p>
</body>
</html>
```

4.1.2 Advantages of CSS

- **CSS Saves Time** → one file can be used for many pages.
- **Consistent Design** → same look everywhere.
- **Responsive Design** → works on mobile, tablet, desktop.
- **Easy to Update** → one change reflects on all pages.
- **Beautiful Styles** → colors, fonts, animations, effects.
- **Cross Browser** → works on Chrome, Firefox, Edge, etc.
- **Faster Pages** → external CSS is cached by browser.
- **Media Support** → different styles for print, screen, speech.
- **Free & Open** → no license, works on all devices.

ZEDKING

Quick Memory Trick

- **CSS1 =** Basic Styling
- **CSS2** = Layout Control
- **CSS3 =** Modern Effects + Responsive Design

4.1.3 Major Differences Between CSS, CSS2, and CSS3

CSS1 – 1996

- First version of CSS.
- Provided basic styling only (colors, fonts, text alignment, background).
- Very limited features.
- No support for advanced layouts, animations, or media queries.

CSS2 – 1998

- Extension of CSS1 with more control over layouts.
- **Introduced:**
  - Positioning (absolute, relative, fixed)
  - Z-index (stacking order)
  - Media types (screen, print, speech)
  - Bidirectional text support (for languages like Arabic, Hebrew).
- Still one big specification (not modular).
- Better browser support compared to CSS1.

CSS3 – 1999 onwards (modular)

- Biggest update to CSS, still evolving.
- Modular structure (broken into separate modules like selectors, backgrounds, animations, etc.).
- **Introduced:**
  - Advanced Selectors → nth-child(), :last-child, :not()

ZED-KING INSTITUTE

- Responsive Design → media queries, flexible layouts
- Advanced Layouts → Flexbox, Grid, Multi-column
- Backgrounds & Borders → gradients, multiple backgrounds, border-radius, box-shadow
- Animations & Effects → transitions, keyframe animations, transforms (2D/3D)
- Color Models → RGBA, HSLA, opacity
- Typography → web fonts (@font-face), text-shadow, word-wrap

Major Differences between CSS, CSS2, and CSS3 in Tabular Form

Feature / Version	CSS (CSS1)	CSS2	CSS3
Introduced	1996	1998	1999 (modular updates continue till now)
Basic Purpose	Styling basic HTML elements (colors, fonts, text alignment)	Added more control over page layout and printing	Advanced styling, responsive design, animations, effects
Modules	No modules (single specification)	Still a single large specification	Split into **modules** (e.g., Selectors, Backgrounds, Animations, Transforms, etc.)
Selectors	Very simple selectors (element, class, ID)	Added advanced selectors like child, adjacent, attribute selectors	More powerful selectors like nth-child(), :last-child, :not(), etc.
Layout Features	Basic styling only	Added positioning, z-index, media types, bidirectional text	Flexbox, Grid system, multi-column layout, box-sizing

ZED-KING INSTITUTE

Feature / Version	CSS (CSS1)	CSS2	CSS3
Media Support	Not supported	Introduced **media types** (screen, print)	Improved with **media queries** (responsive design)
Backgrounds & Borders	Only solid backgrounds	Limited background options	Multiple backgrounds, gradients, border-radius, shadows
Fonts & Text	Basic font properties	Introduced downloadable fonts via @font-face	Advanced typography, text-shadow, word-wrap
Color Support	Basic named colors	Extended color system	RGBA, HSLA, Opacity, Transparency
Animations & Effects	Not available	Not available	Transitions, Animations, Transform (2D/3D)
Browser Compatibility	Very limited	Better, but still inconsistent	Widely supported by modern browsers

Quick Exam Points

- **CSS1:** Basic styling (colors, fonts, text).
- **CSS2:** Added layouts, positioning, media types.
- **CSS3:** Advanced features (modules, responsive design, animations, gradients, flexbox, grid).
- **CSS (1996)** → basic styling only.
- **CSS2 (1998)** → selectors, media types.
- **CSS3 (1999-now)** → responsive (media queries), animations, transitions, grid, flexbox.

ZED-KING INSTITUTE

4.1.4 CSS Comments

Definition

CSS comments are notes written inside a CSS file to **explain code**.

- They are **ignored by the browser** (not shown on the webpage).
- Used only for **developers** to understand the code better.
- **/* This is a comment */**
- Used for notes inside CSS, ignored by browser.

Syntax of CSS Comment

/* This is a CSS comment */

- Start with **/***
- End with ***/**
- Anything written between them is a **comment**.

Example

/* Main heading style */

```
h1 {
  color: blue;     /* heading text color */
  font-size: 30px;   /* heading size */
}
```

Key Points

- CSS does **not support single-line comments** like // (only /* */).
- Comments are helpful for **team projects, learning, and debugging**.
- Can be written **anywhere** in CSS code.
- CSS comments are **notes or explanations** written inside CSS code.
- They are **not executed by the browser** → only for humans to understand the code.
- Help in making code **readable, clear, and easy to maintain**.

ZED-KING INSTITUTE

Types of Comments in CSS

1. Single-line Comment Example

```
<html>
<head>
  <title>Single-line Comment Example</title>
  <style>
    /* This is a single-line comment */
    h1 {
      color: blue;   /* Heading color is blue */
    }

    p {
      font-size: 18px;  /* Paragraph text size */
    }
  </style>
</head>
<body>

  <h1>Welcome</h1>
  <p>This is a paragraph styled using CSS.</p>

</body>
</html>
```

2. Multi-line Comment Example

```
<html>
<head>
  <title>Multi-line Comment Example</title>
  <style>
    /*
      This is a multi-line comment.
      CSS will style the body, h1 and p.
      Comments are not displayed in browser.
    */
    body {
      background-color: lightyellow;
    }

    h1 {
      color: green;
      text-align: center;
    }

    p {
      color: brown;
    }
  </style>
</head>
<body>

  <h1>Welcome</h1>
  <p>This paragraph has CSS styling.</p>

</body>
</html>
```

Single-line: **/* comment */ (used for short notes).**

Multi-line: **/* ... multiple lines ... */ (used for explanations).**

4.2 Types of CSS

1.Inline CSS

1. **Inline CSS** → inside tag.

- Used inside the HTML tag.
- In which style as a attribute behave krega.
- Quick but not reusable.

<h2 style="color: red; font-size: 30px;">Inline CSS Example</h2>

** Hello **

```
<html>
<head>
<title>Inline CSS Example</title>
</head>
<body>
<p style="color:green;">This is my paragraph.</p>
</body>
</html>
```

2.Internal CSS

- Written inside <style> tag within <head>.
- Useful for **single-page styling**.
- **Internal CSS** → inside <style> tag in <head>.

<style>
h1 { color: red; }
</style>

ZED-KIN INSTITUTE

```html
<html>
    <head>
        <title>Internal CSS Example</title>
        <style>
            /* Heading style */
            h2 {
              color: blue;
              text-align: center;
            }

            /* Paragraph style */
            p {
              color: green;
              font-size: 16px;
              font-family: Arial, sans-serif;
            }
        </style>
    </head>
<body>
  <h2>Welcome to My Website</h2>
  <p>This is a simple paragraph with internal CSS.</p>
  <a href="https://example.com">Visit Example</a>
  <br><br>
  <img src="1.jpg" alt="Flower">
  <br><br>
  <marquee>This is a scrolling text!</marquee>
</body>
</html>
```

2. External CSS

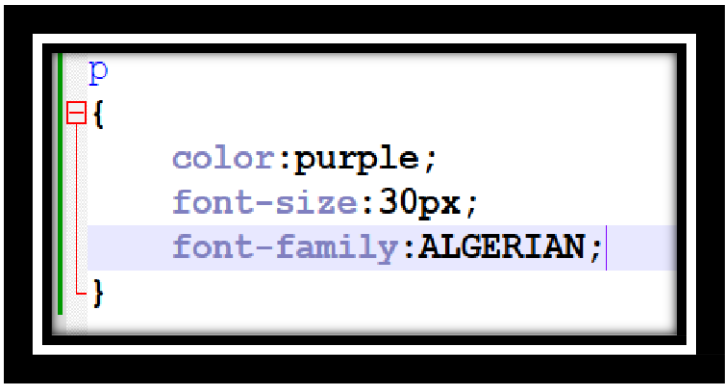
- Written in a separate .css file.
- Best practice for **large websites**.
- **External CSS** → separate .css file linked.
- **Best = External (easy to maintain).**

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

1st Main File(index.html):-

```html
<html>
    <head>
        <title>CSS External Example</title>
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        <p>This article has an hsla transparent purple background color.</p>
    </body>
</html>
```

2nd File(style.css):-



```
p
{
    color:purple;
    font-size:30px;
    font-family:ALGERIAN;
}
```

CSS Priority Order (Cascade Rule)

1. **Inline CSS** → Highest Priority
2. **Internal CSS**
3. **External CSS**
4. **Browser Default** → Lowest Priority

Important Points

- HTML Element → Inline CSS (style="...")
- HTML Head → Internal CSS (<style>...</style>)
- Separate File→ External CSS (style.css linked)
- Inline = Inside tag
- Internal = Inside <style>
- External = External file

Comparison Table (Easy for Exam)

Type	Where written?	Scope	Best For
Inline	Inside HTML tag	One element only	Small fixes
Internal	<style> tag	One webpage	Small websites

ZED-KING INSTITUTE

Type	Where written?	Scope	Best For
External	.css file	Multiple pages	Large projects

4.3 CSS Selectors

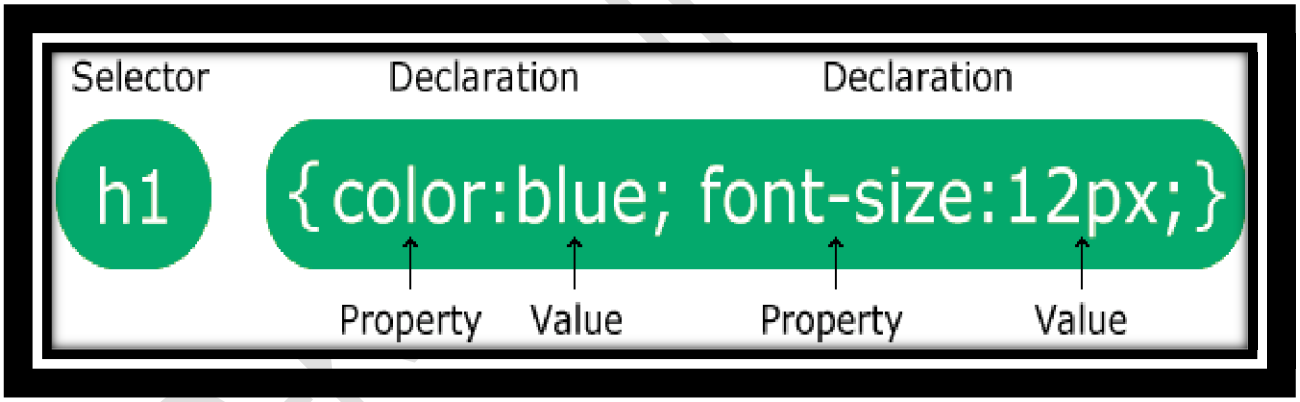
CSS Selectors

- A **CSS Selector** is used to **select (target) HTML elements** so that we can apply styles to them.
- Without selectors, we cannot tell CSS **"which element to style."**

4.3.1 CSS Syntax

A CSS rule consists of a selector and a declaration block.

CSS Syntax



- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

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

ZED-INSTITUTE

4.3.2 Types of Selectors

There are mainly seven type of selectors:-

1. Universal Selector(*)
2. ID Selector(#)
3. Tag Selector or Element Selector or Type Selector(tag name)
4. Class Selector(.)
5. **Sub Selector:-**
   i. Descendent Selector(space)
   ii. Child Selector(>)
   iii. Adjacent Selector(+)
   iv. General Sibling Selector(~)
6. **Attribute Selector**
   i. Fdsdfd
   ii. Dfdfds
   iii. Dfdgfd
   iv. Dfdgdfg
   v. Dfgdfgdf
   vi. Fdgdfg
   vii. gfgnfjkng
7. Group Selector

Types of CSS Selectors in details:-

1. Universal Selector (*)

```
* {
margin: 0;
padding: 0;
color: blue;
}
```

The * applies style to all elements in the page.

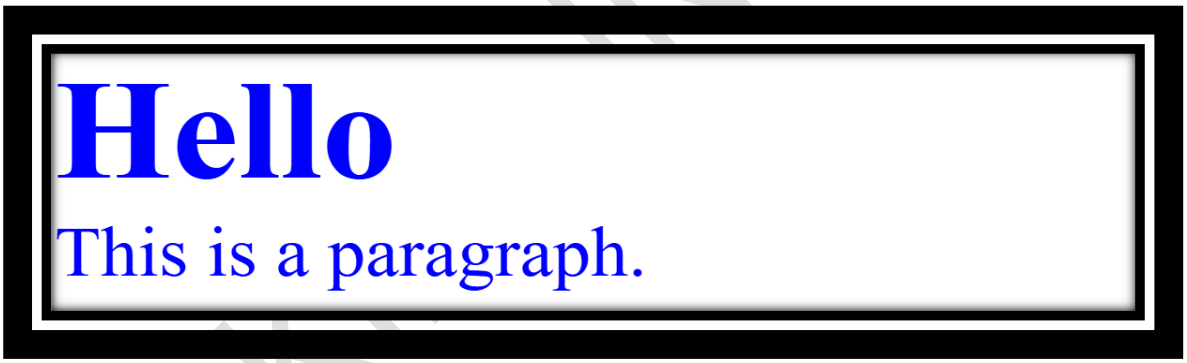
- Removes default margin & padding from every element.

ZED-KING INSTITUTE

- It can be written as * stand for **asterisk** or **star symbol**.
- Turns all text blue.
- The code inside the curly bracket apply to the all elements on the HTML page.

Selects all elements on the page

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  margin: 0;
  padding: 0;
  color: blue;
}
</style>
</head>
<body>
<h1>Hello</h1>
<p>This is a paragraph.</p>
</body>
</html>
```



2. ID Selector (#idname)

- Denoted by a hash (#) followed by ID name.
- Applies styles to one unique element only (ID must be unique in a page).
- An element can have only one ID.
- Used when you need to style a specific single element.
- Has higher priority than tag and class selectors.

#header targets an element with id="header".

- The <h1 id="header"> becomes purple & large.

- ID is unique, so only one element should use it.
- This selector is used to select the **id attribute** of an HTML element for selecting the specific element.
- Write as a **hash (#)** chracater.
- Don’t include any space between the **#** and the **id** of an element.

```
<html>
<head>
<style>
#header {
  color: purple;
  font-size: 24px;
}
</style>
</head>
<body>
<h1 id="header">This is a header</h1>
</body>
</html>
```



3. Tag Selector (Element Selector or Type Selector)

A tag selector (also called an element selector) in CSS is used to select all HTML elements of a specific type and apply styles to them. It targets the HTML tag directly.

Syntax:

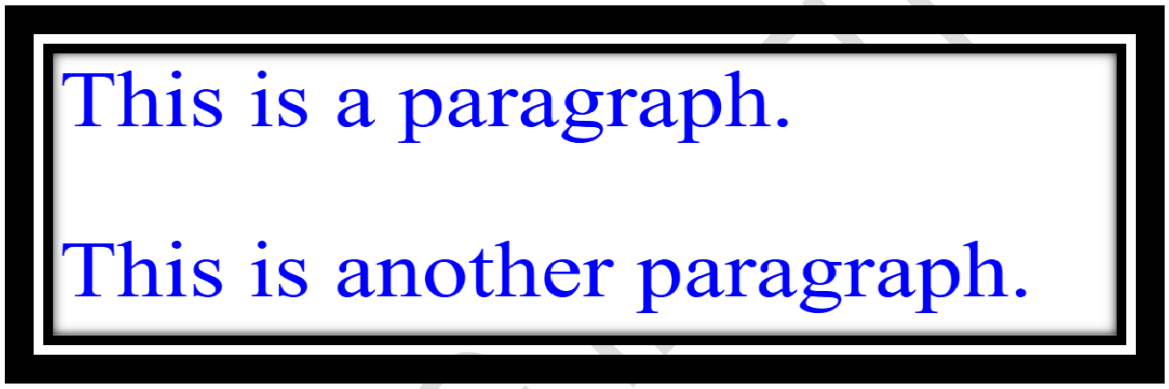
tagname {

property: value;

}

Example:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p {
      color: blue;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <p>This is a paragraph.</p>
  <p>This is another paragraph.</p>
</body>
</html>
```



Explanation:

- Here, p is the tag selector.
- It applies the styles (color: blue; font-size: 18px;) to all <p> elements in the HTML document.
- Targets elements by tag name.

4. Class Selector (.classname)

Here are the important points of Class Selector in brief:

- Denoted by a dot (.) followed by class name.
- Applies styles to all elements with that class.
- Multiple elements can share the same class.
- An element can have multiple classes.
- Allows reusable and flexible styling.

- Can be combined with other selectors for specific targeting.

Selects elements by class attribute.

.note {

background: yellow;

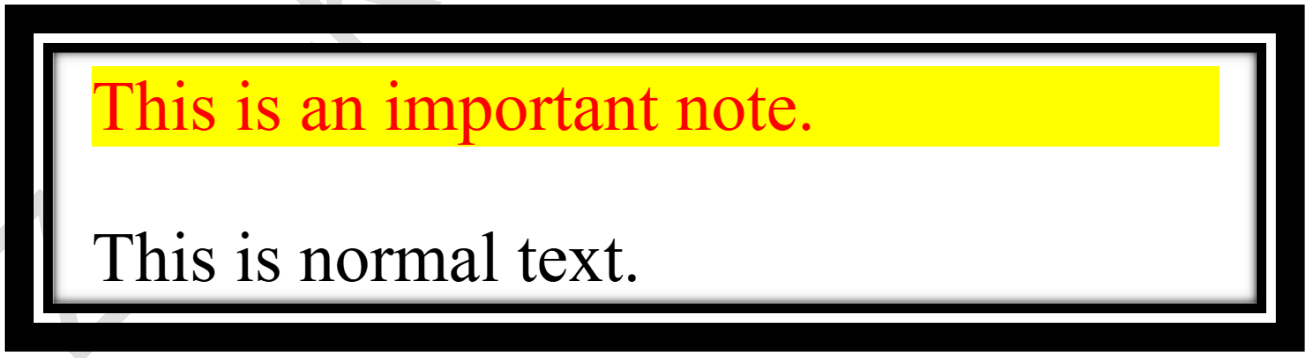
color: red;

}

.note means “select all elements with class note”.

- <p class="note"> will get yellow background & red text.
- Other <p> remain normal.

```
<!DOCTYPE html>
<html>
<head>
<style>
.note {
  background: yellow;
  color: red;
}
</style>
</head>
<body>
<p class="note">This is an important note.</p>
<p>This is normal text.</p>
</body>
</html>
```



5. Sub Selector:-

- A CSS selector can contain more than one simple selectors, it means that, it combines the other selectors.
- Between the simple selectors, we can include a combinator.

- There are four different cobminators in CSS:-

i). Descendant Selector(space)

ii). Child Selector(>)

iii). Adjacent Sibling Selector(+)

iv). General Sibling Selector(~)

There are details description below:-

i) Descendant Selector (space):-

- Selects **all elements inside another element** (nested at any level).
- Matches children, grandchildren, great-grandchildren, etc.

Example:

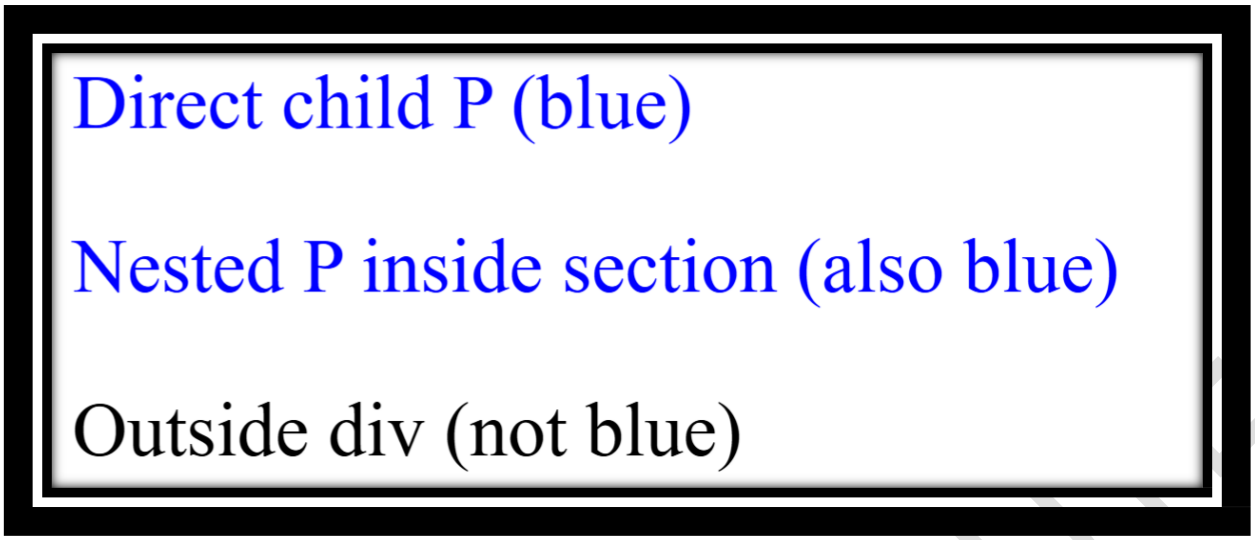
div p {

color: blue;

}

All <p> inside a <div> (no matter how deep) will become blue.

```
<html>
<head>
<style>
div p {
  color: blue;
}
</style>
</head>
<body>
  <div>
    <p>Direct child P (blue)</p>
    <section>
      <p>Nested P inside section (also blue)</p>
    </section>
  </div>
  <p>Outside div (not blue)</p>
</body>
</html>
```



ii) Child Selector (>)

- Selects only the **direct child** of a parent.
- Does not go deeper (grandchildren are not selected).

Example:

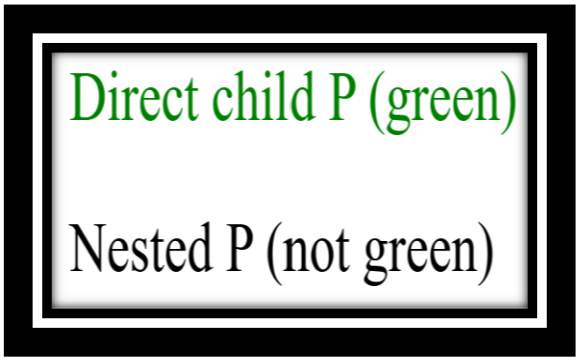
div > p {

color: green;

}

Only <p> that are **direct children** of will become green.

```
<html>
<head>
<style>
div > p {
  color: green;
}
</style>
</head>
<body>
  <div>
    <p>Direct child P (green)</p>
    <section>
      <p>Nested P (not green)</p>
    </section>
  </div>
</body>
</html>
```



KING INSTITUTE

iii) Adjacent Sibling Selector (+)

- Selects an element that is **immediately next to (just after)** another element, sharing the same parent.

Example:

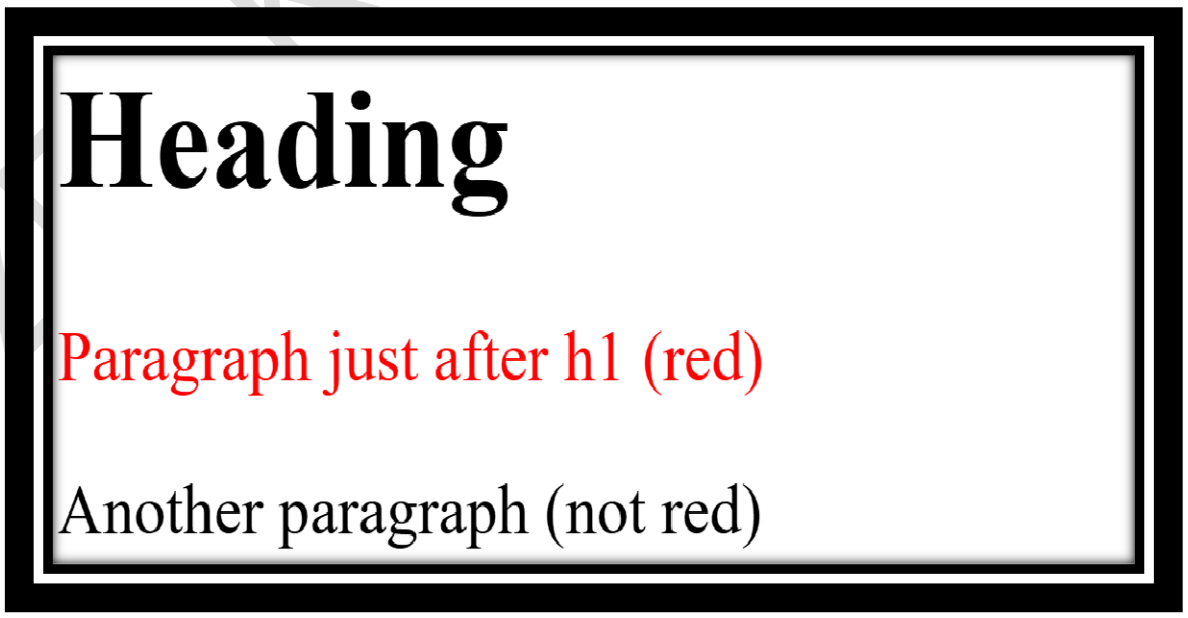
h1 + p {

color: red;

}

The <p> that comes **right after an <h1>** will become red.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 + p {
  color: red;
}
</style>
</head>
<body>
  <h1>Heading</h1>
  <p>Paragraph just after h1 (red)</p>
  <p>Another paragraph (not red)</p>
</body>
</html>
```



iv) General Sibling Selector (~)

- Selects **all siblings** after a specified element (not just the first).
- The general sibling combinator(~) seprates two selectors and matches the second element only if it follows the same
- They must have the same parent.

Example:

```
h1 ~ p {
  color: orange;
}
```

All <p> that come **after <h1> in the same parent** will become orange.

```
<html>
<head>
<style>
h1 ~ p {
  color: orange;
}
</style>
</head>
<body>
  <h1>Heading</h1>
  <p>First paragraph after h1 (orange)</p>
  <p>Second paragraph after h1 (orange)</p>
  <div>Other element</div>
  <p>Third paragraph after h1 (orange)</p>
</body>
</html>
```



6. Attribute Selector

1. `[attribute]`

Element jisme sirf attribute present ho.

Example: `input[required]`

2. `[attribute = value]`

Attribute ki value **exact match** ho.

Example: `input[type="text"]`

3. `[attribute ~= value]`

Attribute me **space-separated list** ho, usme value present ho.

Example: `div[title~="big"]`

4. `[attribute |= value]`

Attribute ki value **exact match** ya value- se start ho (with hyphen).

Example: `p[lang|="en"]`

5. `[attribute ^= value]`

Attribute ki value **value se start** ho.

Example: `a[href^="https"]`

6. `[attribute $= value]`

Attribute ki value **value se end** ho.

Example: `img[src$=".png"]`

7. `[attribute *= value]`

Attribute ki value ke andar **kahin bhi** value ho.

Example: `a[href*="google"]`

8. `:not([attribute])`

Jin elements me **attribute ना ho**, unko select kare.

Example: `input:not([type])`

Types of Attribute Selectors

1. [attr] → Selects elements with a given attribute

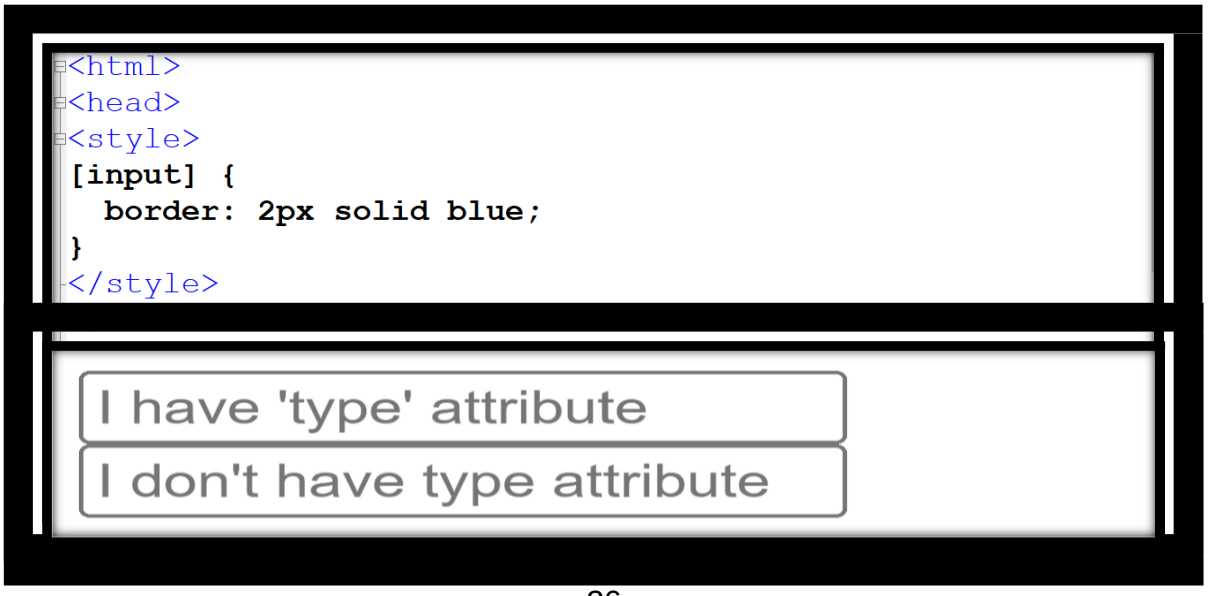
- Matches if the attribute is present, regardless of its value.

[input] {

border: 2px solid blue;

}

- Applies to all elements having an input attribute like <input>, <input type="text">, etc.
- First <input> gets blue border (because it has type).



2. [attr="value"] → Exact match

- Matches if the attribute value is exactly equal to the given value.

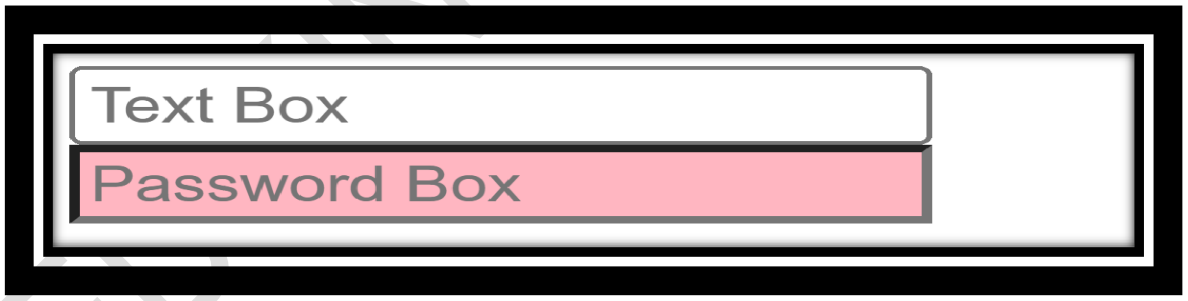
input[type="text"] {

background: lightyellow;

}

- Applies only to <input type="text">.
- Only password box is pink.

```
<html>
<head>
<style>
input[type="password"]
{
  background: lightpink;
}
</style>
</head>
<body>
<input type="text" placeholder="Text Box">
<input type="password" placeholder="Password Box">
</body>
```



3. [attr~="value"] → Contains word

- Matches if the attribute value is a space-separated list of words and one of them matches.

[class~="highlight"]

{

color: red;

}

- Applies to or <p class="box highlight">.
- Does not apply to <p class="highlighted">.
- Matches only if highlight is a separate word.

```
<html>
<head>
<style>
[class~="highlight"] {
  color: red;
}
</style>
</head>
<body>
<p class="highlight">This text is red</p>
<p class="box highlight">This is also red</p>
<p class="highlighted">This is NOT red</p>
</body>
</html>
```

This text is red

This is also red

This is NOT red

4. [attr|="value"] → Hyphen-separated value

- Matches if the attribute value is exactly equal to the given value OR starts with that value followed by a hyphen -.

[lang|="en"] {

font-style: italic;

}

- Matches lang="en" or lang="en-US".
- Does not match lang="english".

- Works for en and en-US, but not english.

```html
<html>
<head>
<style>
[lang|="en"] {
  font-style: italic;
}
</style>
</head>
<body>
<p lang="en">English Text</p>
<p lang="en-US">American English</p>
<p lang="english">Not Selected</p>
</body>
</html>
```



5. [attr^="value"] → Starts with

- Matches if the attribute value begins with the specified string.

```
a[href^="https"] {
  color: green;
}
```

- Matches <a href="https://example.com">.
- Does not match <a href="http://example.com">.
- Only https:// link is green.

```html
<html>
<head>
<style>
a[href^="https"] {
  color: green;
}
</style>
</head>
<body>
<a href="https://example.com">Secure Link</a>
<a href="http://example.com">Normal Link</a>
</body>
</html>
```



6. [attr$="value"] → Ends with

- Matches if the attribute value ends with the specified string.

```
a[href$=".pdf"]{
color: red;
}
```

- Matches <a href="file.pdf">.
- Does not match <a href="file.doc">.
- Only .pdf link turns red.

ZED-KING INSTITUTE

```html
<html>
<head>
<style>
a[href$=".pdf"] {
  color: red;
}
</style>
</head>
<body>
<a href="file.pdf">Download PDF</a>
<a href="file.doc">Download DOC</a>
</body>
</html>
```

Download PDF Download DOC

7. [attr*="value"] → Contains substring

- Matches if the attribute value contains the given substring.

a[href*="google"] {

color: orange;

}

- Matches <a href="https://google.com">.
- Matches <a href="search-google-result.html">.
- Matches any link containing google in href.

```html
<html>
<head>
<style>
a[href*="google"] {
  color: orange;
}
</style>
</head>
<body>
<a href="https://google.com">Google</a>
<a href="https://search-google-results.com">Search</a>
<a href="https://example.com">Example</a>
</body>
</html>
```

Google Search Example

7. Group Selector:-

- The **Group Selector** is used when you want to apply the **same CSS style** to **multiple elements at once**.
- It saves time and reduces code repetition.
- To group selectors, we use a **comma (,)** between them.

Syntax

selector1, selector2, selector3 {

property: value;

}

Example 1 – Basic

- Here h1, h2, and p all get **blue color** and **center alignment**.

```
<html>
<head>
<style>
h1, h2, p {
  color: blue;
  text-align: center;
}
</style>
</head>
<body>
<h1>Main Heading</h1>
<h2>Sub Heading</h2>
<p>This is a paragraph.</p>
</body>
</html>
```

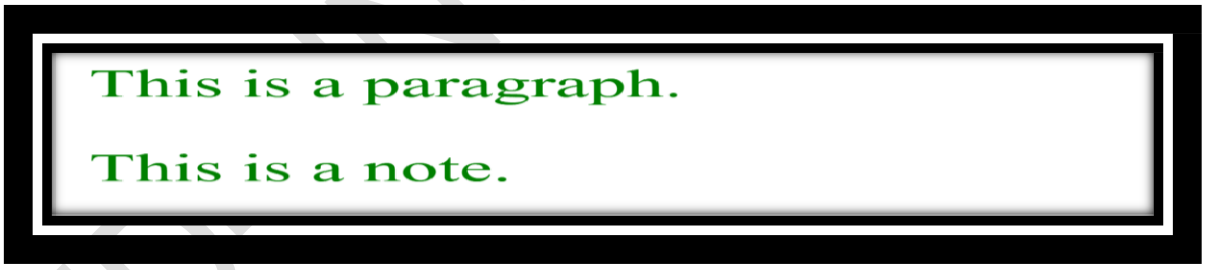
ZEAL TRAINING INSTITUTE



Example 2 – Class & Tag Together

- Both <p> and .note class get the same style.

```
<html>
<head>
<style>
p, .note {
  font-size: 18px;
  color: green;
}
</style>
</head>
<body>
<p>This is a paragraph.</p>
<div class="note">This is a note.</div>
</body>
</html>
```



Example 3 – Grouping with ID

- Header, Footer, and h3 all get a **lightgray background**.

```html
<html>
<head>
<style>
#header, #footer, h3 {
  background-color: lightgray;
  padding: 10px;
}
</style>
</head>
<body>
<div id="header">This is header</div>
<h3>Subheading</h3>
<div id="footer">This is footer</div>
</body>
</html>
```

This is header

Subheading

This is footer

4.4 CSS Properties

4.4.1 Background Properties

CSS Backgrosund Properties

CSS background properties are used to **set the background effects** (like color, image, repeat, position, attachment, size, etc.) of elements.

There are list of background properties:-

Controls **color, image, position, size**.

1. background-color

- Sets the background color of an element.
- Default: **transparent.**
- If not set, background-color is **transparent**.
- It applies only to the **content + padding** area of the element (not the margin).

```
<html>
<head>
<style>
h1 {
  background-color: lightblue;
}
</style>
</head>
<body>
<h1>This heading has a lightblue background</h1>
</body>
</html>
```



Ways to Define Colors:-

1. Named Colors:-

- CSS has 140+ predefined color names.

2. HEX Code

- A 6-digit code representing RGB values in hexadecimal **combination of numbers (0–9) and letters (A–F)**.
- **Format: #RRGGBB**

#RRGGBB

- RR → Red value (00 to FF)
- GG → Green value (00 to FF)
- BB → Blue value (00 to FF)

Each pair is in hexadecimal (base 16), where:

- 00 = 0 (lowest intensity)
- FF = 255 (highest intensity)

Examples:

- #000000 → Black
- #FFFFFF → White
- #FF0000 → Red
- #00FF00 → Green
- #0000FF → Blue
- #FFFF00 → Yellow

Shortcut:

If all pairs repeat (like #FF0000 → #F00), you can use **3-digit HEX**.
Example:

- #FFF = White
- #000 = Black
- #F00 = Red

ZED-KING INSTITUTE

3. HEX Code Examples with RGB Conversion

HEX Code	RGB Value	Color
#000000	Rgb (0,0,0)	Black
#FFFFFF	Rgb (255,255,255)	White
#FF0000	Rgb (255,0,0)	Red
#00FF00	Rgb (0,255,0)	Green
#0000FF	Rgb (0,0,255)	Blue
#FFFF00	Rgb (255,255,0)	Yellow
#00FFFF	Rgb (0,255,255)	Cyan
#FF00FF	Rgb (255,0,255)	Magenta

4. Short HEX Code (3-Digit Form)

If the HEX code has repeating pairs, you can write it in **short form**.

- **#FF0000 → #F00 (Red)**
- **#FFFFFF → #FFF (White)**
- **#000000 → #000 (Black)**

In short:

- **HEX =** #RRGGBB
- **Range =** 00 to FF **per color channel**
- **Maximum colors = 16,777,216**

ZED-KING INSTITUTE

```html
<html>
<head>
<style>
p {
  background-color: #ff0000; /* red */
}
</style>
</head>
<body>
<p>This paragraph has a red background using HEX</p>
</body>
</html>
```

This paragraph has a red background using HEX

3. RGB Color

- Format: rgb(red, green, blue) → each value between 0–255.

```html
<html>
    <head>
        <title>CSS Background properties</title>
    </head>
    <style>
        div
        {
                background-color:rgb(0,255,0); /*Green code color*/
                width:100%;
                height:10%;
        }
    </style>
    <body>
        <div>This div is green background using rgb.</div>
    </body>
</html>
```

This div has a green background using RGB

4. RGBA (RGB + Alpha for Transparency)

- Format: rgba(r, g, b, a) → a is opacity (0 = fully transparent, 1 = fully visible).

Real-life examples:

div → Container box

- Like a **carton box** in which you keep many things together.
- Example: a box containing text + image + button.

span → Highlighter pen

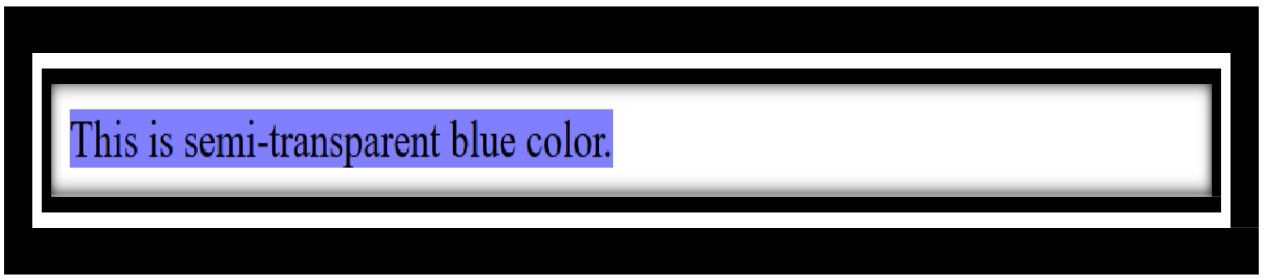
- Like a **highlighter** you use inside a notebook.
- You don't break the sentence, you just highlight a word.
- Example: making one word red inside a paragraph.

section → Chapter in a book

- Like a **chapter** in your school book .
- It organizes content into **meaningful parts** (Intro, About, Contact).
- **div** = Box for arranging things.
- **span** = Highlighter inside text.
- **section** = Chapter in the webpage

```
<html>
    <head>
        <title>CSS Background(RGBA) Example</title>
        <style>
            span {

                background-color:rgba(0, 0, 255, 0.5);
            }
        </style>
    </head>
    <body>
        <span>This is  semi-transparent blue color.</span>
    </body>
</html>
```

e.

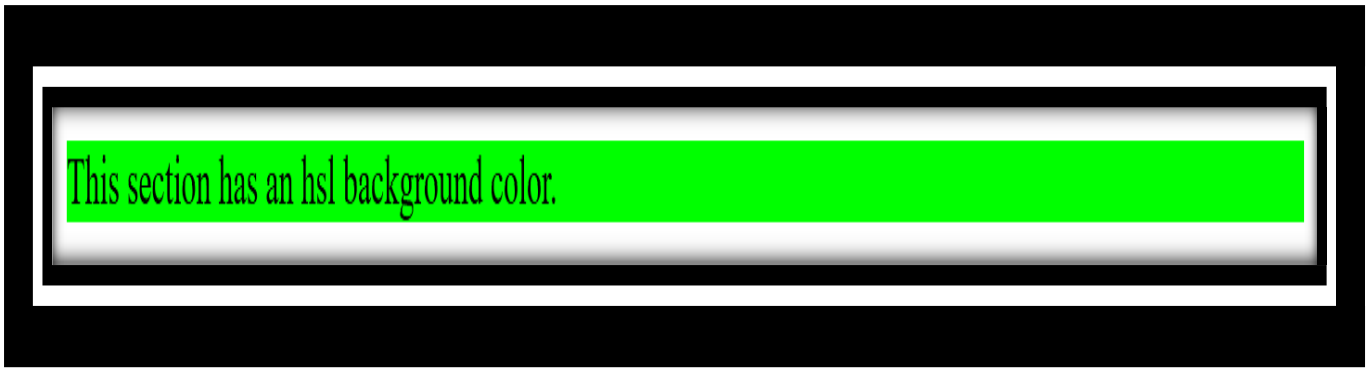


TRAINING INSTITUTE

5. HSL (Hue, Saturation, Lightness):-

- Format: hsl(hue, saturation%, lightness%)
  - Hue: 0–360 (color wheel: 0=red, 120=green, 240=blue).
  - Saturation: 0% (gray) → 100% (full color).
  - Lightness: 0% (black) → 50% (normal) → 100% (white)

```
<html>
    <head>
        <title>CSS Background(RGBA) Example</title>
        <style>
            section {

                background-color:hsl(120, 100%, 50%); /* Pure Green */
            }
        </style>
    </head>
    <body>
        <section>This section has an hsl background color.</section>
    </body>
</html>
```



3. HSLA (HSL + Alpha)

- Adds transparency to HSL colors.

```
<html>
    <head>
        <title>CSS background color with HSLA</title>
        <style>
            span{
            background-color:hsla(240, 100%, 50%,0.5);
            }
        </style>
    </head>
    <body>
        <span>This is my paragrapgh 2</span>
    </body>
</html>
```



Background-image

i. **Background color**
ii. **Background image**
iii. **Background position**
iv. **Background repeat**
v. **Background size**
vi. **Background attachment**
vii. **Background origin**
viii. **Background clip**
ix. **Background blend-mode**
x. **Background gradients (linear, radial, conic)**

1. background-color

- This property is already defined above.

ZED-KING INSTI

2. background-image

- Sets one or multiple background images.

```
div {
  background-image: url("flower.jpg");
}
```

```
<html>
<head>
<style>
div {
  width: 300px;
  height: 150px;
  background-image: url("https://picsum.photos/300/150");
}
</style>
</head>
<body>
<div>Background Image Example</div>
</body>
</html>
```



3. Background-repeat

Controls repeating of image.

```
div {
  background-repeat: repeat;   /* default */
  background-repeat: no-repeat;
```

```
  background-repeat: repeat-x;
  background-repeat: repeat-y;
}
```

```
<html>
<head>
<style>
div {
  width: 300px;
  height: 150px;
  background-image: url("https://picsum.photos/50");
  background-repeat: repeat-x;  /* Try repeat-y, no-repeat */
}
</style>
</head>
<body>
<div>Background Repeat Example</div>
</body>
</html>
```



4. background-position

Sets position of image.

```
div {
  background-position: left top;
  background-position: center;
  background-position: 50% 50%;
  background-position: 20px 40px;
}
```

```html
<html>
<head>
<style>
div {
  width: 300px;
  height: 150px;
  background-image: url("https://picsum.photos/100");
  background-repeat: no-repeat;
  background-position: center;
}
</style>
</head>
<body>
<div>Background Position Example</div>
</body>
</html>
```



5. background-size

Controls size of image.

```
div {
  background-size: auto;       /* default */
  background-size: cover;      /* cover full */
  background-size: contain;    /* fit inside */
  background-size: 100px 200px;
}
```

ZED-KING

```html
<html>
<head>
<style>
div {
  width: 300px;
  height: 150px;
  background-image: url("https://picsum.photos/400/300");
  background-repeat: no-repeat;
  background-size: cover;  /* Try contain or 100px 100px */
}
</style>
</head>
<body>
<div>Background Size Example</div>
</body>
</html>
```



6. background-attachment

Defines scroll behavior.

```
div {
  background-attachment: scroll;  /* default */
  background-attachment: fixed;
  background-attachment: local;
}
```

```html
<html>
<head>
<style>
body
{
background-image:url("https://picsum.photos/1200/800");
background attachment:fixed;
background-size:cover;
}
</style>
</head>
<body>

<h1>Scroll Down</h1>
<p style="height:1000px;">Background stays fixed while scrolling</p>

</body>
</html>
```



7. background-origin

Defines where the background is positioned (relative to border, padding, or content).

```
div {
  background-origin: border-box;
  background-origin: padding-box; /* default */
  background-origin: content-box;
}
```

```html
<html>
<head>
<style>
.box {
  width: 300px;
  height: 150px;
  border: 20px solid rgba(0,0,0,0.5);
  padding: 30px;
  margin: 20px;
  background: url("https://picsum.photos/id/1015/400/300") no-repeat;
  background-size: 100px 100px;
  color: white;
  font-weight: bold;
  text-align: center;
  line-height: 90px;
}

/* Different origins */
.border-box  { background-origin: border-box; }
.padding-box { background-origin: padding-box; }
.content-box { background-origin: content-box; }
</style>
</head>
<body>

<h2>Background-Origin Examples</h2>

<div class="box border-box">border-box</div>
<div class="box padding-box">padding-box (default)</div>
<div class="box content-box">content-box</div>

</body>
</html>
```



8. background-clip

Defines how far the background extends.

```
div {
  background-clip: border-box;   /* default */
  background-clip: padding-box;
  background-clip: content-box;
  background-clip: text;        /* special → background inside text */
}
```

ZED-KING INS

```html
<html>
<head>
<style>
h1 {
  font-size: 50px;
  background: url("https://picsum.photos/600/400") no-repeat center;
  background-size: cover;
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
}
</style>
</head>
<body>
<h1>Background Clip Example</h1>
</body>
</html>
```



9. background-blend-mode

Specifies how background images/colors blend.

```
div {
  background-image: url("flower.jpg"), url("texture.png");
  background-blend-mode: multiply;
}
```

Modes: multiply, screen, overlay, darken, lighten, difference, exclusion etc.

ZED-KING INSTITUTE

```html
<html>
<head>
<style>
.blend-box {
  width: 500px;
  height: 300px;
  border: 2px solid black;

  background:
    url("https://picsum.photos/id/1015/500/300") center/cover no-repeat,
    linear-gradient(to right, rgba(255,0,0,0.6), rgba(0,0,255,0.6));

  /* Blend mode applied */
  background-blend-mode: multiply;
}
</style>
</head>
<body>

<h2>Background Blend Mode Example</h2>
<div class="blend-box"></div>

</body>
</html>
```

10. Multiple Backgrounds

```
div {
  background-image: url("top.png"), url("bg.jpg");
  background-repeat: no-repeat, repeat;
  background-position: top right, center;
}
```

```html
<html>
<head>
<style>
div {
  width: 400px;
  height: 300px;
  border: 2px solid black;

  /* Multiple backgrounds */
  background:
    url("https://picsum.photos/id/1015/400/300") no-repeat center/cover,
    url("https://www.transparenttextures.com/patterns/wood-pattern.png") repeat,
    linear-gradient(to bottom right, rgba(255,0,0,0.3), rgba(0,0,255,0.3));
}
</style>
</head>
<body>

<h2>Multiple Background Example</h2>
<div></div>

</body>
</html>
```

11. Gradients as Background

- Gradients are treated as background images.
- Types: linear-gradient, radial-gradient, conic-gradient, repeating gradients.

(a) Linear Gradient

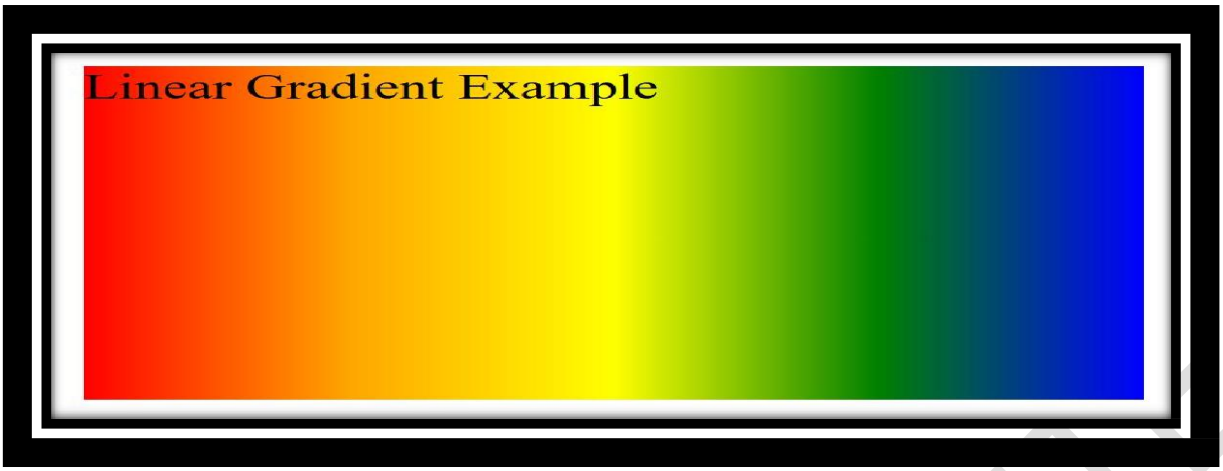
Background color transition in straight line.

```
div {
  background-image: linear-gradient(red, yellow);
}
```

With direction:

```
div {
  background-image: linear-gradient(to right, red, orange, yellow);
  background-image: linear-gradient(45deg, blue, green);
}
```

```
<html>
<head>
<style>
div {
  width: 300px;
  height: 150px;
  background-image: linear-gradient(to right, red, orange, yellow, green, blue);
}
</style>
</head>
<body>
<div>Linear Gradient Example</div>
</body>
</html>
```



(b) Radial Gradient

Colors spread in a circle/ellipse.

```
div {
  background-image: radial-gradient(circle, red, yellow, green);
}
```

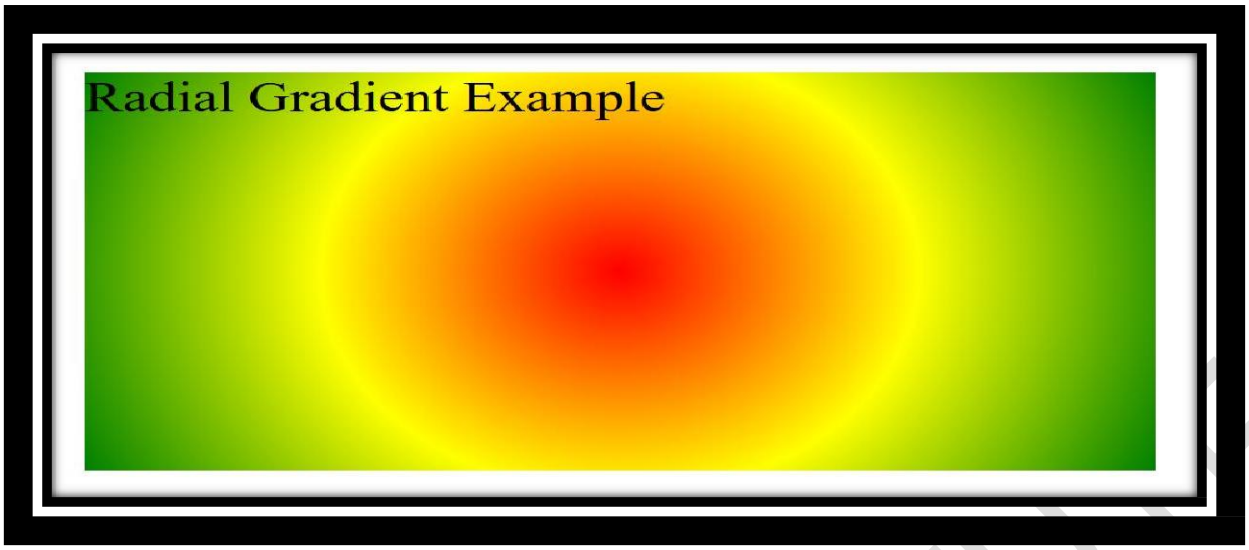
Shapes: circle, ellipse

Size keywords: closest-side, farthest-side, closest-corner, farthest-corner

```
div {
  background-image: radial-gradient(closest-side, red, yellow, green);
}
```

```
<html>
<head>
<style>
div {
  width: 300px;
  height: 150px;
  background-image: radial-gradient(circle, red, yellow, green);
}
</style>
</head>
<body>
<div>Radial Gradient Example</div>
</body>
</html>
```

KING INSTITUTE



(c) Conic Gradient (CSS3+)

Colors spread around a center point like a pie chart.

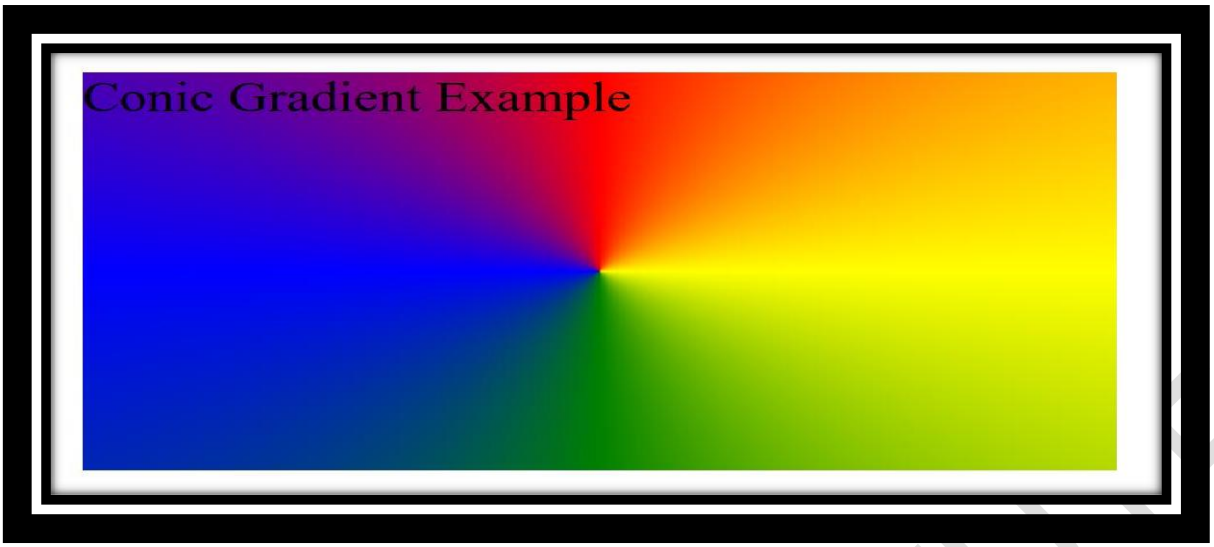
```
div {
  background-image: conic-gradient(red, yellow, green, blue);
}
```

With angle & position:

```
div {
  background-image: conic-gradient(from 90deg at 50% 50%, red, yellow, green);
}
```

```
<html>
<head>
<style>
div {
  width: 300px;
  height: 150px;
  background-image: conic-gradient(red, yellow, green, blue, red);
}
</style>
</head>
<body>
<div>Conic Gradient Example</div>
</body>
</html>
```

TALKING INSTITUTE

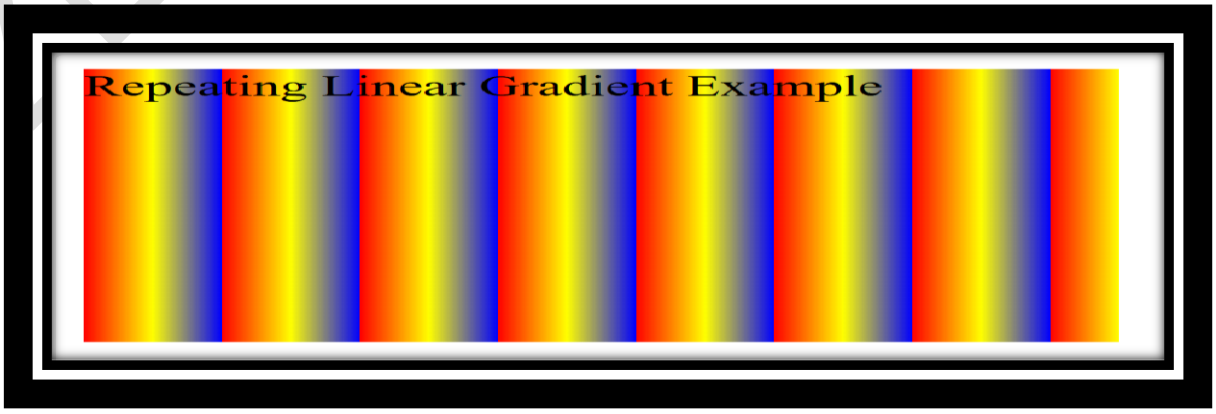


(d) Repeating Linear Gradients

Gradients that repeat like patterns.

```
div {
  background-image: repeating-linear-gradient(to right, red, yellow 20px);
}
```

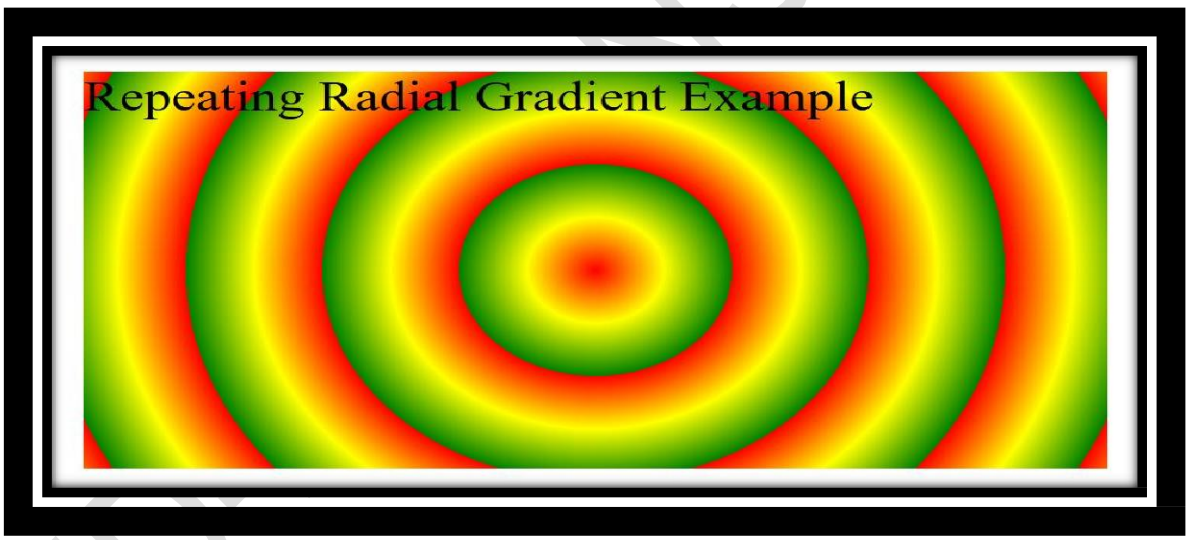
```
<html>
<head>
<style>
div {
  width: 300px;
  height: 150px;
  background-image: repeating-linear-gradient(to right, red 0px, yellow 20px,
}
</style>
</head>
<body>
<div>Repeating Linear Gradient Example</div>
</body>
</html>
```



(e) Repeating Radial Gradients

```
div {
  background-image: repeating-radial-gradient(circle, red, yellow 20px, green 40px);
}
```

```
<html>
<head>
<style>
div {
  width: 300px;
  height: 150px;
  background-image: repeating-radial-gradient(circle, red 0px, yellow 20px, green 40px);
}
</style>
</head>
<body>
<div>Repeating Radial Gradient Example</div>
</body>
</html>
```



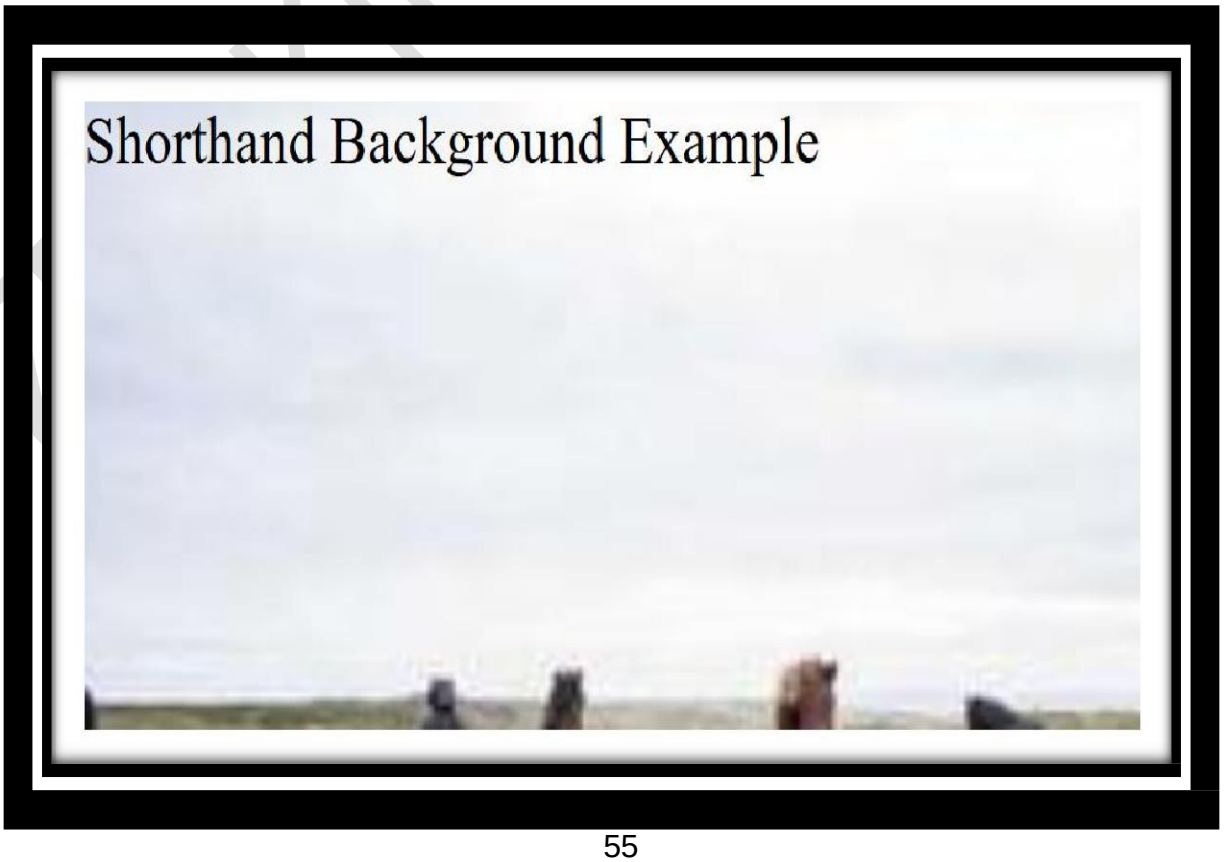
12. Special Values

- inherit → Gets background from parent
- initial → Resets to default
- unset → Acts as inherit/initial

13. Shorthand: background

Combine all in one line:

```
<html>
<head>
<style>
div {
  width: 300px;
  height: 150px;
  background: url("https://picsum.photos/300/150") no-repeat center/cover fixed;
}
</style>
</head>
<body>
<div>Shorthand Background Example</div>
</body>
</html>
```



Quick Summary Table

Property	Purpose
background-color	Sets color
background-image	Image or gradient
background-repeat	Repeat control
background-position	Image position
background-size	Image size
background-attachment	Fixed/scroll
background-origin	Start area
background-clip	Painting area
background-blend-mode	Blend effect
background	Shorthand
Gradients	linear, radial, conic, repeating

ZED-KING INSTITUTE

4.4.2 Block Properties

Important:

- display: block/inline/none → block or inline element.
- visibility: hidden/visible → hide/show element.
- overflow: hidden/scroll/auto → handle extra content.
- Block = takes **full width**.

Block Elements

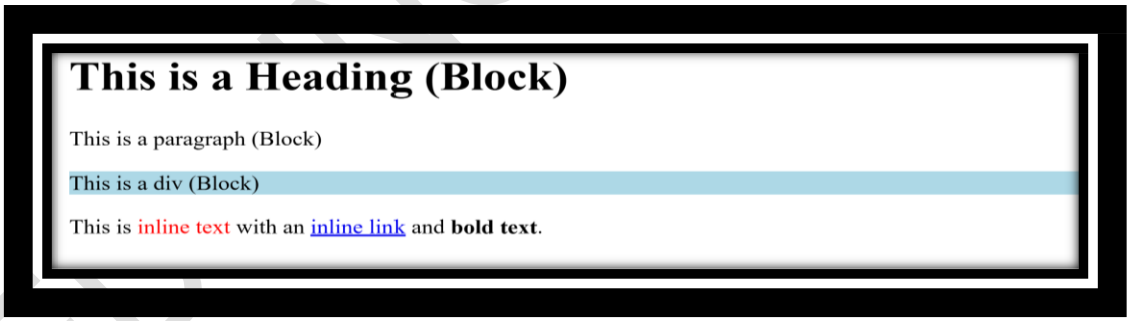
- **Start on a new line** (take up full width of the page/container).
- Always stretch horizontally as far as they can.
- Can contain **other block elements** and inline elements.
- Example: , <p>, <h1>, <section>, <article>, <ul>, <li> etc.

Inline Elements

- **Do not start on a new line** (stay within the same line).
- Only take up as much **width as their content**.
- Cannot contain block elements (only inline elements and text).
- Example: , <a>, , , <img>, <label> etc.

Example Code

```html
<html>
<head>
  <title>Block vs Inline</title>
</head>
<body>

  <!-- Block Elements -->
  <h1>This is a Heading (Block)</h1>
  <p>This is a paragraph (Block)</p>
  <div style="background: lightblue;">This is a div (Block)</div>

  <!-- Inline Elements inside a paragraph -->
  <p>
    This is <span style="color: red;">inline text</span>
    with an <a href="#">inline link</a>
    and <strong>bold text</strong>.
  </p>

</body>
</html>
```



Output Understanding

- The <h1>, <p>, each come on a **new line**.
- The , <a>, stay **inside the same line** within the paragraph.

```html
<html>
<head>
  <title>Block Properties Example</title>
  <style>
    div {
      display: block;
      width: 500px;
      height: 100px;
      background-color: lightblue;
      overflow: scroll;
    }
    p {
      visibility: hidden;
    }
  </style>
</head>
<body>
  <div>
    This is a block element with scroll overflow.
    <p>This paragraph is hidden.</p>
  </div>
</body>
</html>
```

4.4.3 Box Properties (Box Model)

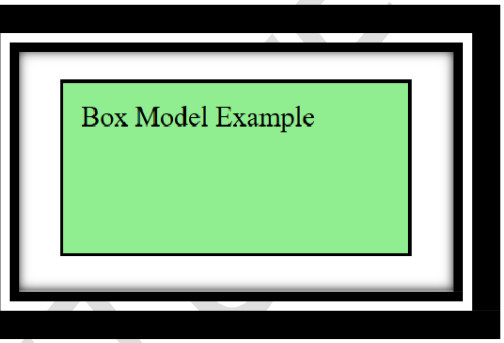
Important:

- width / height → element size
- padding → space inside
- margin → space outside
- border → border style/color/width
- box-sizing: border-box → includes border/padding in width

Margin → Border → Padding → Content

ZED-KING INSTITUTE

```html
<html>
<head>
  <title>Box Properties Example</title>
  <style>
    div {
      width: 200px;
      height: 100px;
      padding: 10px;
      margin: 20px;
      border: 2px solid black;
      box-sizing: border-box;
      background-color: lightgreen;
    }
  </style>
</head>
<body>
  <div>
    Box Model Example
  </div>
</body>
</html>
```



4.4.4 List Properties

Important:

- list-style-type: disc/circle/square/decimal → bullet/number type
- list-style-position: inside/outside → bullet inside/outside
- list-style-image: url('img.png') → custom bullet

ul { list-style-type: square;

ol { list-style-type: upper-roman; }

Controls bullet/number styles.

```html
<html>
<head>
  <title>List Properties Example</title>
  <style>
    ul {
      list-style-type: square;
      list-style-position: inside;
    }
  </style>
</head>
<body>
  <ul>
    <li>First Item</li>
    <li>Second Item</li>
    <li>Third Item</li>
  </ul>
</body>
</html>
```



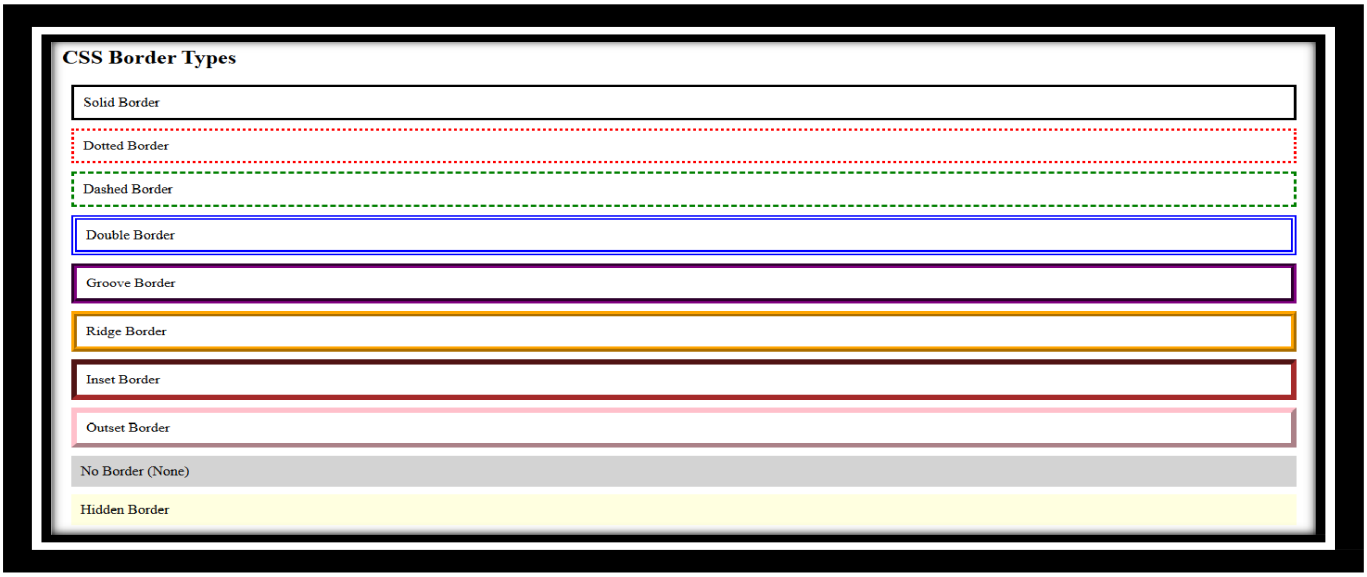
4.4.5 Border Properties

Important:

- border-width → thickness
- border-style → solid/dashed/dotted
- border-color → color
- border-radius → rounded corners
- border → shorthand (width style color)

Example

```html
<html>
<head>
   <title>CSS Border Types Example</title>
   <style>
      .solid { border: 3px solid black; padding: 10px; margin: 10px; }
      .dotted { border: 3px dotted red; padding: 10px; margin: 10px; }
      .dashed { border: 3px dashed green; padding: 10px; margin: 10px; }
      .double { border: 6px double blue; padding: 10px; margin: 10px; }
      .groove { border: 6px groove purple; padding: 10px; margin: 10px; }
      .ridge { border: 6px ridge orange; padding: 10px; margin: 10px; }
      .inset { border: 6px inset brown; padding: 10px; margin: 10px; }
      .outset { border: 6px outset pink; padding: 10px; margin: 10px; }
      .none { border: none; padding: 10px; margin: 10px; background: lightgray; }
      .hidden { border: hidden; padding: 10px; margin: 10px; background: lightyellow; }
   </style>
</head>
<body>
   <h2>CSS Border Types</h2>

   <div class="solid">Solid Border</div>
   <div class="dotted">Dotted Border</div>
   <div class="dashed">Dashed Border</div>
   <div class="double">Double Border</div>
   <div class="groove">Groove Border</div>
   <div class="ridge">Ridge Border</div>
   <div class="inset">Inset Border</div>
   <div class="outset">Outset Border</div>
   <div class="none">No Border (None)</div>
   <div class="hidden">Hidden Border</div>
</body>
</html>
```

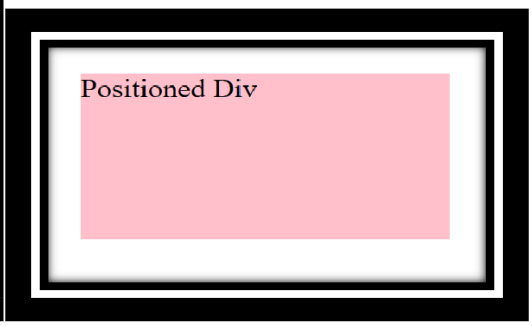


4.4.6 Positioning Properties

CSS Positioning Properties

- position: static/relative/absolute/fixed/sticky

```html
<html>
<head>
  <title>Positioning Properties Example</title>
  <style>
    div {
      position: relative;
      top: 20px;
      left: 30px;
      width: 200px;
      height: 100px;
      background-color: pink;
    }
  </style>
</head>
<body>
  <div>
    Positioned Div
  </div>
</body>
</html>
```

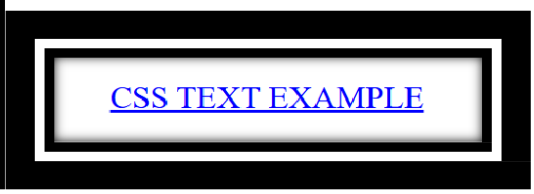


4.4.7 Text Properties

Important:

- color → text color
- text-align → left/center/right/justify
- text-decoration → underline/line-through/none
- text-transform → uppercase/lowercase/capitalize
- line-height → space between l.5

```html
<html>
<head>
  <title>Text Properties Example</title>
  <style>
    p {
      color: blue;
      text-align: center;
      text-transform: uppercase;
      text-decoration: underline;
      line-height: 1.5;
    }
  </style>
</head>
<body>
  <p>
    CSS Text Example
  </p>
</body>
</html>
```

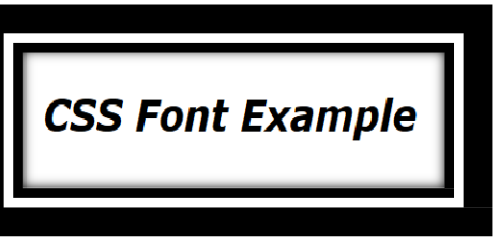


4.4.8 Font Properties

Important:

- font-family → font type
- font-size → size
- font-style → normal/italic/oblique
- font-weight → normal/bold
- font → shorthand (style weight size family)

```
<html>
<head>
  <title>Font Properties Example</title>
  <style>
    p {
      font-family: Verdana, sans-serif;
      font-size: 18px;
      font-style: italic;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <p>
    CSS Font Example
  </p>
</body>
</html>
```



4.4.9 Table Properties

Important:

- border-collapse: collapse/separate → merge/separate borders
- border-spacing → space between cells
- caption-side: top/bottom → caption position
- table-layout: auto/fixed → table width handling
- empty-cells: show/hide → show/hide empty cells

ZED-KING INSTITUTE

```html
<html>
<head>
  <title>Table Properties Example</title>
  <style>
    table {
      border: 2px solid black;
      border-collapse: collapse;
      width: 50%;
    }
    th, td {
      border: 1px solid black;
      padding: 10px;
      text-align: center;
    }
    caption {
      caption-side: top;
      font-weight: bold;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <table>
    <caption>Student Marks</caption>
    <tr>
      <th>Name</th>
      <th>Marks</th>
    </tr>
    <tr>
      <td>Geeta</td>
      <td>95</td>
    </tr>
    <tr>
      <td>Priya</td>
      <td>88</td>
    </tr>
  </table>
</body>
</html>
```

Student Marks

Name	Marks
Geeta	95
Priya	88

4.4.10 CSS Menu Design

1. Horizontal Menu

- Items are displayed in a **row (left to right).**
- ul li { display: inline; } → makes the menu horizontal.
- a tag styled with padding + background color → looks like buttons.
- a:hover → changes background when mouse moves over it.

HORIZONTAL MENU Example Code:

```html
<html>
<head>
  <title>Horizontal Menu Example</title>
  <style>
    /* Remove list style and padding */
    ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
      background-color: #333;
    }

    /* Display list items side by side */
    ul li {
      display: inline;
    }

    /* Style for links */
    ul li a {
      color: white;
      text-decoration: none;
      padding: 14px 20px;
      display: inline-block;
    }

    /* Hover effect */
    ul li a:hover {
      background-color: #111;
    }
  </style>
</head>
<body>

<h2>Simple Horizontal Menu</h2>

<ul>
  <li><a href="#">Home</a></li>
  <li><a href="#">About</a></li>
  <li><a href="#">Services</a></li>
  <li><a href="#">Contact</a></li>
</ul>

</body>
</html>
```



2. Vertical Menu:-

- Items are displayed **top to bottom.**

VERTICAL MENU Example Code:

```
<html>
<head>
  <title>Vertical Menu Example</title>
  <style>
    /* Remove default list style */
    ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
      width: 200px; /* fixed width for menu */
      background-color: #333;
    }

    /* Each list item displayed as block (one below another) */
    ul li a {
      display: block;          /* makes the links full width */
      color: white;
      padding: 12px;
      text-decoration: none;  /* remove underline */
    }

    /* Change color on hover */
    ul li a:hover {
      background-color: #111;
    }
  </style>
</head>
<body>

<h2>Simple Vertical Menu</h2>

<ul>
  <li><a href="#">Home</a></li>
  <li><a href="#">About</a></li>
  <li><a href="#">Services</a></li>
  <li><a href="#">Contact</a></li>
</ul>

</body>
</html>
```



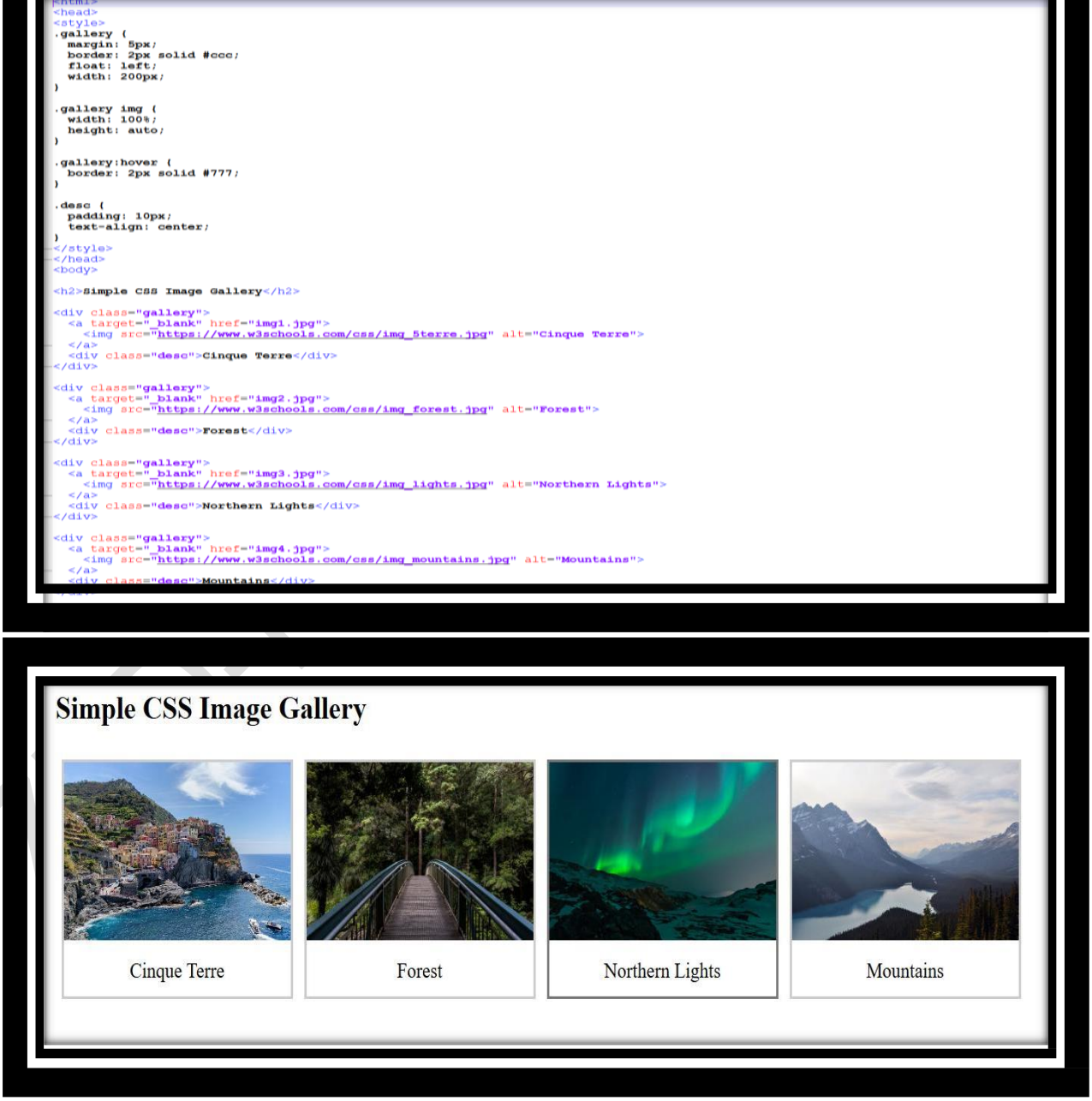
4.4.11 CSS Image Gallery

CSS Image Gallery:-

- An **image gallery** in CSS is a collection of images arranged in a **grid or layout**, often styled to look attractive and user-friendly.
- It is commonly used for **portfolios, e-commerce websites, and photo albums**.
- **Images arranged in rows & columns**.

Example 1: Basic Image Gallery (Using Float)

```
<html>
<head>
<style>
.gallery {
  margin: 5px;
  border: 2px solid #ccc;
  float: left;
  width: 200px;
}

.gallery img {
  width: 100%;
  height: auto;
}

.gallery:hover {
  border: 2px solid #777;
}

.desc {
  padding: 10px;
  text-align: center;
}
</style>
</head>
<body>

<h2>Simple CSS Image Gallery</h2>

<div class="gallery">
  <a target="_blank" href="img1.jpg">
    <img src="https://www.w3schools.com/css/img_5terre.jpg" alt="Cinque Terre">
  </a>
  <div class="desc">Cinque Terre</div>
</div>

<div class="gallery">
  <a target="_blank" href="img2.jpg">
    <img src="https://www.w3schools.com/css/img_forest.jpg" alt="Forest">
  </a>
  <div class="desc">Forest</div>
</div>

<div class="gallery">
  <a target="_blank" href="img3.jpg">
    <img src="https://www.w3schools.com/css/img_lights.jpg" alt="Northern Lights">
  </a>
  <div class="desc">Northern Lights</div>
</div>

<div class="gallery">
  <a target="_blank" href="img4.jpg">
    <img src="https://www.w3schools.com/css/img_mountains.jpg" alt="Mountains">
  </a>
  <div class="desc">Mountains</div>
</div>
```



Example 2: Responsive Image Gallery (Using CSS Grid)

```html
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.gallery {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 10px;
  padding: 10px;
}

.gallery img {
  width: 100%;
  height: 200px;
  object-fit: cover;
  border-radius: 10px;
  transition: transform 0.3s;
}

.gallery img:hover {
  transform: scale(1.1);
}
</style>
</head>
<body>

<h2>Responsive CSS Image Gallery (Grid)</h2>

<div class="gallery">
  <img src="https://www.w3schools.com/css/img_forest.jpg" alt="Forest">
  <img src="https://www.w3schools.com/css/img_lights.jpg" alt="Northern Lights">
  <img src="https://www.w3schools.com/css/img_mountains.jpg" alt="Mountains">
  <img src="https://www.w3schools.com/css/img_5terre.jpg" alt="Sea">
  <img src="https://www.w3schools.com/css/img_forest.jpg" alt="Nature">
</div>

</body>
</html>
```



