**Module 4 - CSS and CSS 3**

**Q-1. What are the benefits of using CSS?**

**Ans-** The main advantage of CSS is that style is applied consistently across variety of sites. It's what makes websites look good and stylish. Here are some benefits of using CSS

**Style and Design –** CSS control how to your website looks. You can change colors font size spacing and more

**Consistency -** With CSS, you can make sure that all the pages of your website have a consistent look

**Easy Updates –**  If you want to change the look of your website, you can do it quickly with CSS. You don't have to go through every page one by one, just update the CSS file, and it applies to the whole site

**Responsive Design -** CSS helps make your website look good on different devices like phones, tablets, and computers. It adjusts the layout and size of elements so they fit well on any screen size

**Q-2. What are the disadvantages of CSS?**

**Ans -** Disadvantage of css is

1. Complexity : CSS can become complex, especially in large projects, making it challenging to maintain and debug.

2. Browser Compatibility : Different browsers may interpret CSS rules differently, leading to inconsistencies in how a website looks across various platforms.

3. Limited Layout Control : CSS has limitations in terms of controlling complex layouts compared to other technologies like JavaScript or frameworks like Flexbox and Grid.

4. Performance Impact : Large CSS files or inefficient CSS selectors can impact page loading times and overall performance.

**Q-3. What is the difference between CSS2 and CSS3?**

**ANS-**

CSS2 and CSS3 are different versions of the Cascading Style Sheets (CSS) specification, each introducing new features and improvements over the previous version. Here are some key differences between CSS2 and CSS3:

**Features**: CSS3 introduced a bunch of new features that weren't available in CSS2. These include things like rounded corners, gradients, shadows, animations, and transitions

**Modularity**: CSS3 is more modular than CSS2, which means it's split into smaller modules that can be updated independently. This makes it easier to add new features and fixes without having to update the entire CSS specification.

**Compatibility**: CSS3 isn't always fully supported by older web browsers, while CSS2 is more widely supported. This means that some of the fancy new features in CSS3 might not work in older browsers

**Evolution**: CSS3 is the newer version, so it builds upon CSS2 and includes improvements and new capabilities.

**Q-4. Name a few CSS style components**

**Ans –**

Here are a few style components along with example

**Color**: This component lets you choose the colors for different parts of your website, like text, backgrounds, borders, and more.

**Font**: With the font component, you can change the style, size, and typeface of the text on your website. It's like choosing the font for a book – whether it's fancy and decorative or simple and easy to read.

**Padding and Margin**: Padding adds space inside an element, while margin adds space outside of it. They help control the spacing between elements on your web page, like the distance between paragraphs or the edges of a box.

**Border**: The border component lets you add borders around elements, like boxes or images, and customize their style, width, and color.

**Background**: This component allows you to add background colors, images, or patterns to elements on your webpage

**Layout**: Layout components help you arrange and position elements on your webpage, like aligning them horizontally or vertically, creating columns, or making them stack on top of each other.

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        p {

  font-family: Arial, sans-serif;

  font-size: 16px;

  color: #333;

}

.background{

    background-color: yellow;

}

.border{

    border: 2px solid black;

}

    </style>

</head>

<body>

    <p>Hello, world!</p>

    <div class="background">thi is backgrond example </div>

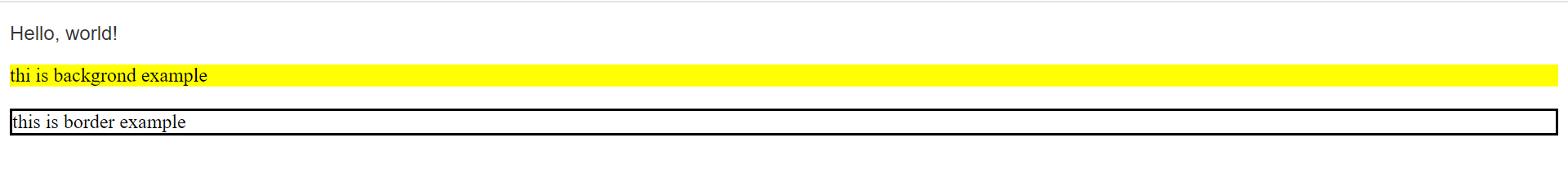
    <br>

    <div class="border">this is border example</div>

</body>

</html>

**Output:**

****

**Q-5. What do you understand by CSS opacity?**

CSS opacity refers to the transparency level of an element on a web page. It allows you to control how "see-through" an element is, ranging from completely transparent (invisible) to fully opaque (completely visible). The opacity property in CSS can be set using values between 0 and 1, where:

- 0 means the element is fully transparent (invisible).

- 1 means the element is fully opaque (completely visible).

**Code :**

Example: <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        .main{

            display: flex;

            gap: 40px;

            justify-content: center;

        }

        .exa{

            background-image: url(download.jpg);

            background-repeat: no-repeat;

            height: 200px;

            width: 200px;

            opacity: 0.5;

        }

        .exa1{

            background-image: url(download.jpg);

            background-repeat: no-repeat;

            height: 200px;

            width: 200px;

            opacity: 1;

        }

    </style>

</head>

<body>

    <div class="main">

        <div class="exa">

            <p>example of transperent box.</p>

        </div>

        <div class="exa1">

<h3>example of  fully opaque (completely visible).</h3>

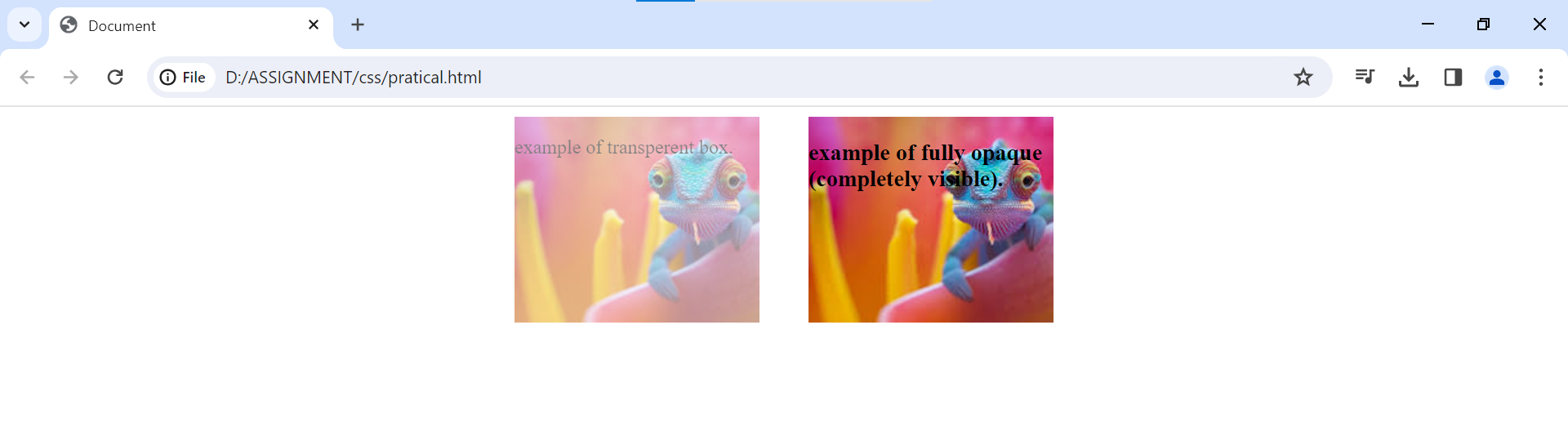
        </div>

    </div>

</body>

</html>

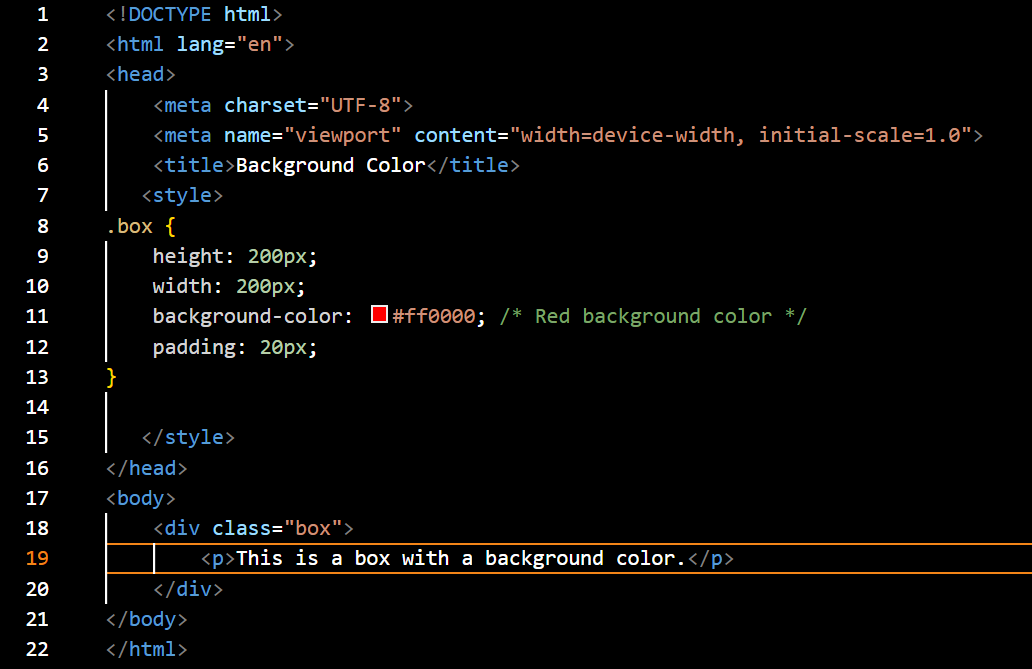
**Output :**

****

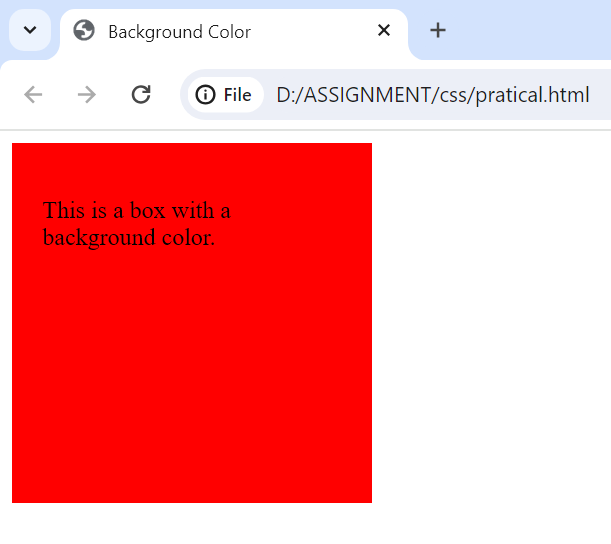
**Q-6. How can the background color of an element be changed?**

**ANS -** To change the background color of an element using CSS, you can use the background-color property. Here's an example:

**Code :**

****

**Output :**

****

**Q-7 How can image repetition of the backup be controlled?**

**ANS-** The **background-repeat property** in CSS is used to repeat the background image both horizontally and vertically. It also decides whether the background image will be repeated or not.

Here’s how you can control the repetition of a background image in CSS:

Using the background-repeat Property:

The background-repeat property in CSS dictates how a background image is repeated within the element’s background areas. Here are the available options:

**~ repeat (default):** Repeats the image both horizontally and vertically until the entire background area is filled.

**~ repeat-x :** Repeats the image only horizontally across the element’s width.

**~ repeat-y :** Repeats the image only vertically across the element’s height.

**~ no-repeat :** Displays the image only once at it’s original position (top-left corner by default).

**~ space :** Repeats the image with spaces in between, similar to repeat, but the spaces are sized to make the images fit exactly without any overflow.

**~ round :** Repeats the image with some parts potentially cut off to fill the entire background area.

• you can combine the background-repeat property with other background

properties like background-position to control the image’s placement within the element.

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Background Repetition Example</title>

   <style>

.background1 {

    width: 300px;

    height: 300px;

    background-image: url('2.jpeg');

    background-repeat: repeat;

}

.background2 {

    width: 300px;

    height: 300px;

    background-image: url('2.jpeg');

    background-repeat: no-repeat;

}

h1{

    color: white;

}

   </style>

</head>

<body>

        <div class="background1">

            <h1>Background repeat example</h1>

        </div>

        <div class="background2">

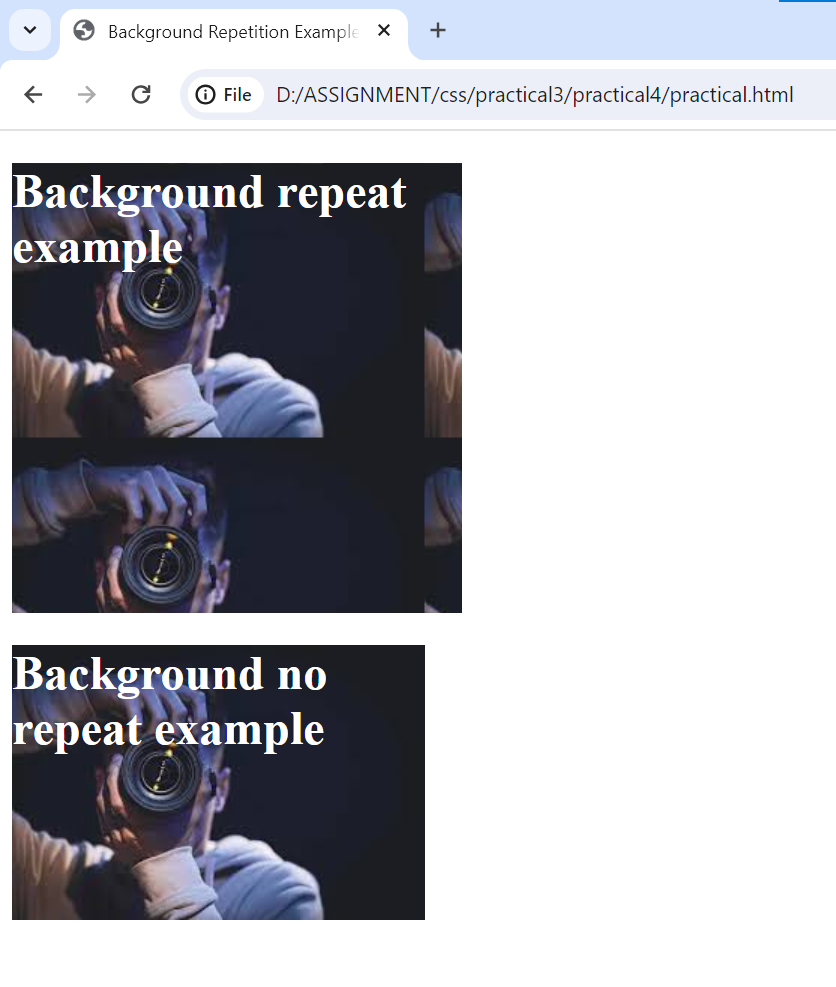
            <h1>Background no repeat example</h1>

        </div>

</body>

</html>

**Output:**

****

**Q-8. What is the use of the background-position property?**

**Ans** The background-position property define the original position of background image it contain keyword value top , bottom left right – specify the edge of the element’s box to which you want to place the background

The following background position value is :

**Property Value:**

left: It is used to set the image at the left position.

center: It is used to set the image at the horizontal center position.

right: It is used to set the image in the right position.

length: It is used to set the image at the horizontal position of the given length.

percentage: It is used to set the image at the horizontal position in terms of percentage height.

initial: It is used to set its default value. Its default value is 0%.

inherit: It inherits this property value from its parent elements.

Example :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content=

        "width=device-width, initial-scale=1.0">

    <title>CSS background-position-x Property</title>

    <style>

        body {

            text-align: center;

            background-image: url("bkg.png");

            background-size: 500px;

            background-repeat: no-repeat;

            background-attachment: fixed;

            background-position-y: 30%;

            background-position-x: 30%;

        }

        h1 {

            color: green;

        }

        img {

            width: 100px;

            height: 100px;

        }

    </style>

</head>

<body>

    <h1></h1>

    <h3>

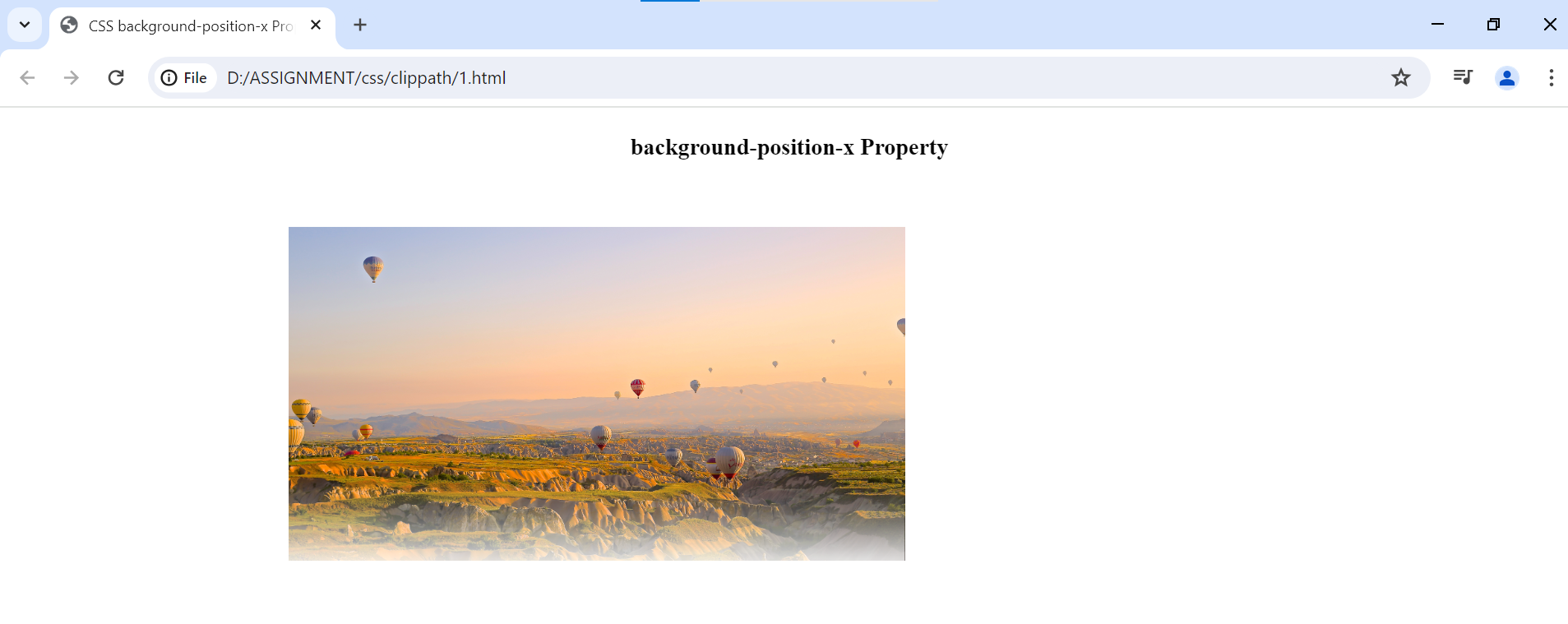
        background-position-x Property

    </h3>

</body>

</html>

**Output:**

****

**Q-9. Which property controls the image scroll in the background?**

**Ans.** The property that controls the scrolling behavior of a background image in CSS is the **background-attachment**  property. It determines whether the background image scrolls with the content of the element or remains fixed in place.

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Background Attachment Example</title>

<style>

  .container {

    width: 50%;

    height: 300px;

    background-image: url('2.jpeg');

    background-size: cover;

    background-repeat: no-repeat;

    background-attachment: fixed;

  }

  .content {

    padding: 20px;

    text-align: center;

    color: white;

  }

</style>

</head>

<body>

<div class="container">

  <div class="content">

    <h1>Welcome!</h1>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Architecto commodi delectus, fuga voluptatum sapiente ipsum minima quod ipsa veniam deserunt, optio cumque voluptates qui voluptate ad saepe culpa, quae quos reprehenderit laborum ducimus et tempore doloribus quo? Voluptas perferendis sint earum iste minus rem consectetur quasi ex vero placeat ullam possimus labore iusto voluptates maxime facere qui, aliquid eaque deserunt delectus, vel impedit eligendi dolorem laborum. Accusantium quis quam at, explicabo est eius earum quas. Illum beatae nam aspernatur vel, laudantium eius officia quae unde optio aperiam nulla ratione delectus quasi expedita incidunt totam minima quod voluptates? Doloribus molestias omnis iste, atque, quia cupiditate unde itaque eveniet sint molestiae rem nemo quae ipsam aliquam velit consequatur tempore dolorem nesciunt quod harum hic, corrupti sequi. Aperiam dolor rerum fugiat pariatur sint voluptatibus molestias quisquam? Voluptatum, aperiam quo, ratione deleniti natus odio dolorem minima facere quibusdam ducimus ipsum! Animi, suscipit fuga unde saepe maxime reiciendis possimus necessitatibus quisquam consequatur officiis natus quod incidunt minima perspiciatis sint consectetur ab quas! Similique possimus atque, id ipsum nobis eum corporis, nesciunt facilis, quos illo iste. Minima, eius laboriosam! Incidunt quae est id error adipisci maiores sapiente eveniet explicabo. Porro minus voluptate, saepe nobis reiciendis accusamus, iste blanditiis sit in dolore doloremque dicta temporibus eos? Sequi id fugit, quisquam eveniet eos dignissimos vero, nemo eligendi assumenda non excepturi dolorem autem quia tempore doloremque magnam atque quae exercitationem nisi. Recusandae quaerat laborum autem adipisci voluptatem delectus hic animi libero reiciendis ab. Voluptates soluta vero provident autem tempore! Doloremque et, voluptates iusto reprehenderit provident dolorum magni iste, deserunt aperiam modi cumque. Tempore voluptatum consequuntur incidunt non quia architecto impedit asperiores. Impedit ipsam vel, assumenda inventore iusto minima expedita, rem nulla numquam dolores quaerat eaque aliquid. Est repellendus eligendi, in quibusdam ad ea eos minus ratione laborum placeat consequatur, mollitia ullam quisquam maxime illum et. Harum sapiente possimus omnis molestiae itaque dolorem obcaecati ex eaque est a. Magni recusandae cupiditate tempora, architecto at maxime minus quod a perferendis alias fuga eveniet nihil ut nisi ullam esse minima laborum optio. Non quis optio molestias, molestiae quia delectus hic quas ipsam excepturi esse, similique porro deleniti laudantium, voluptatem rem illum aliquid? Facere, minus. Nostrum repellendus necessitatibus ipsa? Dolorum sint fugit recusandae harum eaque ipsum laboriosam saepe quam reiciendis sapiente excepturi, reprehenderit eveniet, deserunt minima cum, quasi exercitationem quidem libero totam quaerat a doloremque! Ab quod aliquam eaque dolor labore sequi. Optio possimus minima dolore dolor est obcaecati natus eaque, odit nisi ut perspiciatis facere unde quis cum sequi voluptatem ullam dignissimos atque, cumque id, quasi esse? Fugiat reiciendis facere voluptates quos, commodi vero deleniti repellat at, sint iusto corrupti debitis, consectetur hic sunt quisquam quae. Molestias aspernatur, doloremque optio ratione quaerat at sint, ipsam ipsum nemo nesciunt sapiente tenetur accusamus hic odit totam, ea consequuntur qui aliquam porro ab quibusdam esse iste reprehenderit? Doloribus eligendi mollitia iure fugit, accusantium inventore ratione aliquam dicta asperiores magnam. Deserunt saepe unde vitae, sit, rerum, magni nihil laudantium dolor aspernatur consequuntur sapiente impedit omnis quisquam.

    </p>

  </div>

</div>

<div class="other-content">

  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Rem id voluptas iste qui optio aspernatur laboriosam sed assumenda modi. Aut corporis cupiditate neque eveniet. Nam quaerat ea dolores porro iste voluptas laudantium aperiam pariatur labore eligendi ipsam deleniti, adipisci qui fugit rem quisquam nesciunt ipsa, asperiores maxime totam nisi dolor. Beatae consequatur aliquam quasi nesciunt ullam harum natus tenetur ex unde, ipsum excepturi optio dicta molestias cupiditate praesentium. Itaque quis reiciendis autem quos dolor facilis atque, officia officiis similique ex consequuntur cumque accusamus aperiam fugiat eum. A accusantium assumenda quam, aperiam minus eum optio qui quas culpa voluptas illo nulla corporis commodi quaerat cumque, neque dignissimos illum omnis nisi ea. Perspiciatis alias id fugit, earum amet mollitia. Sit repellat eligendi esse doloribus, magni architecto omnis perferendis ut asperiores amet saepe cupiditate delectus ab laudantium distinctio! Blanditiis voluptatibus necessitatibus totam, sit possimus deleniti magni repellat consequuntur nesciunt cumque cum veniam debitis corrupti cupiditate, veritatis quia commodi porro praesentium nobis ipsam maiores, odit recusandae eligendi. Fuga inventore, totam ut voluptates omnis recusandae, sint quo hic cupiditate veniam quidem cum blanditiis in maxime animi a ullam vitae quaerat. In molestiae exercitationem ipsam repellendus fugit placeat omnis odio aut, temporibus impedit dignissimos itaque nihil earum et quaerat enim repellat delectus officia cum? Saepe quas tempora esse dolore accusamus numquam fugit ullam at, explicabo voluptates vitae commodi. Tempora minima molestias nesciunt amet eius dolorum corporis culpa mollitia enim dolor explicabo asperiores reprehenderit iure, voluptates saepe hic. Ipsum facere suscipit laborum voluptatem error provident neque accusantium ab soluta recusandae laboriosam, reiciendis nisi iste excepturi consequuntur nostrum. Expedita cupiditate saepe molestias iste rerum eaque! Corrupti facilis libero maiores quam iste delectus eveniet cupiditate, nesciunt, modi fugit recusandae deleniti, praesentium

</div>

</body>

</html>

**Output :**

****

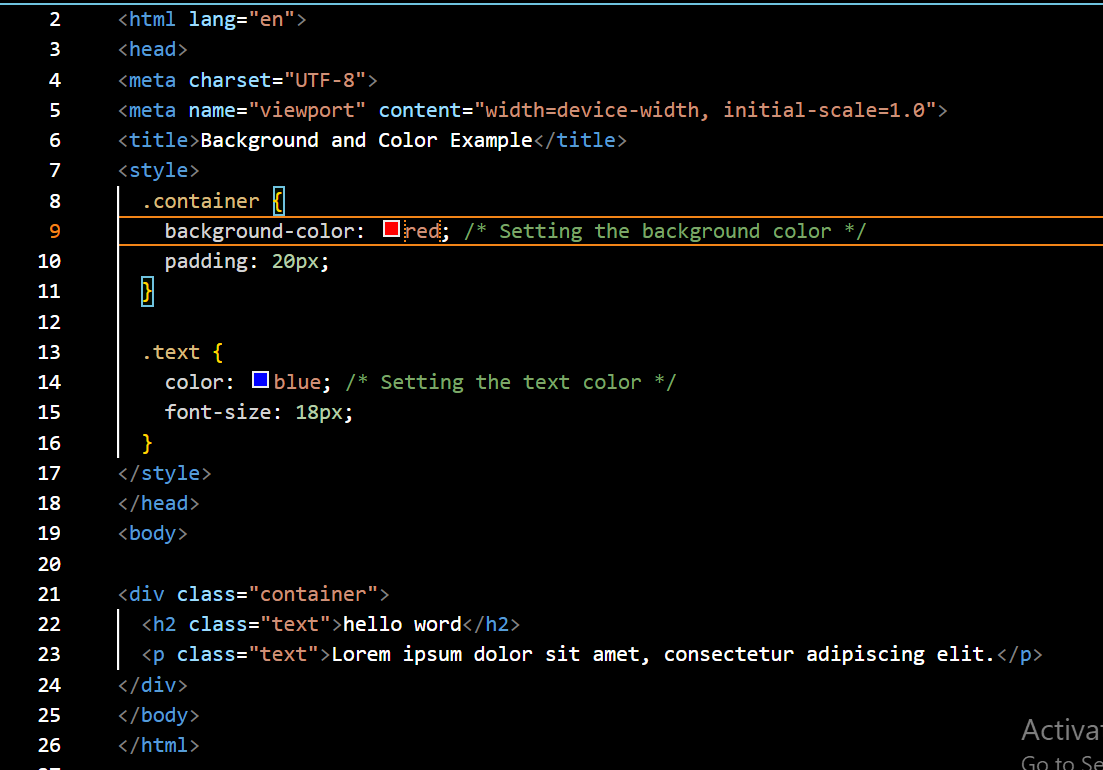
**Q-10. Why should background and color be used as separate properties?**

**Ans –**

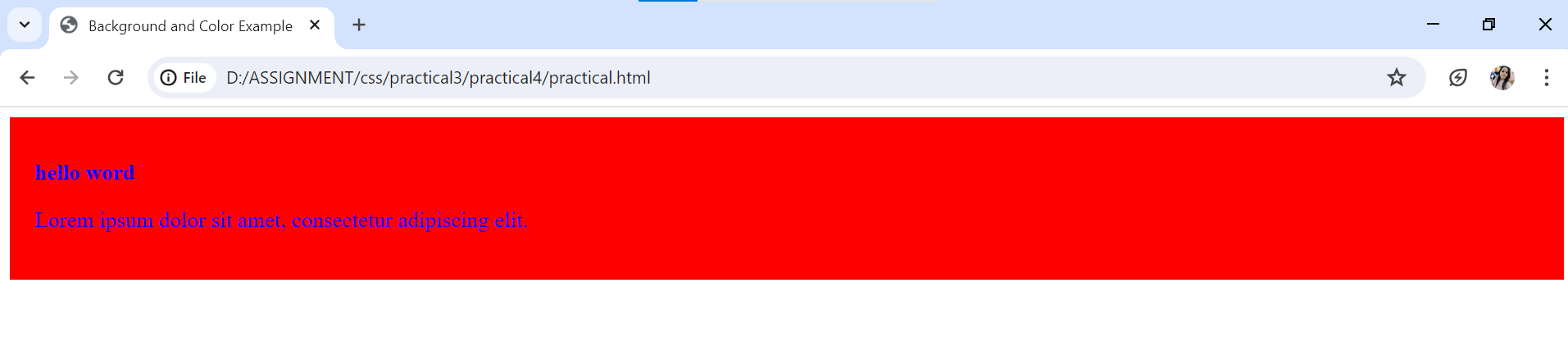
**Background properties -** The background property is used to set things like background images, color, and other effect behind an element.

**Color Property:** The color property, on the other hand, is used to set the color of the text or foreground content inside an element. It determines the color of the text you see on the webpage.

**Code :**



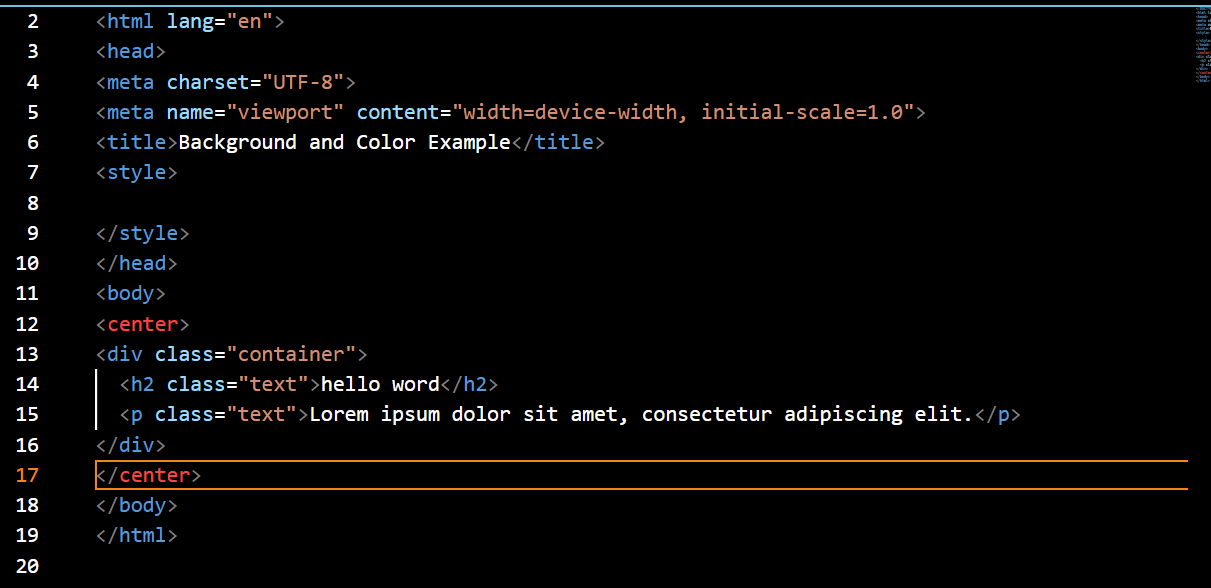
**Output :**



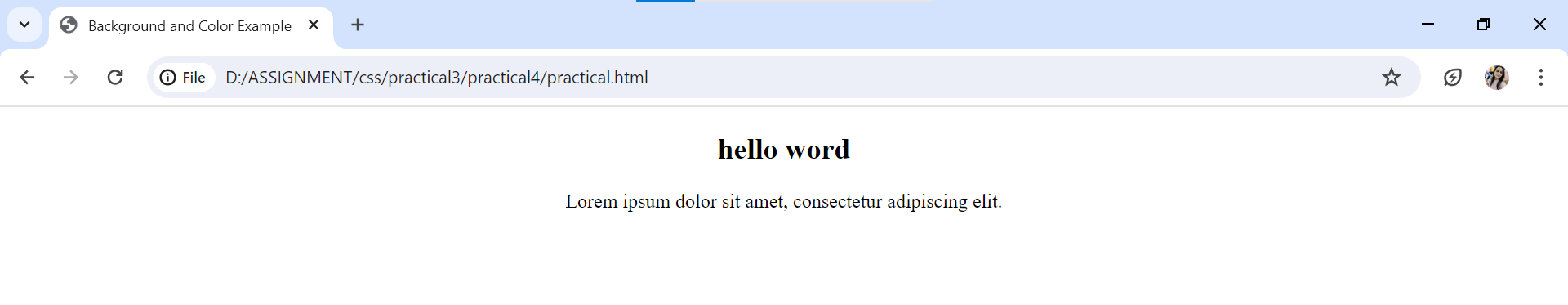
**Q-11. How to center block elements using CSS1?**

**Ans –** center block element in css1 using <center> tag

**Code :**

****

**Output :**

****

**Q-12. How to maintain the CSS specifications?**

**Ans.** The Specification defines how CSS properties should be implemented by browser

The Specification also includes:

1. The syntax and data types of the language

2. Detailed explanation on CSS Selectors

3. How you can assign values to properties

4. The Cascade (the "C" in CSS)

5. How inheritance works

The Box Model e.t.c

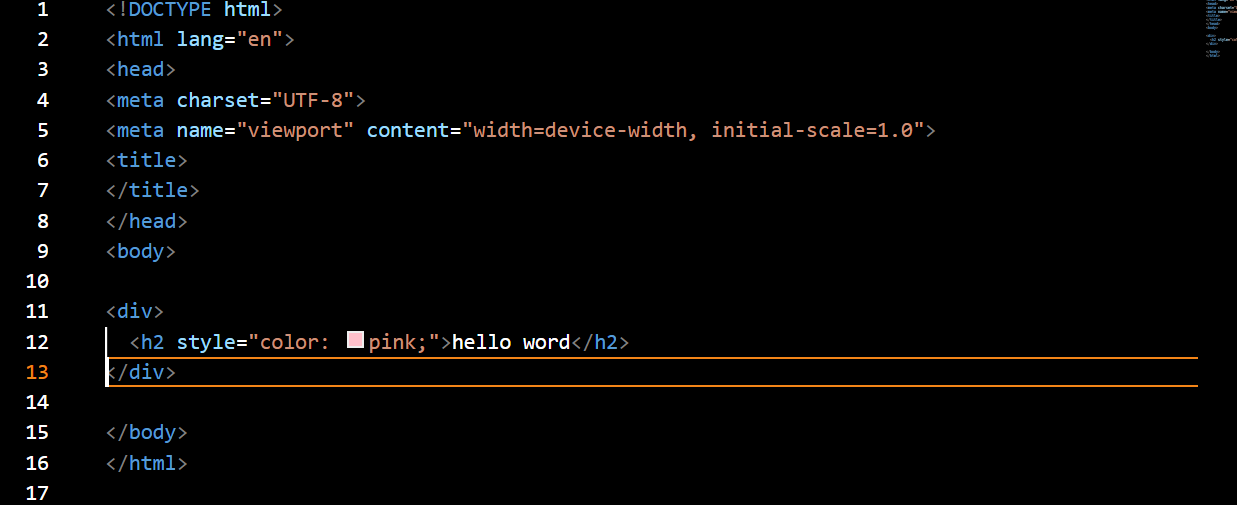
The Specification should be your guide if you need to understand how a specific property or feature works behind the scene and how it works with other CSS properties

**Q-13. What are the ways to integrate CSS as a web page?**

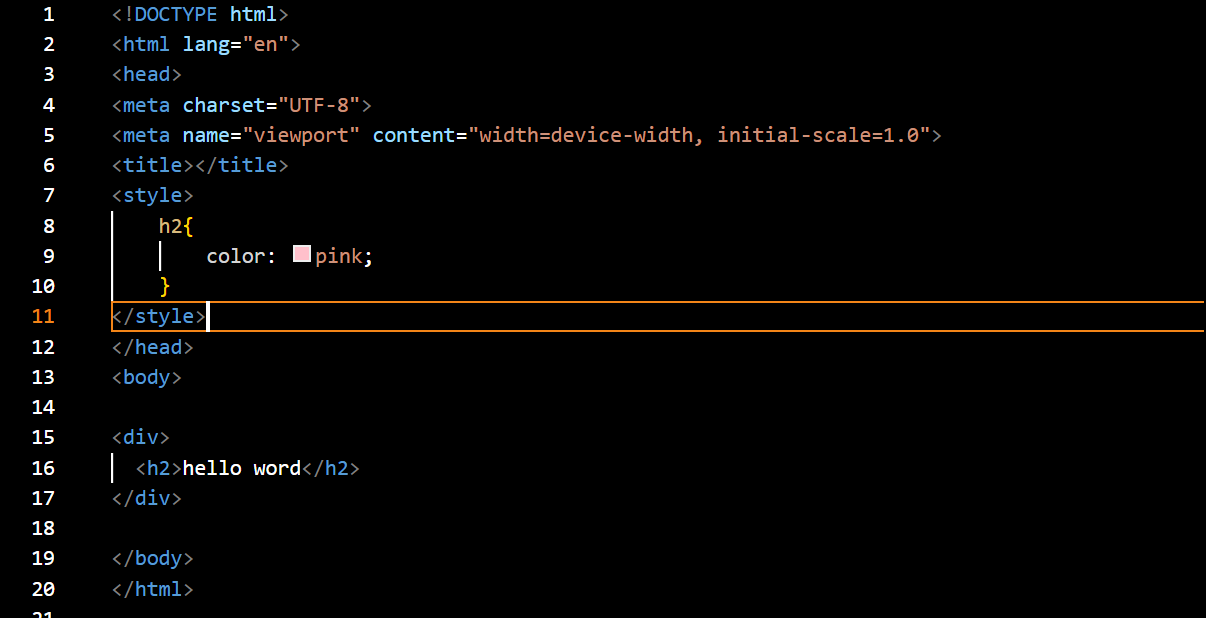
**Ans -** Integrating CSS into a web page can be done in a few different ways, each with its own advantages and use cases. Here are the main methods:

**Inline CSS**: With inline CSS, you write the CSS code directly within the HTML document, using the style attribute. This method is simple and convenient for applying styles to individual elements

**Code :**



**Internal CSS**: Internal CSS involves writing CSS code within a <style> element in the <head> section of the HTML document. This method is useful when you want to apply styles to multiple elements within the same document.



**External CSS**: External CSS involves linking an external CSS file to your HTML document using the <link> element. This method is ideal for maintaining separation of concerns and reusing styles across multiple pages.

**Code :**

****

**Q-14. What is embedded style sheets?**

**Ans.** Embedded style sheets refer to CSS code that is written directly within the HTML document using the <style> element. This method allows you to define styles specific to that HTML document without creating an external CSS file.

**Code :**

****

**Q-15. What are the external style sheets?**

**Ans.** External style sheets are CSS files that contain style rules and are linked to HTML documents using the <link> element

**Code :**



**Q-16. What are the advantages and disadvantages of using external style sheets?**

**Ans.** There are a few advantages and disadvantage of external style sheet

**Advantages:**

1. **Centralized Control**: You can keep all your styling in one place, making it easier to manage and update across multiple pages.

2**. Consistency**: Your website will have a consistent look and feel because all pages reference the same style sheet.

3**. Faster Loading**: Once a user downloads the style sheet, it's cached, so subsequent pages load faster.

4. **Easier Maintenance**: Updating styles is simple; you just need to modify the external style sheet, and changes apply to all linked pages.

**Disadvantages:**

1**. Dependency**: If the style sheet isn't available (due to server issues, for example), your website's styling will be affected.

2**. Extra HTTP Request**: Each external style sheet adds another request to the server, which can slightly slow down page loading.

3. **Cascading Issues**: Conflicts between styles or specificity problems can lead to unexpected results and require troubleshooting.

4**. Limited Control**: External style sheets apply global styles, limiting your ability to customize individual elements on specific pages without additional coding.

**Q - 17. What is the meaning of the CSS selector?**

**Ans.** In CSS, a selector is a pattern used to select and target specific HTML elements to apply styles to. Selectors are a fundamental part of CSS and allow you to precisely define which elements you want to style.

Selectors can target HTML elements based on various criteria, including element type, class, ID, attributes, and relationships between elements. Here are some common types of CSS selectors:

**Element Selector**: Selects all instances of a specific HTML element type.

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>CSS Selectors</title>

    <style>

        /\* Selects all <p> elements \*/

p {

    color: blue;

}

    </style>

</head>

<body>

    <header id="header" class="highlight">

        <h1>Welcome</h1>

    </header>

    <section>

        <p>This is a paragraph.</p>

        <div>

            <span>This is a span element.</span>

        </div>

        <ul>

            <li>Item 1</li>

            <li>Item 2</li>

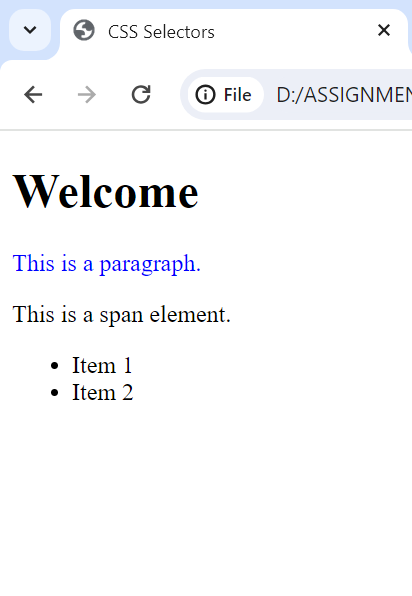
        </ul>

    </section>

</body>

</html>

**Output :**



**Class Selector**: Selects elements with a specific class attribute.

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>CSS Selectors</title>

    <style>

/\* Selects all elements with class "highlight" \*/

.highlight {

    background-color: yellow;

}

    </style>

</head>

<body>

    <header id="header" class="highlight">

        <h1>Welcome</h1>

    </header>

    <section>

        <p>This is a paragraph.</p>

        <div>

            <span>This is a span element.</span>

        </div>

        <ul>

            <li>Item 1</li>

            <li>Item 2</li>

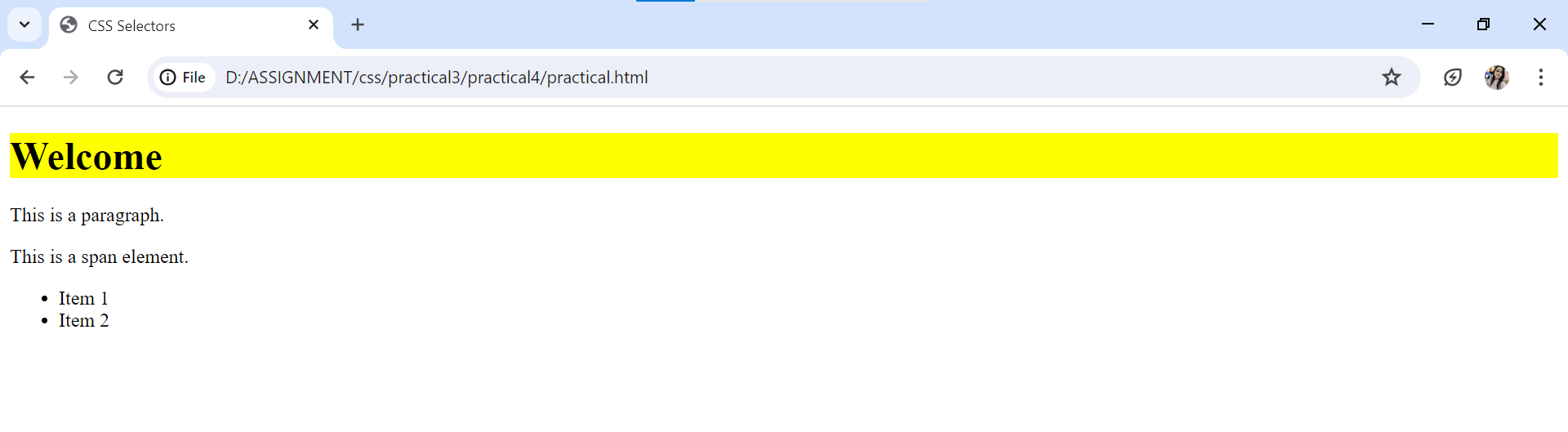
        </ul>

    </section>

</body>

</html>

**Output:**



**ID Selector**: Selects a single element with a specific ID attribute.

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>CSS Selectors</title>

    <style>

/\* Selects the element with id "header" \*/

#header {

    font-size: 24px;

}

    </style>

</head>

<body>

    <header id="header" class="highlight">

        <h1>Welcome</h1>

    </header>

    <section>

        <p>This is a paragraph.</p>

        <div>

            <span>This is a span element.</span>

        </div>

        <ul>

            <li>Item 1</li>

            <li>Item 2</li>

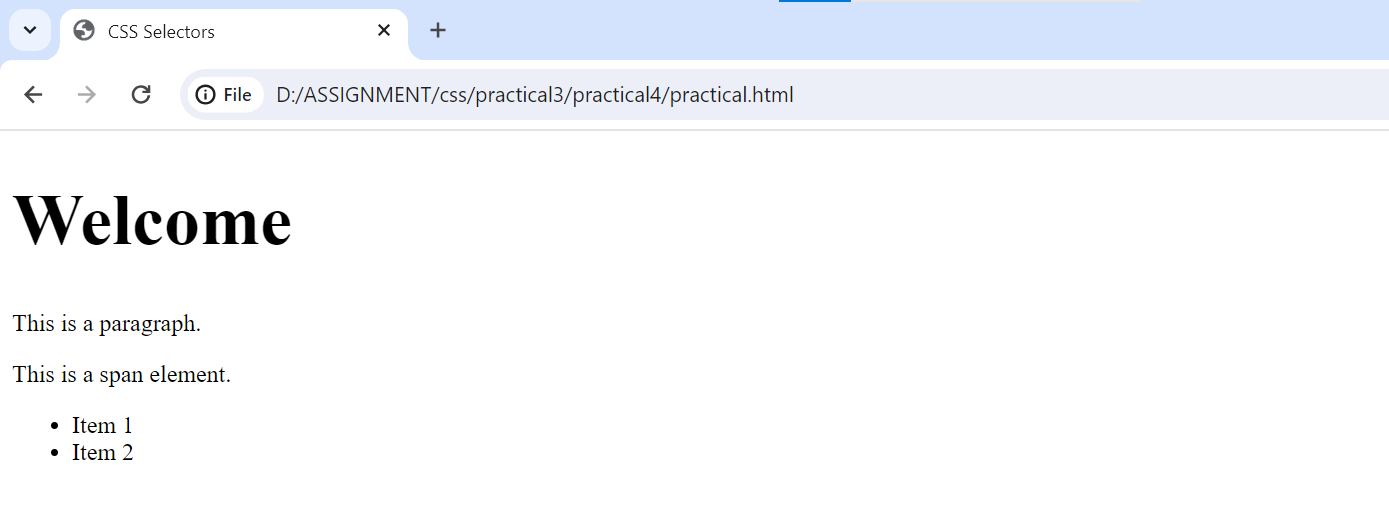
        </ul>

    </section>

</body>

</html>

**Output :**



**Attribute Selector**: Selects elements based on their attributes.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>CSS Selectors</title>

    <style>

/\* Selects all elements with a "target" attribute \*/

[target] {

    color: red;

}

 </style>

</head>

<body>

    <header id="header" class="highlight">

        <h1>Welcome</h1>

    </header>

    <section>

        <p>This is a paragraph.</p>

        <div>

            <span>This is a span element.</span>

        </div>

        <ul>

            <li>Item 1</li>

            <li>Item 2</li>

        </ul>

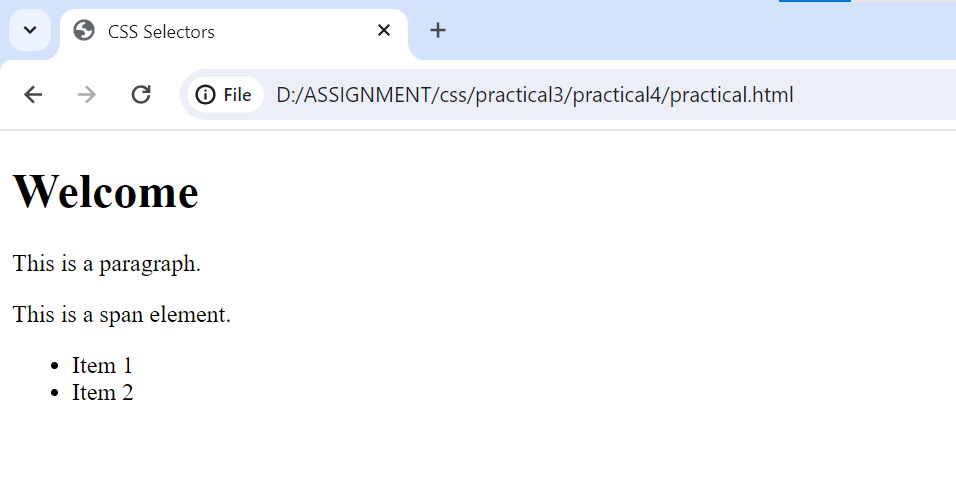
    </section>

</body>

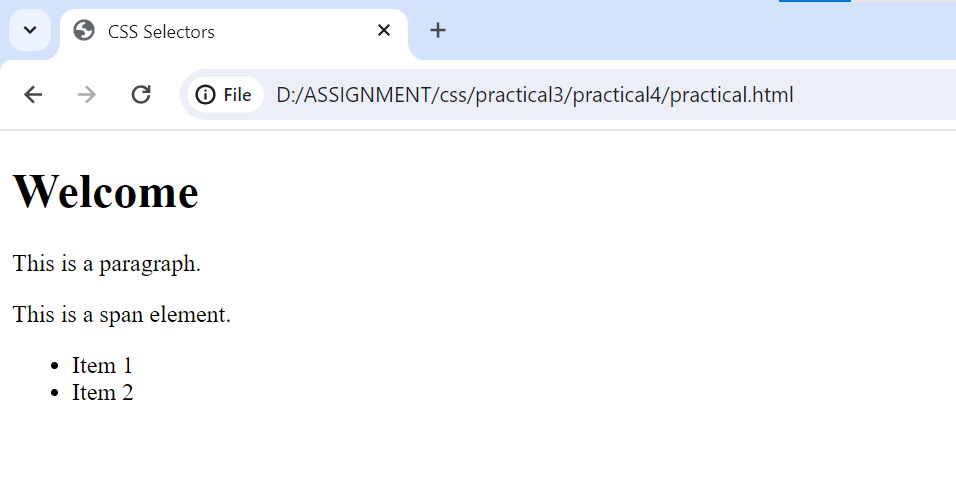
</html>

**Output :**

**Output :**



**Output :**



**Descendant Selector**: Selects an element that is a descendant of another element.

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>CSS Selectors</title>

    <style>

/\* Selects all <span> elements within a <div> \*/

    div span {

    font-weight: bold;

}

 </style>

</head>

<body>

    <header id="header" class="highlight">

        <h1>Welcome</h1>

    </header>

    <section>

        <p>This is a paragraph.</p>

        <div>

            <span>This is a span element.</span>

        </div>

        <ul>

            <li>Item 1</li>

            <li>Item 2</li>

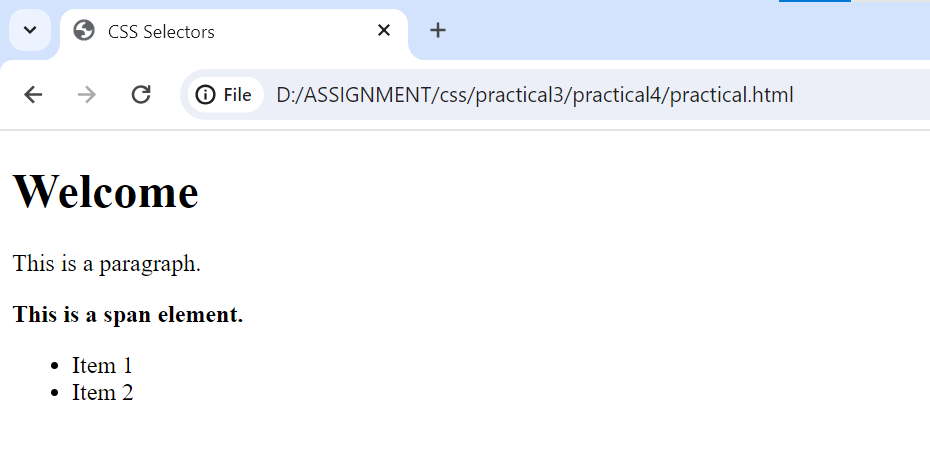
        </ul>

    </section>

</body>

</html>

**Output :**



**Child Selector**: Selects an element that is a direct child of another element.

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>CSS Selectors</title>

    <style>

/\* Selects all <li> elements that are direct children of <ul> \*/

    ul > li {

    list-style-type: circle;

}

 </style>

</head>

<body>

    <header id="header" class="highlight">

        <h1>Welcome</h1>

    </header>

    <section>

        <p>This is a paragraph.</p>

        <div>

            <span>This is a span element.</span>

        </div>

        <ul>

            <li>Item 1</li>

            <li>Item 2</li>

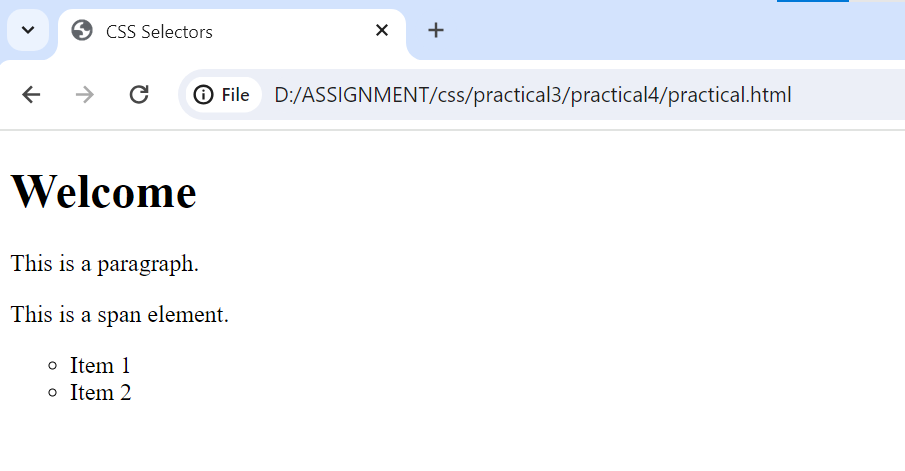
        </ul>

    </section>

</body>

</html>

**Output :**



**Pseudo-class Selector**: Selects elements based on their state or position within the document.

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>CSS Selectors</title>

    <style>

/\* Selects the first <p> element within a <div> \*/

    div p:first-child {

    font-weight: bold;

}

 </style>

</head>

<body>

    <header id="header" class="highlight">

        <h1>Welcome</h1>

    </header>

    <section>

        <p>This is a paragraph.</p>

        <div>

            <span>This is a span element.</span>

        </div>

        <ul>

            <li>Item 1</li>

            <li>Item 2</li>

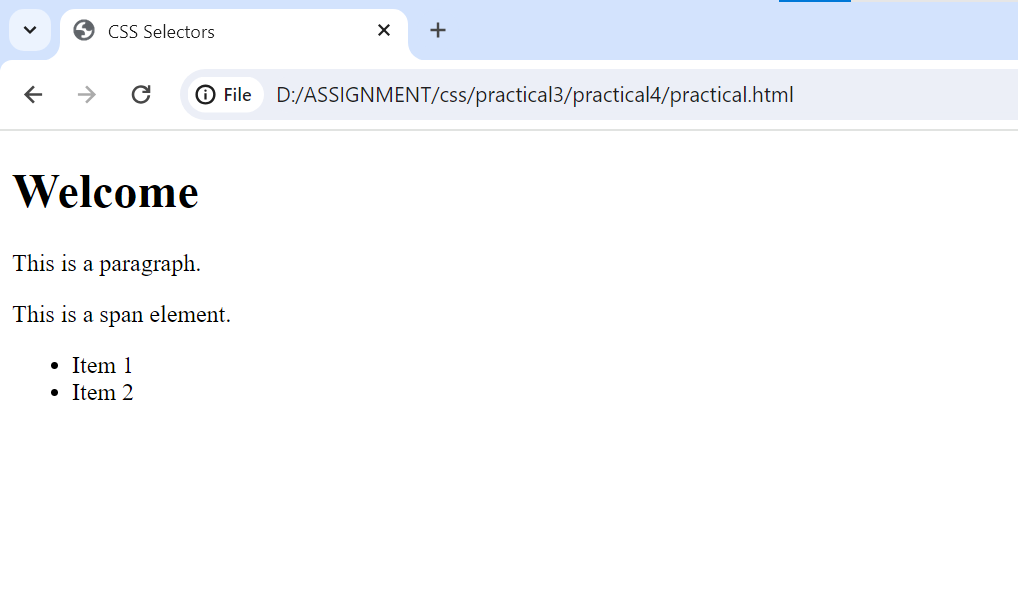
        </ul>

    </section>

</body>

</html>

**Output :**

****

**Q-18. What are the media types allowed by CSS?**

Ans- In CSS, media types define different ways that styles can be applied based on where or how the content is being viewed. Here are the main media types:

**Screen**: Styles for devices with screens, like computers, tablets, or smartphones.

**Print**: Styles for when the content is printed on paper.

**Speech**: Styles for when the content is read aloud by a screen reader.

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Media Types Example</title>

   <style>

    /\* Styles for screens \*/

@media screen {

  body {

    font-family: Arial, sans-serif;

    background-color: lightblue;

  }

  h1 {

    color: blue;

  }

}

Styles for print

@media print {

  body {

    font-family: Times New Roman, serif;

    background-color: white;

  }

  h1 {

    color: black;

  }

}

/\* Styles for speech \*/

@media speech {

  body {

   font-family: sans-serif;

    background-color: white;

    color: black; /\* Set text color for speech \*/

  }

}

</style>

</head>

<body>

    <h1>Welcome to My Website</h1>

    <p>hello word</p>

</body>

</html>

**Output :**



**Q-19. What is the rule set?**

**Ans.** In CSS, a rule set is a combination of a selector and one or more declarations, which together define how HTML elements should be styled.

Selector: This is like the ingredient you want to style, such as a heading (<h1>), a paragraph (<p>), or a class like .container. It tells the browser which HTML elements the styles should be applied to.

Declaration Block: This is like the list of instructions for how you want to style the selected element. It contains one or more declarations, where each declaration includes a property (like color, font-size, or background-color) and its corresponding value (like red, 16px, or #ffffff).

**Code :**



**Output :**

