**Assignment**

**Module (CSS and CSS 3) -2**

**KETAN PILLAI**

**1) What are the benefits of using CSS?**

Ans:- Cascading Style Sheets (CSS) offer numerous benefits for web development by providing a way to control the presentation and layout of a website. Here are some key benefits of using CSS:

1. **Separation of Content and Presentation**
2. **Consistency**
3. **Easy Maintenance**
4. **Responsive Design:**
5. **Accessibility**
6. **Animations and Transitions**

**2) What are the disadvantages of CSS?**

Ans:- Cascading Style Sheets (CSS) have proven to be a crucial technology for web development, but like any tool, they come with their own set of challenges and limitations. Here are some disadvantages of CSS:

1. Browser Compatibility:
2. Limited Layout Control:
3. Responsive Design Complexity:
4. Performance Impact:
5. Debugging Challenges:
6. Print Styling Limitations:

**3) What is the difference between CSS2 and CSS3?**

Ans:- CSS (Cascading Style Sheets) is a style sheet language used for describing the look and formatting of a document written in HTML or XML. CSS has evolved through different versions, and two significant milestones are CSS2 and CSS3. Here are the key differences between CSS2 and CSS3:

1. **Modularity:**

CSS2: CSS2 was a monolithic specification, meaning it was delivered as a single, comprehensive document. It introduced many features like positioning, floating, and media types.

CSS3: CSS3 is modularized, with each module covering a specific aspect of styling. This allows for more flexibility, and new modules can be added or updated independently without affecting the entire specification.

1. **Selectors:**

CSS2: CSS2 introduced a set of selectors for targeting elements on a webpage. This included basic element selectors, class selectors, ID selectors, etc.

CSS3: CSS3 expanded the selector capabilities significantly. It introduced more advanced selectors like attribute selectors, sibling selectors, and pseudo-classes.

1. **Animations and Transitions:**

CSS2: CSS2 did not have native support for animations and transitions.

CSS3: CSS3 introduced the @keyframes rule, which allows developers to create animations, and the transition property for smooth transitions between states.

1. **Flexbox and Grid Layout:**

CSS2: CSS2 had limited support for layout, primarily relying on the box model and float-based layouts.

CSS3: CSS3 introduced Flexbox and Grid Layout, providing powerful and flexible tools for creating complex and responsive layouts.

1. **Round Corners and Shadows:**

CSS2: Achieving rounded corners and box shadows required the use of background images or other workarounds.

CSS3: CSS3 introduced properties like border-radius for rounded corners and box-shadow for adding shadows, making these styling elements much easier to implement.

**4) Name a few CSS style components.**

Ans:- Here are a few CSS style components or properties:

**1. Color:**

- Properties like `color`, `background-color`, and `border-color` are used to define the color of text, background, and borders, respectively.

**2. Typography:**

- Properties such as `font-family`, `font-size`, `font-weight`, and `line-height` are used to control the typography and text formatting.

**3. Box Model:**

- The box model is a fundamental concept in CSS. Properties like `margin`, `padding`, `border`, and `box-sizing` determine the spacing and dimensions of elements.

**4. Layout:**

- CSS provides various properties for layout, including `display`, `position`, `float`, `flexbox`, and `grid`, which determine how elements are positioned and flow within the document.

**5. Flexbox:**

- The `flexbox` layout model is designed for building flexible and responsive layouts. Properties like `flex-direction`, `justify-content`, and `align-items` are used to control the layout of flex containers and their items.

**6. Grid Layout:**

- The `grid` layout is a powerful system for creating two-dimensional layouts. Properties like `grid-template-columns`, `grid-template-rows`, and `grid-gap` are used to define the grid structure.

**7. Transitions and Animations:**

- CSS properties like `transition` and `animation` enable the creation of smooth transitions and animations for elements on the page.

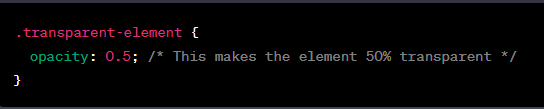
**8. Responsive Design: -**

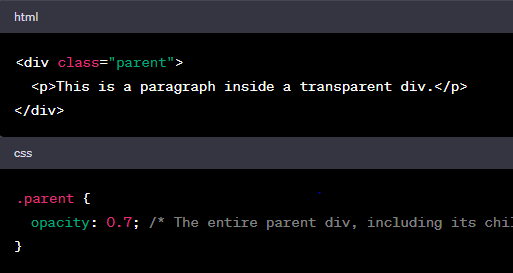
Media queries (`@media`) and related properties allow developers to create responsive designs, adapting layouts and styles based on the characteristics of the device or viewport.

**5) What do you understand by CSS opacity?**

Ans: - CSS opacity is a property that controls the transparency of an element. It is used to specify the degree to which an element is opaque or transparent. The opacity property takes values in the range from 0 to 1, where 0 represents completely transparent (invisible), and 1 represents completely opaque (fully visible).

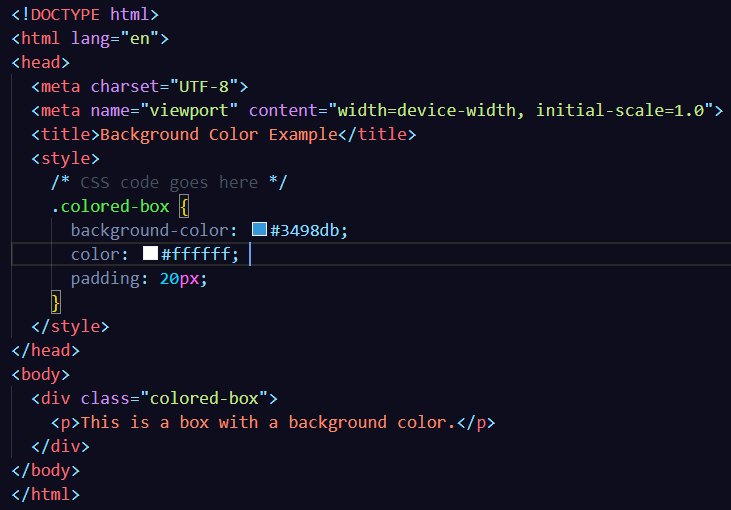
Eg:





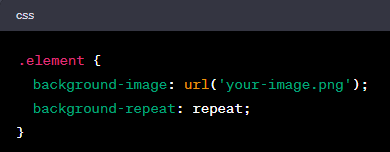
**6) How can the background color of an element be changed?**

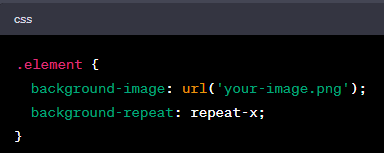
Ans:- The background color of an element in HTML can be changed using CSS by applying the background-color property. Here's a simple example:

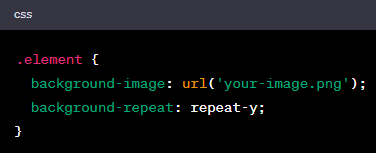


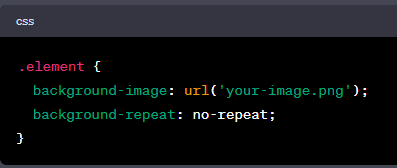
1. **How can image repetition of the backup be controlled?**

Ans:- In CSS, you can control how a background image repeats using the background-repeat property. The property can take the following values:







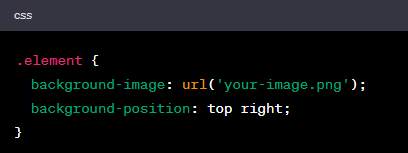


1. **What is the use of the background-position property?**

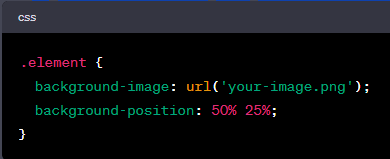
Ans:- The background-position property in CSS is used to specify the starting position of a background image within its containing element. It determines where the top-left corner of the background image should be placed relative to the top-left corner of the element.

Examples:-

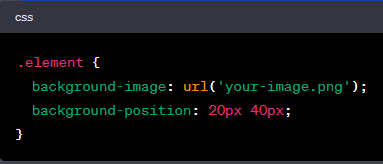
1. **Keyword Values:** top, bottom, left, right, center: These keywords can be used individually or combined to position the background image.



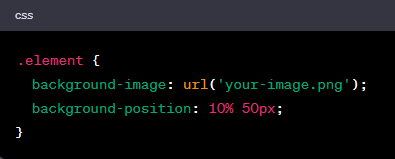
1. **Percentage Values:** Percentages represent the distance as a percentage of the containing element's size. 0% is the left or top, and 100% is the right or bottom.



1. **Length Values**: You can use length units (like pixels or em) to specify an exact distance.



1. **Combination:** Combining different units for horizontal and vertical positioning.

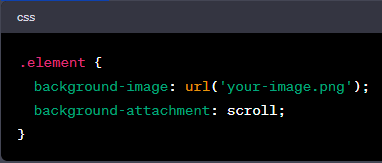


1. **Which property controls the image scroll in the background?**

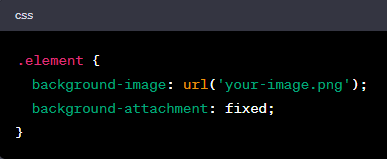
Ans:- The property that controls the scrolling behaviour of a background image in CSS is the background-attachment property. This property specifies whether the background image should scroll with the content or remain fixed as the user scrolls down the page.

The background-attachment property can take the following values:

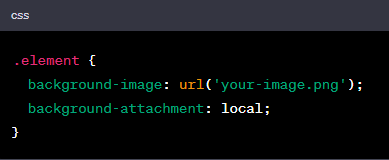
1. **scroll (default):** The background image scrolls along with the content as the user scrolls down the page.



1. **fixed:** The background image remains fixed in the viewport, and it does not move as the user scrolls.



1. **local:** The background image scrolls with the element's contents. This is similar to scroll

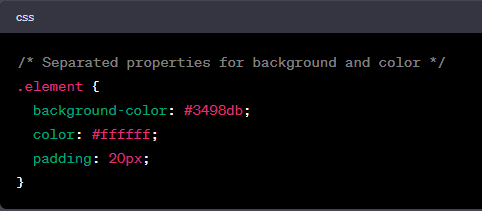


1. **Why should background and color be used as separate properties?**

Ans:- Using separate properties for background and color in CSS allows for better flexibility, maintainability, and clarity in your code. Here are some reasons why it's beneficial to use distinct properties for background and text color:

1. **Separation of Concerns:**
2. **Readability and Maintainability:**
3. **Flexibility in Design:**
4. **Responsive Design:**
5. **Modularity**

Eg:-

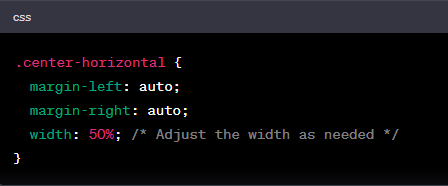


1. **How to center block elements using CSS1?**

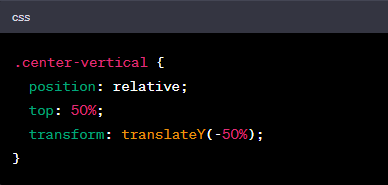
Ans:- Centering block-level elements in CSS1 typically involved using a combination of properties like text-align and setting margins.

Here are a couple of techniques you could use to center block elements in CSS1:

1. **Horizontally Centering:**



1. **Vertically Centering:**



1. **How to maintain the CSS specifications?**

Ans:- Maintaining CSS specifications involves several best practices and strategies to ensure consistency, scalability, and ease of maintenance. Here are some tips:

1. **Documentation**:

- Document your CSS code thoroughly. Use comments to explain complex styles, the purpose of specific rules, and any dependencies.

**2. Modularization:**

- Break down your CSS into modular components. Each component should be responsible for styling a specific part of your application or webpage.

3. **Avoid !important:**

- Minimize the use of `!important`. This ensures that styles are applied in the natural cascading order, making it easier to predict and troubleshoot.

4. **Version Control:**

- Use version control systems (e.g., Git) to track changes to your CSS files. This allows you to roll back changes, collaborate with others, and maintain a history of your styles.

5. **Testing:**

Regularly test your styles across different browsers to catch any inconsistencies or issues.

6. **Responsive Design:**

- Design and test for responsiveness. Use media queries and other responsive design techniques to ensure your styles adapt well to various screen sizes and devices.

7. **Performance:**

- Optimize your CSS for performance. Minify and concatenate your stylesheets for production.

1. **What are the ways to integrate CSS as a web page?**

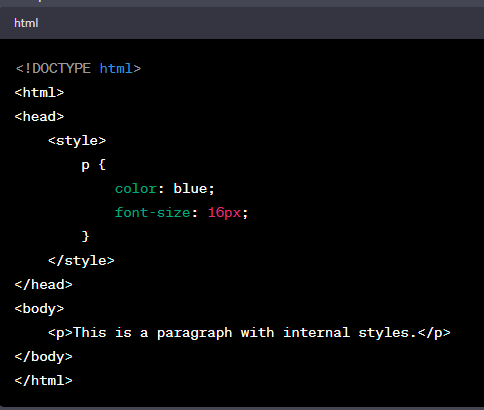
Ans:- There are several ways to integrate CSS into a web page. The method you choose depends on the specific requirements of your project and your personal or team preferences. Here are some common ways to integrate CSS:

1. **Inline Styles:**

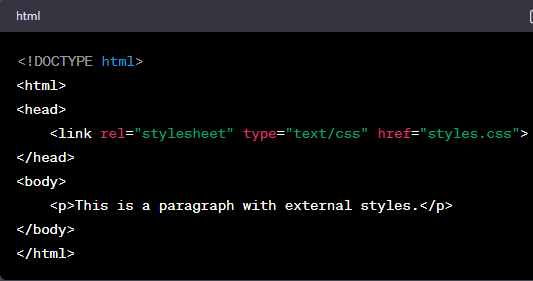


1. **Internal Styles:**

Internal styles involve placing the CSS code within the <style> tag in the head of the HTML document.



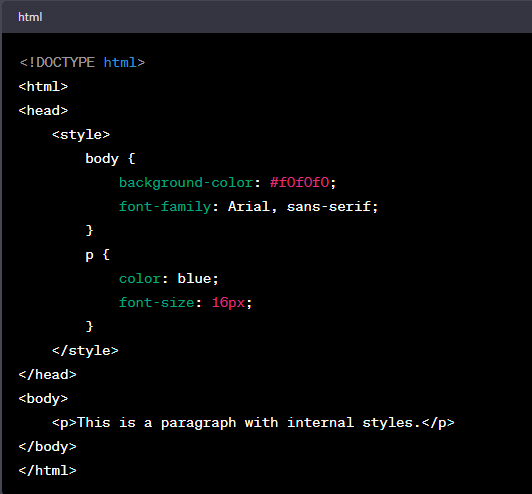
1. **External Styles:** External styles are stored in a separate CSS file, which is linked to the HTML document. This is a common and recommended practice for larger projects.



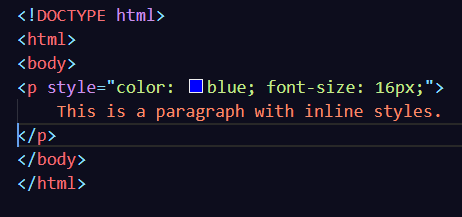
1. **What is embedded style sheets?**

Ans:- Embedded styles, also known as internal or inline styles, refer to the practice of including CSS rules directly within the HTML document. There are two main types of embedded styles:

1. **Internal Styles (Embedded in the <style> tag):**



1. **Inline Styles (Applied directly to HTML elements):**



1. **What are the external style sheets?**

Ans:- External style sheets are a way to separate the styling of a website from its HTML content. Instead of embedding styles directly within the HTML document, you can create a separate file with the extension ".

Here's how you typically link an external style sheet to an HTML document:

1. **Create a CSS File:**

Create a separate file with a ".css" extension (e.g., styles.css). In this file, you define the styles for various HTML elements.



1. **Link the CSS File to HTML:**



1. **What are the advantages and disadvantages of using external style sheets?**

Ans:- Using external style sheets in web development comes with several advantages and some potential disadvantages. Here's a breakdown of the pros and cons:

**Advantages:**

1. **Consistency:** External style sheets promote consistency in design. By centralizing your styles in a single CSS file, you ensure that the same styles are applied consistently across multiple pages of your website.

2. **Maintenance:** Changes to the styling can be made in one central location. This makes maintenance easier, as you only need to update the external style sheet to see the changes reflected throughout the entire site.

3. **Ease of Update:** If you need to update the styling for a particular element or aspect of your site, you can do so in the external style sheet without having to modify each individual HTML file.

4. **Bandwidth Efficiency:**

External style sheets can be cached by the browser, potentially reducing the amount of data that needs to be downloaded when a user visits multiple pages of your site.

5. **Reuse of Styles:** You can reuse the same external style sheet across multiple projects, promoting a consistent look and feel if you are working on multiple websites.

**Disadvantages:**

1. **Dependency on External File:** If the external style sheet fails to load or is missing, the styling for your entire website may not be applied correctly.

2. **Extra HTTP Request:** Loading an external style sheet requires an additional HTTP request, which may slightly impact the initial loading time of a webpage.

3. **Rendering Blocking:** External style sheets can be render-blocking, meaning that the browser may need to wait for the external CSS file to be downloaded and processed before rendering the content of the page. This can potentially slow down the perceived load time.

4. **Limited Offline Functionality:** If a user is offline or the external style sheet fails to load, the website might not display correctly, especially if it relies heavily on external styling.

5. **Specificity Challenges:** Achieving specificity (prioritizing which styles should be applied) can be more challenging when dealing with multiple external style sheets.

1. **What is the meaning of the CSS selector?**

Ans:- Cascading Style Sheets (CSS), a selector is a pattern that is used to select and style one or more HTML elements.

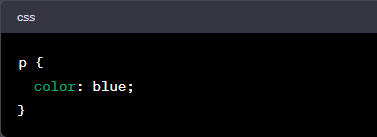
Selector: Specifies which HTML elements the styles should apply to.

Property: The aspect of the element you want to style (e.g., color, font-size).

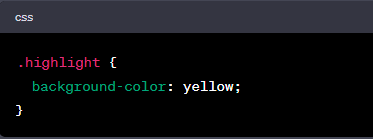
Value: The specific value you want to apply to the property.

Here are some common types of CSS selectors:

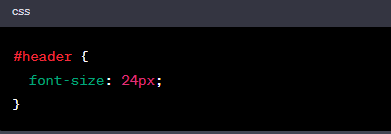
1. **Element Selector:**



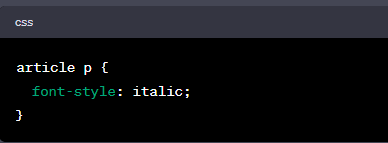
1. **Class Selector:**



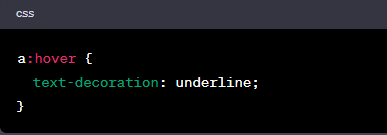
1. **ID Selector:**



1. **Descendant Selector:**



1. **Pseudo-class Selector:**

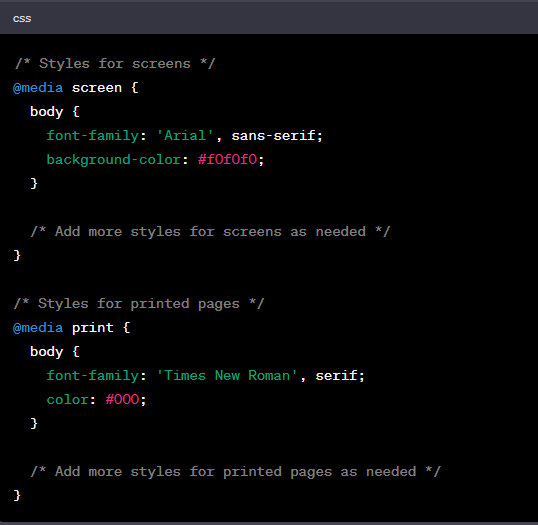


1. **What are the media types allowed by CSS?**

Ans:- CSS (Cascading Style Sheets) supports different media types, allowing you to define styles specifically for different types of devices or media. The @media rule is used to apply styles based on the characteristics of the output device. Here are some of the commonly used media types in CSS:

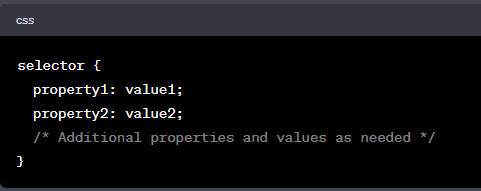
1. **all**
2. **print**
3. **screen**
4. **speech**
5. **braille**
6. **embossed**

Here's an example of using the @media rule with a media type:

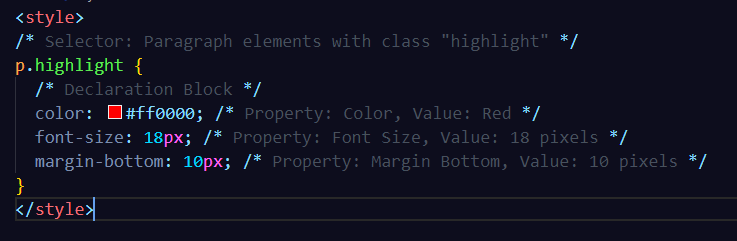


1. **What is the rule set?**

Ans:- A rule set in Cascading Style Sheets (CSS) consists of one or more CSS declarations grouped together. Each rule set is used to define the styling for a specific set of HTML elements that match a given CSS selector. The basic syntax of a CSS rule set is as follows:



Here's an example of a CSS rule set:



CSS rule sets play a central role in defining the visual presentation of HTML documents and are fundamental to the separation of content and style in web development.

1. **Create Layouts:**

Please find attached file:-