Web Designing Assignment Term-1

Module (CSS and CSS 3) -2

1. What are the benefits of using CSS?

Ans.: Advantages of CSS:

- CSS plays an important role, by using CSS you simply got to specify a repeated style for element once & use it multiple times as because CSS will automatically apply the required styles.
- The main advantage of CSS is that style is applied consistently across variety of sites. One instruction can control several areas which is advantageous.
- Web designers needs to use few lines of programming for every page improving site speed.
- Cascading sheet not only simplifies website development, but also simplifies the maintenance as a change of one line of code affects the whole web site and maintenance time.
- It is less complex therefore the effort are significantly reduced.
- It helps to form spontaneous and consistent changes.
- CSS changes are device friendly. With people employing a batch of various range of smart devices to access websites over the web, there's a requirement for responsive web design.
- It has the power for re-positioning. It helps us to determine the changes within the position of web elements who are there on the page.
- These bandwidth savings are substantial figures of insignificant tags that are indistinct from a mess of pages.
- Easy for the user to customize the online page
- It reduces the file transfer size.

2. What are the disadvantages of CSS?

Ans.: Disadvantages of CSS:

- CSS, CSS 1 up to CSS3, result in creating of confusion among web browsers.
- With CSS, what works with one browser might not always work with another. The web developers need to test for compatibility, running the program across multiple browsers.
- There exists a scarcity of security.
- After making the changes we need to confirm the compatibility if they appear. The similar change affects on all the browsers.
- The programming language world is complicated for non-developers and beginners. Different levels of CSS i.e. CSS, CSS 2, CSS 3 are often quite confusing.
- Browser compatibility (some styles sheet are supported and some are not).
- CSS works differently on different browsers. IE and Opera supports CSS as different logic.
- There might be cross-browser issues while using CSS.
- There are multiple levels which creates confusion for non-developers and beginners.

3. What is the difference between CSS2 and CSS3?

Ans.:

CSS 2	**CSS 3**
CSS splits up different sections Of the code into modules.	Both CSS and HTML were put into a single file, there was no concept of modules before.
There are new ways you can write CSS rules with a bunch of CSS selectors	There were no new ways of writing the CSS rules.
There is no backward compatibility with CSS.	There is backward compatibility maintained with CSS3

With CSS2 only web safe fonts can be used	With CSS3 special fonts can be used such as those in Google Fonts & TypeCast
With CSS2 the concept of simple selectors were present	With CSS3 the selectors were called as a sequence of simple selectors.com
Using CSS2 for rounded borders, coding the css style were complex	There is provision for automatically assigning rounded borders to objects
Splitting text into multiple columns required complex coding because the standard was not equipped enough to break the text into columns so that it would fit into a box	It has the capability to split text into various columns so that each text block appears as a layout of the newspaper.
CSS2 doesn't support the Border-Box property	CSS3 supports the Border-Box property

4. Name a few CSS style components.

Ans. list of some common CSS style components:

1. Typography:
    - Font family
    - Font size
    - Font weight
    - Line height
    - Text alignment
    - Text decoration
    - Text transform (uppercase, lowercase, capitalize)
2. Color:
    - Text color
    - Background color
    - Border color

- Gradient backgrounds
- Opacity

3. Layout:
    - Width and height
    - Margin
    - Padding
    - Display property (inline, block, inline-block, flex, grid)
    - Positioning (relative, absolute, fixed, sticky)
4. Box Model:
    - Border width
    - Border style
    - Border radius
    - Box shadow
    - Outline
5. Flexbox:
    - Flex direction
    - Flex wrap
    - Justify content
    - Align items
    - Align self
    - Flex grow, shrink, basis
6. Grid:
    - Grid template columns
    - Grid template rows
    - Grid gap
    - Grid column start and end
    - Grid row start and end
    - Grid area
7. Positioning:
    - Relative positioning
    - Absolute positioning
    - Fixed positioning
    - Sticky positioning

8. Animations and Transitions:
    - Transition property
    - Transition duration
    - Transition timing function
    - Animation property
    - Key frames
9. Responsive Design:
    - Media queries
    - Viewport units (vw, vh, vmin, vmax)
    - Responsive images (srcset, sizes)
10. Other Components:
    - Lists (list-style-type, list-style-image)
    - Tables
    - Forms
    - Buttons
    - Icons
    - Images
    - Backgrounds

5. What do you understand by CSS opacity?

Ans. CSS opacity is a property that controls the transparency of an element, allowing you to make it partially transparent. It affects the entire element, including its content, background, border, and any child elements.

The opacity property accepts values between 0 and 1, where:

0: The element is completely transparent, meaning it is invisible.

1: The element is fully opaque, with no transparency.

Values between 0 and 1 represent varying degrees of transparency. For example:

0.5: The element is 50% transparent.

0.75: The element is 75% transparent, meaning it's more opaque than if it were 50% transparent.

6. How can the background color of an element be changed?
Ans. The background color of an element in CSS using the background-color property. Here's how you can do it:
CSS:-

```
.element {
background-color: #ff0000; /* Set the background color to red */
}
```

In this example:
.element is the class name or selector for the element whose background color you want to change.

background-color is the CSS property used to specify the background color.

#ff0000 is the color value. In this case, it's a hexadecimal color value representing red. You can also use other color formats such as RGB, RGBA, HSL, HSLA, or color names.

7. How can image repetition of the backup be controlled?
Ans. Background images repetition control using the background-repeat property. This property allows you to specify whether and how a background image should repeat both horizontally and vertically. The background-repeat property can take the following values:
repeat: The background image will repeat both horizontally and vertically. This is the default behavior.
repeat-x: The background image will repeat only horizontally.
repeat-y: The background image will repeat only vertically.
no-repeat: The background image will not repeat, and it will be shown only once.
Here's example of background-repeat property in CSS:

```
.element {
   background-image: url('image.jpg'); /* Set the background image */
```

```
    background-repeat: repeat; /* Image will repeat both horizontally and vertically */
}
```

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

Ans. The background-position property in CSS is mainly used to sets the initial position for the background image ie., it is used to set an image at a certain position. The position that is relative to the positioning layer, can be set by using the background-origin property.

Syntax:

background-position: value;

Property values:

- background-position: left top: This property is used to set the image at the left top.
- background-position: left center :This property is used to set the image at the left center.
- background-position: left bottom; This property is used to set the image at the left bottom.
- background-position: center top; This property is used to set the image at the center top position.
- background-position: center center; This property is used to set the image at the center center position.
- background-position: center bottom; This property is used to set the image at the center bottom position.
- background-position: right top; This property is used to set the image at the right top position.
- background-position: right center; This property is used to set the image at the right center position.

- background-position: right bottom; This property is used to set the image at the right bottom position.
- background-position: 25% 75%: This property is used to set the image at 25% from the left and 75% from the top.
- background-position: 30px 80px;: This property is used to set the image at the 30px from left and 80px from top.

9. Which property controls the image scroll in the background ?

Ans. The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page).

Specify that the background image should scroll with the rest of the page:

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: scroll;
}
```

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

Ans. The separation of the background and color properties in CSS allows for greater flexibility and specificity when styling elements. By having separate properties, developers can easily target and modify the background and text color independently. This separation also allows for more efficient use of CSS, as it enables the inheritance and overriding of specific styles for different elements. While these properties are often set together for readability and aesthetics, separating them provides more control over the styling of web elements.

11. How to center block elements using CSS1?

Ans. There are two steps to center a block-level element –

Step 1: **Define the external width** – We need to define the external width. Block-level elements have the default width of 100% of the webpage, so for centering the block element, we need space around it. So for generating the space, we are giving it a width.

Step 2: **Set the left-margin and the right-margin of the element to auto** – Since we produced a remaining space by providing external width so now we need to align that space properly that's why we should use margin property. Margin is a property that tells how to align a remaining space. So for centering the element you must set left-margin to auto and right-margin to auto.

12. How to maintain the CSS specification?

Ans. Maintaining CSS specifications involves several key practices to ensure consistency, clarity, and compatibility across different platforms and devices. Here's a guideline:

1. Documentation: Thoroughly document each CSS property, its accepted values, browser compatibility, and any special considerations. Keep the documentation updated with each specification revision.
2. Version Control: Use a version control system like Git to track changes to the specification over time. This allows for easy collaboration and rollbacks if necessary.
3. Feedback Mechanism: Establish a feedback mechanism, such as mailing lists, forums, or GitHub issues, where developers can provide feedback, report bugs, and suggest improvements.
4. Regular Updates: Regularly update the specification to address new features, bug fixes, and changes in web technology. This ensures

that the specification remains relevant and up-to-date with industry trends.

5. Cross-browser Testing: Test CSS properties across different browsers and devices to ensure consistent behavior. Consider using automated testing tools and browser compatibility services to streamline this process.
6. Community Engagement: Engage with the web development community through conferences, workshops, and online forums to gather insights, share updates, and foster collaboration.
7. Clear Communication: Clearly communicate any changes or updates to the CSS specification through release notes, blog posts, and other channels. This helps developers stay informed and adapt their code accordingly.
8. Backward Compatibility: Whenever possible, maintain backward compatibility with older versions of the specification to support existing codebases and minimize disruptions for developers.
9. Accessibility Considerations: Ensure that the CSS specification takes into account accessibility best practices, such as proper use of semantic markup, keyboard navigation, and screen reader compatibility.

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

Ans. There are several ways to integrate CSS (Cascading Style Sheets) into a web page:

1. Inline CSS: You can apply CSS directly to HTML elements using the style attribute. For example:
   Html

   ```
   <p style="color: red; font-size: 16px;">This is a paragraph with inline CSS.</p>
   ```

2. Internal CSS: You can embed CSS within the <style> tags in the <head> section of an HTML document. This method is useful when you want to apply styles to multiple elements within a single HTML file.
   html
   ```
   <head>
    <style> p { color: blue; font-size: 14px; }
   </style>
    </head>
   ```

3. External CSS: You can create a separate CSS file and link it to your HTML document using the <link> tag. This method is preferred for larger projects as it keeps the HTML and CSS separate, making the code easier to manage and maintain.
   html
   ```
   <head>
    <link rel="stylesheet" type="text/css" href="styles.css">
   </head>
   ```

4. Import CSS: Similar to external CSS, you can import a CSS file within another CSS file using the @import rule. This method is less common but can be useful for organizing CSS files.
   css
   ```
   /* styles.css */
   @import url("other-styles.css");
   ```
5. CSS Frameworks: You can utilize CSS frameworks like Bootstrap, Foundation, or Bulma, which provide pre-written CSS and often JavaScript components to build responsive and attractive web pages more quickly.

14. What is embedded style sheets?

Ans. Embedded style sheets, also known as internal style sheets, refer to CSS styles that are placed directly within an HTML document's <style> element. This allows you to define styles that apply only to the elements within that specific HTML file.

Here's an example of how embedded style sheets are used:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Embedded Style Sheet Example
</title>
 <style> /* Embedded CSS */
 body  {
              font-family: Arial, sans-serif;
              background-color: #f0f0f0;
       }
 h1 {
            color: blue;
      }
p {
      font-size: 16px;
      line-height: 1.5;
  }
</style>
</head>
<body> <h1>This is a Heading</h1> <p>This is a paragraph with some text.</p>
 </body>
</html>
```

In this example:

- The CSS styles are enclosed within the <style> element in the <head> section of the HTML document.
- The styles defined within the <style> element apply only to the elements within the same HTML document.
- You can define various styles for different HTML elements, such as body, h1, and p, among others.
- These styles will be applied to the corresponding HTML elements when the page is rendered in a web browser.

Embedded style sheets are useful for small-scale projects or when you want to define styles that are specific to a single HTML document. However, for larger projects, external style sheets are often preferred as they promote better code organization and reusability.

15. What are the external style sheets?

Ans. External style sheets are separate CSS files that contain styles that can be applied to one or more HTML documents. These CSS files are linked to HTML documents using the <link> element within the <head> section. External style sheets promote modularity, reusability, and easier maintenance of styles across multiple web pages.

Here's an example of how external style sheets are used:

Create a CSS file (e.g., styles.css) and define your styles within it:

```
/* styles.css */
body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
}
h1 {
  color: blue;
}
```

```
p {
    font-size: 16px;
    line-height: 1.5;
}
```

Link the external CSS file to your HTML document using the <link> element within the <head> section:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>External Style Sheet Example</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph with some text.</p>
</body>
</html>
```

Using external style sheets allows you to maintain consistency across multiple web pages by applying the same styles to different HTML documents. It also makes it easier to update styles across your website since changes made to the external CSS file will automatically reflect on all linked HTML documents.

16. What are the advantages and disadvantages of using external style sheet?

Ans. Using external style sheets in web development offers several advantages and some potential drawbacks:

Advantages:

Modularity and Reusability: External style sheets promote modularity by separating style definitions from HTML content. Styles defined in an external CSS file can be reused across multiple HTML documents, leading to more consistent and maintainable code.

Consistency: External style sheets enable consistent styling across an entire website. By linking multiple HTML documents to the same CSS file, you ensure that all pages share the same visual appearance, which enhances user experience and brand identity.

Ease of Maintenance: Making changes to styles is easier with external style sheets. Instead of modifying styles within individual HTML files, you can update a single CSS file, and the changes will be reflected across all linked HTML documents automatically. This streamlines the maintenance process and reduces the chance of inconsistencies.

Faster Page Loading: External style sheets are cached by web browsers after the initial download, which can improve page loading times for subsequent visits to your website. This caching mechanism reduces the amount of data that needs to be transmitted over the network, leading to faster loading times and better overall performance.

Disadvantages:

Additional HTTP Requests: Each external style sheet linked to an HTML document requires a separate HTTP request, which can slightly increase page loading times, particularly for larger websites with multiple CSS files. However, this drawback is often outweighed by the benefits of modularity and maintainability.

Dependency on External Resources: If the external CSS file fails to load or is inaccessible, the styling of your HTML documents may be compromised. This dependency on external resources introduces a

potential point of failure, especially if the CSS file is hosted on a separate server or CDN (Content Delivery Network).

Potential Specificity Issues: When using external style sheets, it's essential to manage CSS specificity carefully to avoid conflicts and unintended styling overrides. Conflicting styles defined in different CSS files or inline styles can lead to unexpected behavior, requiring careful organization and planning of your style sheets.

17. What is the meaning of the CSS selector?

Ans. A CSS selector is a pattern used to select and style elements in an HTML document. It allows you to target specific elements or groups of elements based on their attributes, such as their tag name, class, ID, or relationship to other elements in the document. CSS selectors are given below:

1. Type Selector:

Targets elements based on their tag name.

```
p {
/* Styles for all <p> elements */
}
```

2. Class Selector:

Targets elements with a specific class attribute.

```
.example {
 /* Styles for elements with class="example" */
 }
```

3. ID Selector:

Targets a single element based on its ID attribute.

```
#unique {
/* Styles for element with id="unique" */
}
```

4. Descendant Selector:
Targets elements that are descendants of a specified element.

```
div p {
/* Styles for <p> elements inside <div> elements */
}
```

5. Child Selector:
Targets elements that are direct children of a specified element.

```
div > p {
/* Styles for <p> elements that are direct children of <div> */
}
```

6. Adjacent Sibling Selector:
Targets an element that is immediately preceded by a specified element.

```
h2 + p {
/* Styles for <p> immediately after <h2> */
}
```

7. Attribute Selector:
Targets elements with a specific attribute or attribute value.

```
input[type="text"] {
/* Styles for <input> elements with type="text" */
}
```

8. Pseudo-classes:
Targets elements based on their state or position in the document.

```
a:hover {
/* Styles for <a> elements when hovered */
}
```

9. Pseudo-elements:
Targets specific parts of elements.

```
p::first-line {
/* Styles for the first line of <p> elements */
}
```

10. Universal Selector:
Targets all elements in the document.

```
* {
/* Styles for all elements */
}
```

These are the basic building blocks of CSS selectors, and you can combine them to create more specific and targeted styles for your HTML elements. Understanding these selectors well can greatly enhance your ability to apply styles effectively across your webpages.

18. What are the media types allowed by CSS?
Ans. CSS supports various media types, allowing styles to be applied selectively based on the output device or medium. Here are some of the media types allowed by CSS:

all: Suitable for all devices.

screen: Intended for screens (desktops, laptops, tablets, smartphones).

print: For printed documents and documents viewed in print preview mode.

speech: Designed for speech synthesizers.

projection: Intended for projected presentations, like slideshows.

handheld: Targeted at handheld devices like smartphones and tablets.

braille: Intended for braille tactile feedback devices.

embossed: Targeted at paged braille printers.

tv: Intended for television-type devices.

aural: For speech synthesizers to render the document.

These media types allow developers to create styles that are optimized for different output devices, ensuring a better user experience across various mediums. You can specify media types in CSS using the @media rule, like so:

```
@media screen {
  /* Styles for screens */
}

@media print {
  /* Styles for print */
}
```

Using media types, you can create responsive designs that adapt to different devices and environments, enhancing accessibility and usability.

19. What is the rule set?

Ans. In CSS, a rule set consists of one or more CSS rules that define how HTML elements should be styled. Each rule set typically contains a selector and a declaration block.

Here's a breakdown of the components of a CSS rule set:

Selector: It specifies the HTML element(s) to which the styles should be applied. Selectors can target elements based on their tag name, class, ID, attributes, or relationships with other elements.

```
/* Example selectors */
h1 {
```

```
    /* Styles for <h1> elements */
}

.example {
    /* Styles for elements with class="example" */
}

#unique {
    /* Styles for element with id="unique" */
}
```

Declaration Block: It consists of one or more declarations enclosed within curly braces { }. Each declaration includes a CSS property and its corresponding value.

```
/* Declaration block */
{
    color: red;
    font-size: 16px;
}
```

Combining the selector and declaration block forms a complete CSS rule set:

```
/* Rule set */
h1 {
    color: red;
    font-size: 24px;
}
```

In this example, the rule set targets all <h1> elements and specifies that they should be styled with red color and a font size of 24 pixels.

CSS rule sets allow web developers to define styles for HTML elements, making it possible to control the appearance and layout of web pages. They provide a powerful mechanism for creating visually appealing and consistent designs across different parts of a website.