WEB DESIGNING ASSIGNMENT

Module(CSS and CSS 3)-2

1. What are the benefits of using CSS?

- CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, etc.
- The following are the benefits of CSS:
    A. **CSS saves time-** You can write CSS code once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many web pages as you want.
    B. **Easy maintenance-** To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
    C. **Global web standards-** Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.
    D. **Platform Independence-** The script offer consistent platform independence and can support latest browsers as well.

2. What are the disadvantage of CSS?

- 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 programing language world is complicated for non-developers and beginners. Different levels of CSS. EX. CSS, CSS 2, CSS 3 are often quite confusing.
- CSS works differently on different browsers. IE and Opera supports CSS as different logic.
- There might be cross-browser issues while using CSS.

3. What is the difference between CSS2 and CSS3?

CSS2	**CSS3**
It was released in 1998 with added styles for other media types so that it can be used for page layout designing.	It was released in 1999 and representation-style properties were added in it that allows you to build a presentation from documents.
It was comprised of a single document.	CSS3 has its specifications divided into many individual modules, which makes CSS3 a whole lot easier to handle.
With CSS2, designers could only use “web-safe fonts” for being 100% sure to use fonts that would always display the same on every machine.	With CSS3, the designers can now use special fonts, like those available in Google Fonts and Typecast.
CSS2 had a ‘simple selectors’.	CSS3 calls the components as ‘a sequence of simple selectors’.
In CSS2, the developers had difficulty because the standard	CSS3 has the capability to split text sections into multiple

was not equipped with automatically breaking the text so that it fits within a box.	columns so that it can be read like a newspaper.

4. Name a few CSS style components?

- The components of css style are:
- **Selector{property:value;}**
    1. **Selector:** HTML element name, id name, class name.
    2. **Property:** It's like an attribute such as background color, font-size, position, text-align, color, border etc.
    3. **Value:** Which defines property or values allocate for properties.

5. What do you understand by CSS opacity?

- The opacity CSS property sets the opacity of an element. Opacity is the degree to which content behind an element is hidden, and is the opposite of transparency.
- Set opacity value in between 0 to 1.
- Example:

```
<!DOCTYPE html>
<html>

<head>
    <title>document</title>
    <style>
        img {
            height: 600px;
            width: 500px;
            opacity: 0.7;
        }
    </style>
</head>

<body>
    <img src="pexels-lucas-pezeta-2280845.jpg" alt="" />
</body>

```

```
</html>
```

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

- EXAMPLE:

```
<!DOCTYPE html>
<html>

<head>
    <title>document</title>
    <style>
        h1 {
            height: 50%;
            width: 50%;
            margin: auto;
            margin-top: 50px;
            border: 10px double black;
            background-color: aquamarine;
        }
    </style>
</head>

<body>
    <h1>Lorem ipsum dolor sit amet consectetur adipisicing elit. Vitae,
labore. Optio possimus nesciunt quas saepe.
        Tempora voluptatibus soluta quos laboriosam molestiae
voluptatum nisi fugiat quia libero, magnam, quo velit cum.
    </h1>
</body>

</html>
```

7. How can image repetition of the backup be controlled?

- To control the repetition of an image in the background, use the **background-repeat** property. You can use no-repeat value for the background-repeat property if you do not want to repeat an image, in this case, the image will display only once.
- Example:

```
<!DOCTYPE html>
<html>

<head>
    <title>Repetition</title>
    <style>
        body {
```

```
-               background-image: url(pexels-lucas-pezeta-2280845.jpg);
-               background-size: 20%;
-               background-repeat: no-repeat;
-           }
-       </style>
-   </head>
-
-   <body>
-       <h1>Hello World</h1>
-   </body>
-
-   </html>
```

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

- The **background-position** property sets the starting position of a background image.
- By default, a background-image is placed at the **top-left** corner of an element, and repeated both vertically and horizontally.
- Example:

```
-   <!DOCTYPE html>
-   <html>
-
-   <head>
-       <title>Repetition</title>
-       <style>
-           body {
-               background-image: url(pexels-lucas-pezeta-2280845.jpg);
-               background-size: 20%;
-               background-repeat: no-repeat;
-               background-attachment: fixed;
-               background-position: right;
-           }
-       </style>
-   </head>
-
-   <body>
-       <h1>Hello World</h1>
-   </body>
-
-   </html>
```

9. Which property controls the image scroll in the background?
   - The **background-attachment** property sets whether a background image scrolls with the rest of the page, or is fixed.
   - Example:

```
<!DOCTYPE html>
<html>

<head>
    <title>Repetition</title>
    <style>
        body {
            background-image: url(pexels-lucas-pezeta-2280845.jpg);
            background-size: 20%;
            background-repeat: no-repeat;
            background-attachment: fixed;
            background-position: right;
        }
    </style>
</head>

<body>
    <h1>Hello World</h1>
</body>

</html>
```

10. Why should background and color be used as separate properties?
   - It enhances the legibility of style sheets. The background property is a complex property in CSS, and if it is combined with color, the complexity will further increase.
   - Color is an inherited property while the background is not. So this can make confusion further.

11. How to center block elements using CSS1?
   - With css the way to center anything that's a block level element is with the margin property. One of the values of margin is auto

and by setting auto on the left and right margin our block level element will center itself.

12. How to maintain the CSS specifications?

- The Specification defines how CSS properties should be implemented by browser vendors along with detailed algorithms, code samples and tabular information.
- The Specification also include:
    - The syntax and data types of the language.
    - Detailed explanation on CSS Selectors.
    - How you can assign value to properties.
    - The cascade
    - How inheritance works
    - The box model

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

- There are three ways to insert css in HTML documents.
    1. **Inline css**:- By using the style attribute inside html particular tag.
    2. **Internal css**:- By using a <style> tag element in the <head> section.
    3. **External css**:- First create an external file filename.css then in to the <head> section using a <link> element to link to an external css file.

14. What is embedded style sheets?

- Embedded style sheets are particularly useful for HTML documents that have unique style requirements from the rest of the documents in your project.
- However, if the styles need to be applied across multiple documents, you should link to an external style sheet instead of using individual embedded style sheets.

- Syntax: The CSS syntax for embedded style sheets is exactly the same as other CSS code, apart from the fact that it is now wrapped within the <style></style> tags. The <style> tag takes the ‘type’ attribute that defines the type of style sheet being used.

15. What are the external style sheets?

- An external style sheet is a separate CSS file that can be accessed by creating a link within the head section of the webpage. Multiple webpages can use the same link to access the stylesheet. The link to an external style sheet is placed within the head section of the page.
- Example:

```
<!DOCTYPE html>
<html>

<head>
    <title>CSS</title>
    <link rel="stylesheet" href="example.css">
</head>

<body>
    <div>
        <h1>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Nihil at deserunt molestias porro quod
            necessitatibus labore asperiores, excepturi aliquam saepe ea, unde nisi voluptatem tenetur tempore.
            Consectetur, nihil! Labore, aut.</h1>
    </div>
</body>

</html>
```

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

- **Advantages:**
    - The style of a few documents can be controlled from the site by utilizing them.

- Multiple HTML elements can have numerous documents, where classes can be made.
- To assemble styles in complex circumstances, selector and grouping strategies are utilized.

- **Disadvantages:**
  - The additional download is expected to import documents having style information.
  - To render the documents, the outer template ought to be stacked.
  - Not practical for small style definitions.

17. What is the meaning of the CSS selector?

- CSS selector selects the HTML elements you want to style. CSS selectors are used to "find" (or select) the HTML elements you want to style.
- We can divide CSS selectors into five categories:
  1. Simple Selectors: - select elements based on name, id, class
  2. Combinator Selectors:- select elements based on a specific relationship between them
  3. Pseudo-class Selectors: - select elements based on a certain state
  4. Pseudo-element Selectors: - select and style a part of an element
  5. Attribute Selectors: - select elements based on an attribute or attribute value

18. What are the media type allowed by CSS?

- You can use the CSS @media at-rule to create media queries that test for certain media types, then apply styles accordingly.
- For ex, you can specify one sets of styles for when your web page is viewed on computer screen and another for when it’s printed out.

- Lists of CSS media type:
    - **all** :- matches all devices.
    - **print** :- matches printers, and devices intended to reproduce a printed display, such as a web browser showing a document in “print priview”.
    - **screen** :- matches all devices that aren’t matched by print or speech.
    - **speech** :- matches screenreaders and other devices that read out the content of a page.

19. What is the ruleset?

- A CSS ruleset is various affirmations to various pieces or elements of the document. The objective is to apply a bunch of properties for certain distinct qualities to a solitary, or a particular arrangement of components in the connected HTML page.
- Ex:

```
.info p {
    color: white;
    padding-bottom: 30px;
    font-size: 0.2in;
    font-family: cursive;
    font-weight: bold;
}
```

20. Create layouts?

- **HTML Code:**

```
<!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>Document</title>
    <link rel="stylesheet" href="css_assignment.css">
```

```
-   </head>
-
-   <body>
-       <section class="main">
-
-           <div class="img1">
-               <img src="img1.jpg" alt="">
-               <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Sint, voluptatem magni. Sint facere eum
-                   minus!
-               </p>
-               <button>View</button>
-               <button>Edit</button>
-           </div>
-           <div class="img1">
-               <img src="img1.jpg" alt="">
-               <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Sint, voluptatem magni. Sint facere eum
-                   minus!
-               </p>
-               <button>View</button>
-               <button>Edit</button>
-           </div>
-           <div class="img1">
-               <img src="img1.jpg" alt="">
-               <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Sint, voluptatem magni. Sint facere eum
-                   minus!
-               </p>
-               <button>View</button>
-               <button>Edit</button>
-           </div>
-
-           <div class="img1">
-               <img src="img1.jpg" alt="">
-               <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Sint, voluptatem magni. Sint facere eum
-                   minus!
-               </p>
-               <button>View</button>
-               <button>Edit</button>
-           </div>
-           <div class="img1">
-               <img src="img1.jpg" alt="">
-               <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Sint, voluptatem magni. Sint facere eum
-                   minus!
-               </p>
```

```
-               <button>View</button>
-               <button>Edit</button>
-           </div>
-           <div class="img1">
-               <img src="img1.jpg" alt="">
-               <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
   Sint, voluptatem magni. Sint facere eum
-                   minus!
-               </p>
-               <button>View</button>
-               <button>Edit</button>
-           </div>
-
-       </section>
-   </body>
-
-   </html>
```

- **CSS Code:**

```
.main {
    /* height: 600px; */
    height: 700px;
    width: 100%;
    background-color: gray;
    background-size: cover;
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    gap: 20px;

}

.main .img1 {
    height: 300px;
    width: 400px;
    margin-top: 30px;
    margin-left: 50px;
    text-align: center;
    background-color: white;
}

.main .img1 img {
    height: 200px;
    width: 300px;
    /* background-size: cover; */
}
```

```css
.main .img1 p {
    font-size: 18px;
    text-align: justify;
}

.main .img1 button {
    font-size: 20px;
    margin-top: 10px;
    padding: 3px;
}
```

- **Output:**

-