Css And Css3

1) What are the benefits of using CSS?

➔ There are a number of benefits of CSS, including:

1) Faster Page Speed

More code means slower page speed. And CSS enables you to use less code. CSS allows you to use one CSS rule and apply it to all occurrences of a certain tag within an HTML document.

2) Better User Experience

CSS not only makes web pages easy on the eye, it also allows for user-friendly formatting. When buttons and text are in logical places and well organized, user experience improves.

3) Quicker Development Time

With CSS, you can apply specific formatting rules and styles to multiple pages with one string of code. One cascading style sheet can be replicated across several website pages. If, for instance, you have product pages that should all have the same formatting, look, and feel, writing CSS rules for one page will suffice for all pages of that same type.

4) Easy Formatting Changes

If you need to change the format of a specific set of pages, it's easy to do so with CSS. There's no need to fix every individual page. Just edit the corresponding CSS stylesheet and you'll see changes applied to all the pages that are using that style sheet.

5) Compatibility Across Devices

Responsive web design matters. In today's day and age, web pages must be fully visible and easily navigable on all devices. Whether mobile or tablet, desktop, or even smart TV, CSS combines with HTML to make responsive design possible.

2) What are the disadvantages of CSS?

-> 1. Come in different levels

There's CSS, CSS 1 up to CSS3, which has resulted in confusion among developers and web browsers. One type of CSS should be enough. It would be preferable than having to choose which CSS level to use.

2. Fragmentation

With CSS, what works with one browser may not always work with another. This is why web developers have to test for compatibility, running the program across multiple browsers before a website is set live. If only people use Mozilla or Chrome, but they don't.

3. Lack of security

Because it is an open text-based system, CSS doesn't have the built-in security that will protect it from being overridden. Anyone who has a read/write access to a website can change the CSS file, alter the links or disrupt the formatting, whether by accident or design.

3) What is the difference between CSS2 and CSS3?

->

CSS2

CSS2 is the improvement of CSS1. It removed the not fully interoperable features. It also included the browser extensions.

It had many new features such as absolute, relative and fixed

positioning of elements. It supported different media types. It also included new font properties such as shadow.

CSS3

CSS3 is the most recent and currently used. It has XHTML specification. CSS3 has its major focus on modularisation and separation of concerns.

Different modules now go through different stages of the recommendation process. CSS3 has support for almost all recent web browsers.

It has even included new selectors along with new combinator and new pseudo-elements.

CSS3 has several new CSS properties. It supports animation which is not a part of earlier recommendations.

There were various properties added such as transforms, gradients, animation and transition for animation effect in the website.

Recent add-ons are like border-radius, box-shadow, flex-box and CSS grid.

4) Name a few CSS style components

->

Some CSS Style components are:

- Selector
- Property

- Value

5) What do you understand by CSS opacity?

->

The opacity property sets the opacity level for an element.

The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see-through, and 0 is completely transparent.

```
<style>
img.trans {
    opacity: 0.4;
    filter: alpha(opacity=40); /* For IE8 and earlier */
}
</style>
```

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

->

In CSS, we use the background-color property to specify the background color of an element.

The *value* of the *background-color* property can be specified in three ways:

1. Using *RGB values* e.g. rgb(250,0,0), rgb(0,250,0), etc.
2. Using a *Hexadecimal(HEX) value* e.g. #FFFFFF, #000000, etc.
3. Using a valid *color name* e.g. “white”, “red”, “black”, “blue”, etc.

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

->

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.

Syntax:

background-repeat: repeat|repeat-x|repeat-y|no-repeat|initial|inherit;

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

->

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;

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.

CSS Syntax

background-attachment: scroll|fixed|local|initial|inherit;

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

->

There are two reasons behind this:

- 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?

->

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.

Syntax:

```
element {
   width:200px;
```

```
    margin: auto;
}
```

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 values to properties
- The Cascade (the "C" in CSS)
- How inheritance works
- The Box Model e.t.c

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

->

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the style attribute inside HTML elements
- **Internal** - by using a <style> element in the <head> section
- **External** - by using a <link> element to link to an external CSS file

Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the style attribute of an HTML element.

Example:

```
<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>
```

Internal CSS

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the <head> section of an HTML page, within a <style> element.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

```
</body>
</html>
```

External CSS

An external style sheet is used to define the style for many HTML pages.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Style.css

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
```

```
  color: red;
}
```

14) What is embedded style sheets?

->

Embedded Stylesheet: Embedded style sheets refer to when you embed style sheet information into an HTML document using the <style> element. You do this by embedding the style sheet information within <style></style> tags in the head of your document.

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 (ie. text/CSS).

Example 1:

```
<!DOCTYPE html>
<html>

<head>
   <title>Page Title</title>

   <!-- Embedded stylesheet -->
   <style>
      h2 {
         font-size: 1.5rem;
         color: #2f8d46;
         text-align: center;
      }
```

```
        p {
            font-variant: italic;
        }
    </style>
</head>

<body>
    <h2>Welcome To GFG</h2>
    <p>This document is using an embedded stylesheet!</p>
    <p>This is a paragraph</p>
    <p>This is another paragraph</p>
</body>

</html>
```

15) What are the external style sheets?

->

An external style sheet is a separate file where you can declare all the styles that you want to use on your website. You then link to the external style sheet from all your HTML pages.

This means you only need to set the styles for each element once. If you want to update the style of your website, you only need to do it in one place.

Create the Style Sheet

Type CSS code into a plain text file, and save with a .css extension (for example, styles.css).

Here's an example of some CSS code.

```
body {
```

```
    background-color: darkslategrey;
    color: azure;
    font-size: 1.1em;
}
h1 {
    color: coral;
}
#intro {
    font-size: 1.3em;
}
.colorful {
    color: orange;
}
```

Link to the Style Sheet from the HTML Documents

Add the following code between the <head></head> tags of all HTML documents that you want to reference the external style sheet. This code uses the HTML <link> element to link to the external style sheet.

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Example</title>
        <link rel="stylesheet" href="styles.css">
```

```
</head>
<body>
    <h1>External Styles</h1>
    <p id="intro">Allow you to define styles for the whole website.</p>
    <p class="colorful">This has a style applied via a class.</p>
</body>
</html>
```

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

->

The advantages of External Style Sheets are as follows :

- With the help of External Style Sheets, the styles of numerous documents can be organized from one single file.
- In External Style Sheets, Classes can be made for use on numerous HTML element types in many forms of the site.
- In complex contexts, Methods like selector and grouping can be implemented to apply styles.

The disadvantages of External Style Sheets are as follows :

- An extra download is essential to import style information for each file.

- The execution of the file may be deferred till the external style sheet is loaded.
- While implementing style sheets, we need to test Web pages with multiple browsers in order to check compatibility issues.

17) What is the meaning of the CSS selector?

->

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

The CSS element Selector

The element selector selects HTML elements based on the element name.

Example:

```
p {
  text-align: center;
```

```
  color: red;
}
```

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example:

```
#para1 {
  text-align: center;
  color: red;
}
```

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

Example:

```
.center {
  text-align: center;
  color: red;
}
```

The CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

Example:

```
* {
  text-align: center;
  color: blue;
}
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Example:

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

18) What are the media types allowed by CSS?

->

One of the most important features of style sheets is that they specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, etc.

We have currently two ways to specify media dependencies for style sheets –

- Specify the target medium from a style sheet with the @media or @import at-rules.
- Specify the target medium within the document language.

The @media rule

An *@media* rule specifies the target media types (separated by commas) of a set of rules.

Given below is an example −

```
<style tyle = "text/css">
  <!--
    @media print {
      body { font-size: 10pt }
    }

    @media screen {
      body { font-size: 12pt }
    }
    @media screen, print {
      body { line-height: 1.2 }
    }
  -->
</style>
```

The Document Language

In HTML 4.0, the *media* attribute on the LINK element specifies the target media of an external style sheet −

Following is an example −

```
<style tyle = "text/css">
  <!--
    <!doctype html public "-//w3c//dtd html 4.0//en">
    <html>
      <head>
        <title>link to a target medium</title>
        <link rel = "stylesheet" type = "text/css" media = "print,
```

```
        handheld" href = "foo.css">
    </head>

    <body>
      <p>the body...
    </body>
  </html>
 -->
</style>
```

Recognized Media Types

The names chosen for CSS media types reflect target devices for which the relevant properties make sense. They give a sense of what device the media type is meant to refer to. Given below is a list of various media types −

Sr.No.	Value & Description
1	**all** Suitable for all devices.
2	**aural** Intended for speech synthesizers.
3	**braille** Intended for braille tactile feedback devices.
4	**embossed** Intended for paged braille printers.
5	**handheld**

	Intended for handheld devices (typically small screen, monochrome, limited bandwidth).
6	**print** Intended for paged, opaque material and for documents viewed on screen in print preview mode. Please consult the section on paged media.
7	**projection** Intended for projected presentations, for example projectors or print to transparencies. Please consult the section on paged media.
8	**screen** Intended primarily for color computer screens.
9	**tty** Intended for media using a fixed-pitch character grid, such as teletypes, terminals, or portable devices with limited display capabilities.
10	**tv** Intended for television-type devices.

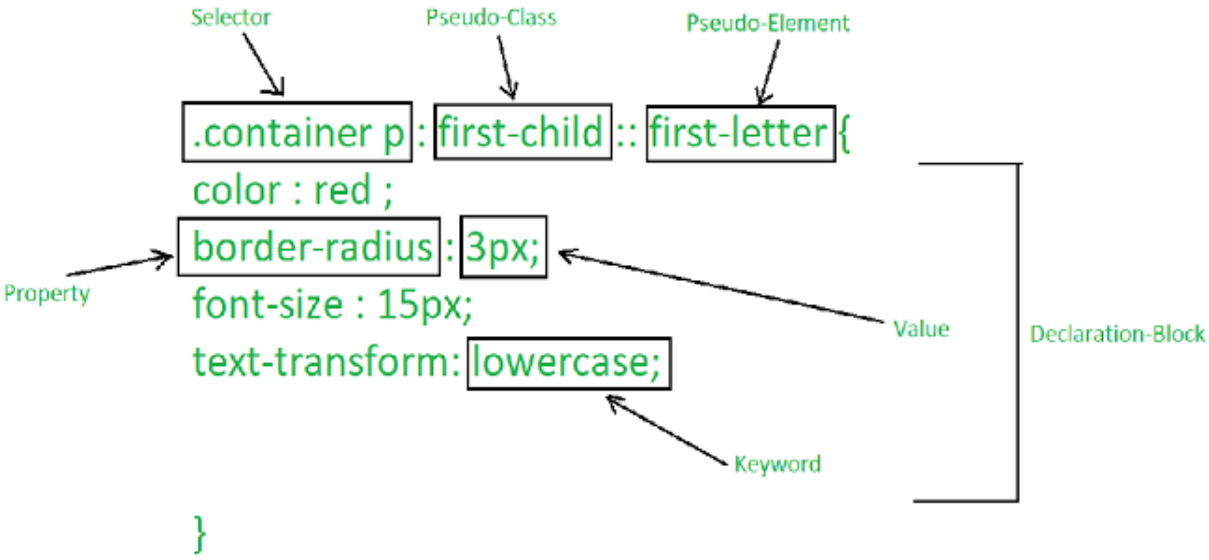
19) What is the rule set?

->

A CSS rule set contains one or more selectors and one or more declarations. The selector(s), which in this example is h1, points to an HTML element. The declaration(s), which in this example are color: blue and text-align: center style the element with a property and value. The rule set is the main building block of a CSS sheet.

Example:

```
h1 {
  color: blue;
  text-align: center;
}
```



20) Create Layouts

->

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>
    <style>
        *{
            box-sizing: border-box;
            background: lightblue;
        }
        .box
        {
            display: flex;
            flex-wrap: wrap;
            padding: 2px;
        }
        .main
        {
            height: 300px;
```

```
        width: 400px;
        margin: 20px;
    }
    .main .first
    {
        height: 180px;
        width: 100%;
        background: gray;
        font-size: 30px;
        display: flex;
        justify-content: center;
        align-items: center;
        color: white;
        font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial,
sans-serif;
    }
    .main .second
    {
        height: 120px;
        width: 100%;
        background: white;
```

```
    padding: 15px 30px 15px 15px;


}
.main .second .button
{
    height: 30px;
    width: 130px;
    background: transparent;
    margin: 10px 0 0 0;
    display: flex;
    font-size: 20px;
}
.button .view
{
    width: 55%;
    background: transparent;
    border: 2px solid;
    border-right: 0;
    padding: 2px 13px;
}
.button .edit
```

```
        {
            width: 45%;
            background: transparent;
            border: 2px solid;
            padding: 2px 10px;
        }
    </style>
</head>
<body>
<div class="box">
    <div class="main">
        <div class="first">
            Thumbnail
        </div>
        <div class="second">
            Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
            <div class="button">
                <div class="view">
                    View
                </div>
```

```
            <div class="edit">
                Edit
            </div>
        </div>
    </div>
</div>


<div class="main">
    <div class="first">
        Thumbnail
    </div>
    <div class="second">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
        <div class="button">
            <div class="view">
                View
            </div>
            <div class="edit">
                Edit
            </div>
```

```
        </div>
    </div>
</div>


<div class="main">
    <div class="first">
        Thumbnail
    </div>
    <div class="second">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
        <div class="button">
            <div class="view">
                View
            </div>
            <div class="edit">
                Edit
            </div>
        </div>
    </div>
</div>
```

```
<div class="main">
    <div class="first">
        Thumbnail
    </div>
    <div class="second">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
        <div class="button">
            <div class="view">
                View
            </div>
            <div class="edit">
                Edit
            </div>
        </div>
    </div>
</div>


<div class="main">
    <div class="first">
```

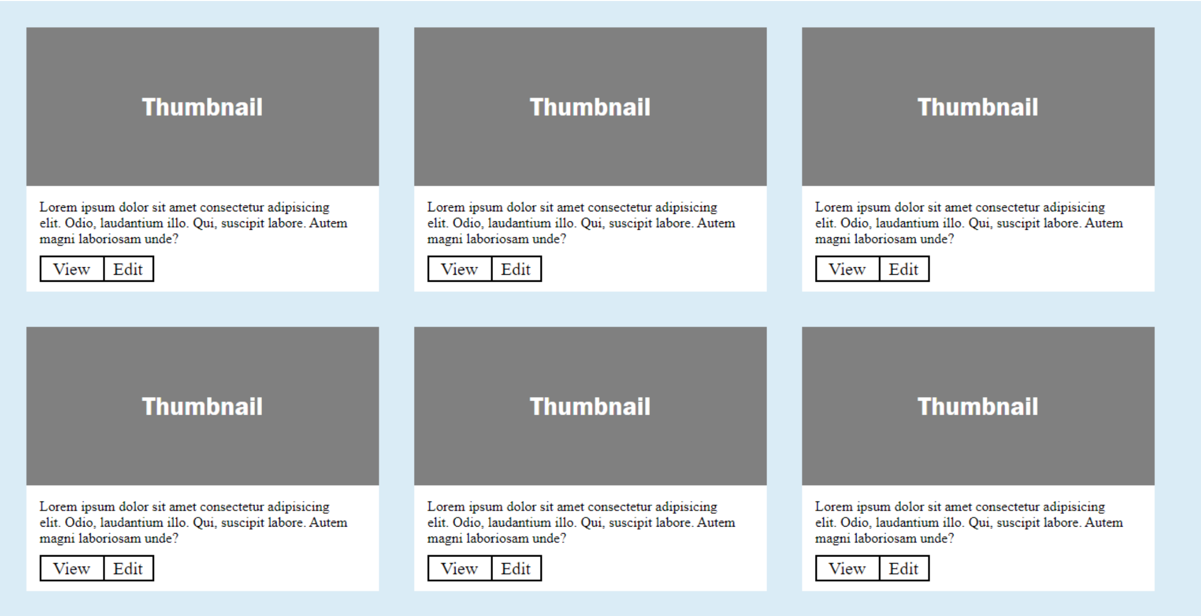
```
        Thumbnail
    </div>
    <div class="second">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
        <div class="button">
            <div class="view">
                View
            </div>
            <div class="edit">
                Edit
            </div>
        </div>
    </div>
</div>

<div class="main">
    <div class="first">
        Thumbnail
    </div>
    <div class="second">
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?

```
        <div class="button">
            <div class="view">
                View
            </div>
            <div class="edit">
                Edit
            </div>
        </div>
    </div>
  </div>
</div>
</body>
</html>
```

Output:

Css And Css3


Thumbnail
Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
View
Edit
Thumbnail
Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
View
Edit
Thumbnail
Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
View
Edit
Thumbnail
Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
View
Edit
Thumbnail
Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
View
Edit
Thumbnail
Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio, laudantium illo. Qui, suscipit labore. Autem magni laboriosam unde?
View
Edit