CSS Selectors & Styling

**1 What is a CSS selector? Provide examples of element, class, and ID selectors**

A CSS selector is the first part of a CSS Rule. It is a pattern of elements and other terms that tell the browser which HTML elements should be selected to have the CSS property values inside the rule applied to them.

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

The CSS rule below will be applied to the HTML element with id="para1":

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

The CSS element Selector

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

Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

p {  
  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

In this example all HTML elements with class="center" will be red and center-aligned:

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

**2 Explain the concept of CSS specificity. How do conflicts between multiple stylesget resolved?**

Specificity is the algorithm used by browsers to determine the CSS declaration that is the most relevant to an element, which in turn, determines the property value to apply to the element.

In CSS, styles sheets cascade by order of importance. If rules in different style sheets conflict with

one another, the rule from the most important style sheet wins.

The order in which stylesheets are linked in the HTML document influences the application of styles. Styles from a stylesheet loaded later can override those from an earlier one. By organizing the order of stylesheet inclusion, you control which styles have priority, helping to manage and resolve conflicts.

3 What is the difference between internal, external, and inline CSS? Discuss the advantages and disadvantages of each approach.

Css is used to add styles on web pages that contain [HTML elements](https://www.geeksforgeeks.org/html-elements/). There are three methods to add styles on web pages, these are – Inline, Internal, and External. In this article, we will see the differences between Inline, Internal, and External CSS styles.

Table of Content

* [Inline CSS](https://www.geeksforgeeks.org/difference-between-inline-internal-and-external-css/#inline-css)
* [Internal CSS](https://www.geeksforgeeks.org/difference-between-inline-internal-and-external-css/#internal-css)
* [External CSS](https://www.geeksforgeeks.org/difference-between-inline-internal-and-external-css/#external-css)

Internal CSS

Internal or embedded CSS requires you to add a <style> tag in the <head> section of your HTML document.

Here’s how you can use internal CSS:

1. Open your HTML page and locate <head> opening tag.
2. Put the following code right after the <head> tag

Advantages of Internal CSS:

* You can use class and [ID selectors](https://www.bitdegree.org/learn/css-id) in this style sheet, like the following example:

.class {

property1 : value1;

property2 : value2;

property3 : value3;

}

#id {

property1 : value1;

property2 : value2;

property3 : value3;

Disadvantages of Internal CSS:

* Adding the code to the HTML document can increase the page’s size and loading time.

Inline CSS

Inline CSS is used to style a specific HTML element. For this CSS style, you’ll only need to add the style attribute to each HTML tag, without using selectors.

Advantages of Inline CSS:

* You can easily and quickly insert CSS rules into an HTML page. That’s why this method is useful for testing or previewing the changes and performing quick fixes to your website.
* You don’t need to create and upload a separate document as in the external style.

Disadvantages of Inline CSS:

* Adding CSS rules to every HTML element is time-consuming and makes your HTML structure messy.
* Styling multiple elements can affect your page’s size and download time.

External CSS

With external CSS, you’ll link your web pages to an external .css file, which can be created by any text editor in your device (e.g., [Notepad++](https://notepad-plus-plus.org/)).

This CSS type is a more efficient method, especially for styling a large website. By editing one .css file, you can change your entire site at once.

Advantages of External CSS:

* Since the CSS code is in a separate document, your HTML files will have a cleaner structure and are smaller in size.
* You can use the same .css file for multiple pages.

Disadvantages of External CSS:

* Your pages may not be rendered correctly until the external CSS is loaded.
* Uploading or linking to multiple CSS files can increase your site’s download time.CSS Box Model

**1 Explain the CSS box model and its components (content, padding, border, margin). How does each affect the size of an element?**

The CSS Box Model defines how elements are sized, positioned, and rendered on a webpage. When a browser loads an HTML document, it creates a [DOM tree](https://www.geeksforgeeks.org/dom-document-object-model/)and assigns a box to each element. This box calculates the [element’s dimensions](https://www.geeksforgeeks.org/how-to-get-the-elements-actual-width-and-height-in-javascript/) and [position relative](https://www.geeksforgeeks.org/difference-between-relative-and-absolute-position-in-css/) to its parent or the root <html> element, ensuring accurate layout and spacing.

1. Content Area

* The content area is the central part of the CSS box model, containing the [main content](https://www.geeksforgeeks.org/html-main-tag/) (e.g., text, images, videos, or elements like <p> or <span>).
* It can be styled with [CSS properties](https://www.geeksforgeeks.org/css-properties-complete-reference/)like height and width.

The content edge refers to the four edges of the content area

* Left content edge
* Right content edge
* Top content edge
* Bottom content edge

2. Padding Area

* The padding area is the space between the content and the border of an element.
* It includes the areas highlighted in light green and skin color in the example.
* The distance between the [content edge](https://www.geeksforgeeks.org/what-does-meta-http-equivx-ua-compatible-contentieedge-do/)and the border is the padding.
* The border marks the end of the padding area.
* The padding area contributes to the element’s total dimensions.
* Padding can be adjusted using [CSS properties](https://www.geeksforgeeks.org/css-properties-complete-reference/).
* It works similarly with box-sizing: content-box and box-sizing: border-box, but with slight calculation differences.

3. Border Area

* The area that marks the end of an element is called as the[border](https://www.geeksforgeeks.org/css-borders/)it is the outer fencing for the element.
* The default border properties are provided in CSS to control the thickness of this outer fencing.
* The border area also add ‘s up to the complete[height and width](https://www.geeksforgeeks.org/how-to-set-the-width-and-height-of-an-image-using-html/)of the element.
* The more the border [width](https://www.geeksforgeeks.org/css-width-property/) the more will be the height or width of the element.
* In the above image the area marked with skin color is called the border area.

4. Margin Area

* The area outside the border of an element is called the [margin area](https://www.geeksforgeeks.org/css-box-model/).
* Basically this area depends on the parent of the element.
* The distance between the border of the parent element and the border of the child element is called as the margin.
* CSS has provides certain [margin properties](https://www.geeksforgeeks.org/css-margins-padding/)to get control over this scenario.

Factors affecting the atomic size are:

1. Number of shells: As number of shells increases atomic size i.e. the distance of the outermost shell from the nucleus also increases.

2. Nuclear charge: As nuclear charge increases atomic size decreases.

**2. What is the difference between border-box and content-box box-sizing inCSS? Which is the default?**

*border-box*and *content-box* are the two different values of [*box-sizing*](https://www.geeksforgeeks.org/css-box-sizing-property/).

* content-box: This is the default value of box-sizing. The dimension of element only includes ‘height’ and ‘width’ and does not include ‘border’ and ‘padding’ given to element. Padding and Border take space outside the element.
* border-box: In this value, not only width and height properties are included but you will find padding and border inside of the box for example .box {width: 200px; border: 10px solid black;} renders a box that is 200px wide.

This is often considered the default value.

* It calculates the total width and height of an element by considering only the content, excluding padding

CSS Flexbox

**1. What is CSS Flexbox, and how is it useful for layout design? Explain the terms flex-container and flex-item.**

Flexbox is short for the Flexible Box Layout module.

Flexbox is a layout method for arranging items in rows or columns.

Flexbox makes it easier to design a flexible responsive layout structure, without using float or positioning.

Before the Flexible Box Layout module, there were four layout modes:

* Block, for sections in a webpage
* Inline, for text
* Table, for two-dimensional table data
* Positioned, for explicit position of an element

CSS flexbox is supported in all modern browsers.

CSS Flexbox Components

A flexbox always consists of:

* a Flex Container - the parent (container) <div> element
* Flex Items - the items inside the container <div>

Flex Container

* To start using CSS Flexbox, you need to first define a flex container.
* The flex container becomes flexible by setting the display property to flex.

Flex Items

The direct child elements of a flex container automatically becomes flex items.

**3 Describe the properties justify-content, align-items, and flex- direction used in Flexbox.**

The justify-content property aligns the flexible container's items when the items do not use all available space on the main-axis (horizontally).

Example

Align the flex items at the center of the container:

div {  
  display: flex;  
  justify-content: cen

CSS Align-Items

The align-items CSS property specifies the alignment for the items inside a flexbox or a grid container.

Syntax:

align-items:

normal | stretch | *positional alignment* | flex-start | flex-end | baseline | initial | inherit;

Flex direction establishes the axis all content within the parent flex container displays itself on. Flexbox allows you to set a single-direction layout.

The two directions can either be horizontal (row) or vertical (column).

The default value of every container with display: flex defined is flex-direction: row.

CSS Grid

**1 Explain CSS Grid and how it differs from Flexbox. When would you use Grid over Flexbox?**

Difference between CSS Grid and Flexbox

CSS Grid Layout

CSS [Grid Layout](https://www.geeksforgeeks.org/css-grid-property/), is a two-dimensional grid-based layout system with rows and columns. It makes easier to design web pages without having to use floats and positioning. Like tables, grid layout allow us to align elements into columns and rows.

To get started, you have to define a container element as a grid with [display: grid](https://www.geeksforgeeks.org/css-grid-property/), set the column and row sizes with grid-template-columns and grid-template-rows, and then place its child elements into the grid with grid-column and grid-row.

CSS Flexbox

The [CSS Flexbox](https://www.geeksforgeeks.org/introduction-to-css-flexbox/) offers one-dimensional layout. It is helpful in allocating and aligning the space among items in a container (made of grids). It works with all kinds of display devices and screen sizes. To get started you have to define a container element as a grid with [display: flex](https://www.geeksforgeeks.org/introduction-to-css-flexbox/);

The basic difference between CSS grid layout and [CSS flexbox layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flexible_box_layout) is that flexbox was designed for layout in one dimension - either a row *or* a column. Grid was designed for two-dimensional layout - rows, and columns at the same time.

The two specifications share some common features, however, and if you have already learned how to use flexbox, the similarities should help you get to grips with grid.

**2 : Describe the grid-template-columns, grid-template-rows, and grid-gap properties. Provide examples of how to use them.**

The grid-template-columns

The grid-template-columns property defines the number of columns in your grid layout, and it can define the width of each column.

The value is a space-separated-list, where each value defines the width of the respective column.

Example

Make a grid with 5 columns:

.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto auto auto;  
}

The grid-template-rows

The grid-template-rows property defines the height of each row.

Example

.grid-container {  
  display: grid;  
  grid-template-rows: 80px 200px;  
}

The gap property

Defines the size of the gap between the rows and between the columns in flexbox, grid or multi-column layout.

It is a shorthand for the following properties:

* [row-gap](https://www.w3schools.com/cssref/css3_pr_row-gap.php)
* [column-gap](https://www.w3schools.com/cssref/css3_pr_column-gap.php)

Syntax:

grid-gap: grid-row-gap grid-column-gap;

Property Values:

* [grid-row-gap](https://www.geeksforgeeks.org/css-grid-row-gap-property/): Sets the size of the gap between the rows in a grid layout. The default value is 0
* [grid-column-gap](https://www.geeksforgeeks.org/css-grid-column-gap-property/): Sets the size of the gap between the columns in a grid layout. The default value is 0.

Responsive Web Design with Media Queries

1 What are media queries in CSS, and why are they important for responsive design?

Media queries in CSS are a feature that allows developers to apply styles conditionally based on the characteristics of the user's device or viewport, such as screen size, resolution, orientation, or color depth. They enable responsive design by allowing the layout and styling of a webpage to adapt dynamically to different devices and screen sizes.

Importance in Responsive Design:

1. Improved User Experience: By adapting the design to fit the screen size and orientation, media queries ensure content is easily readable and navigable on all devices.
2. Device Independence: They allow a single website to function well across a variety of devices, including smartphones, tablets, and desktops, reducing the need for multiple versions of the site.
3. SEO Benefits: Search engines prefer mobile-friendly websites, and responsive design can improve search rankings.
4. Future-Proofing: Media queries help accommodate new devices with varying screen sizes and resolutions.

2 Basic Media Query for Adjusting Font Size

Here’s a simple example of a media query that reduces the font size for screens smaller than 600px:

css

Copy code

/\* Default font size \*/

body {

font-size: 16px;

}

/\* Media query for screens smaller than 600px \*/

@media (max-width: 600px) {

body {

font-size: 14px;

}

}

In this example:

* The default font size is 16px.
* When the viewport width is 600px or smaller, the font size is adjusted to 14px.

Typography and Web Fonts

1 Explain the difference between web-safe fonts and custom web fonts. Why might you use a web-safe font over a custom font?

Web-safe fonts:

Web-safe fonts are fonts that are universally pre-installed on most devices and operating systems. Examples include Arial, Times New Roman, and Courier New. Since these fonts are already available on users' devices, they do not require additional downloads.

Custom web fonts:

Custom web fonts are not pre-installed on devices. They are downloaded from a web server when the webpage is loaded. Examples include fonts hosted on platforms like Google Fonts (e.g., Roboto, Open Sans) or Adobe Fonts.

Key differences:

1. Availability:
   * Web-safe fonts: Pre-installed and always available.
   * Custom web fonts: Need to be loaded from a server.
2. Performance:
   * Web-safe fonts load instantly since they are local to the user's device.
   * Custom web fonts may introduce slight delays as they are fetched from a server.
3. Design Flexibility:
   * Web-safe fonts offer limited design choices.
   * Custom web fonts provide a vast array of styles and weights, enhancing creative freedom.

Why use a web-safe font over a custom font?

* Performance: Faster loading times and reduced bandwidth usage.
* Fallback: Ensures text is rendered correctly even in environments with limited internet connectivity.
* Compatibility: Guaranteed to work across all devices without requiring external resources.

2 What is the font-family property in CSS? How do you apply a custom Google Font to a webpage?

font-family property:

The font-family property in CSS specifies the typeface or a list of fallback fonts for text content. It allows developers to define the preferred font followed by alternative options in case the first choice isn't available.

css

Copy code

p {

font-family: "Roboto", Arial, sans-serif;

}

In this example, "Roboto" is the primary font. If unavailable, the browser falls back to Arial, then to any generic sans-serif font.

Applying a custom Google Font to a webpage:

1. Include the Google Font in the HTML: Use a <link> tag in the <head> section of your HTML to load the font.

html

Copy code

<head>

<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap" rel="stylesheet">

</head>

1. Use the font in your CSS: Reference the font in your font-family property.

css

Copy code

body {

font-family: "Roboto", sans-serif;

}

By combining these steps, the custom Google Font will be applied to the webpage, enhancing its typography.

Module 5-Frontend - HTML5

**1 Difference Between HTML and HTML5**

HTML (HyperText Markup Language):

* The standard language for creating web pages.
* Initial versions lacked support for modern multimedia, advanced form controls, and semantic tags.
* Relied on external plugins (e.g., Flash) for features like audio, video, and animations.

HTML5:

* The fifth revision of HTML, introduced in 2014.
* Designed to support modern web technologies, making it more efficient and interactive.
* Provides enhanced multimedia support and advanced features natively without plugins.

**2 What Are the Additional Tags Used in HTML5?**

HTML5 introduced new tags to enhance structure, multimedia, and functionality. These tags can be grouped into categories:

1. Semantic Tags:

Provide better structure and meaning to web documents:

* <header>: Defines the header section.
* <footer>: Defines the footer section.
* <article>: Represents an independent piece of content.
* <section>: Groups related content together.
* <aside>: Content tangentially related to the main content.
* <nav>: Navigation links section.
* <main>: The main content of the document.

2. Multimedia Tags:

Enable native support for media:

* <audio>: Embeds audio content.
* <video>: Embeds video content.
* <source>: Specifies multiple media resources.
* <track>: Adds text tracks to videos (e.g., subtitles).

3. Graphics and Scripting Tags:

Support interactive and graphical content:

* <canvas>: A container for drawing graphics with JavaScript.
* <svg>: Embeds scalable vector graphics.

4. Form-Related Tags:

Enhance form functionality:

* <datalist>: Provides a dropdown list of predefined options.
* <keygen> (deprecated): Generates a key-pair for forms.
* <output>: Displays the result of a calculation or script.

5. Structural Tags:

* <figure>: Groups media elements like images and their captions.
* <figcaption>: Caption for a <figure> element.
* <mark>: Highlights text.
* <progress>: Represents the progress of a task.
* <meter>: Displays a scalar measurement within a known range.
* <time>: Represents a point in time or duration.

These additions make HTML5 more versatile, functional, and aligned with modern web development practices.

Bottom of Form