Lesson:

Chrome dev tools for CSS

PW SKILLS

Topics

- Introduction to the Chrome dev tool for CSS
- Inspecting and selecting the HTML elements
- Applying temporary CSS changes for testing
- Using box model tools for layout and positioning
- Using the Color picker and Color contrast analyzer
- Working with media Query and responsive designs
- Testing the CSS flexbox and grid layout

#pre-requisite - Chrome browser should be installed in the system.

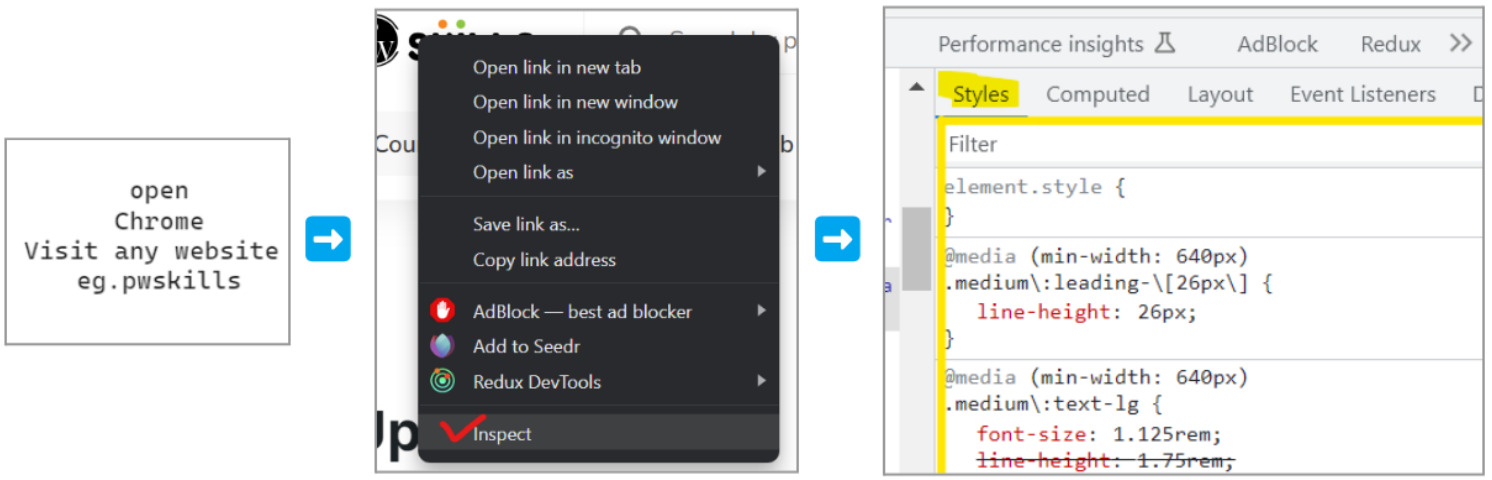
Introduction to the Chrome dev tool for CSS

The Chrome DevTools CSS panel is a powerful tool that allows developers to inspect and modify the CSS of a web page. It can be used to -

- View the CSS properties of any element on the page
- Find and fix CSS errors
- Override CSS rules
- Experiment with different CSS styles
- Preview CSS changes before they applied to the page

To open the Chrome dev tools CSS section

Go to ChromeBrowser>enter any browserPage>right click>select inspect>styles. i.e.

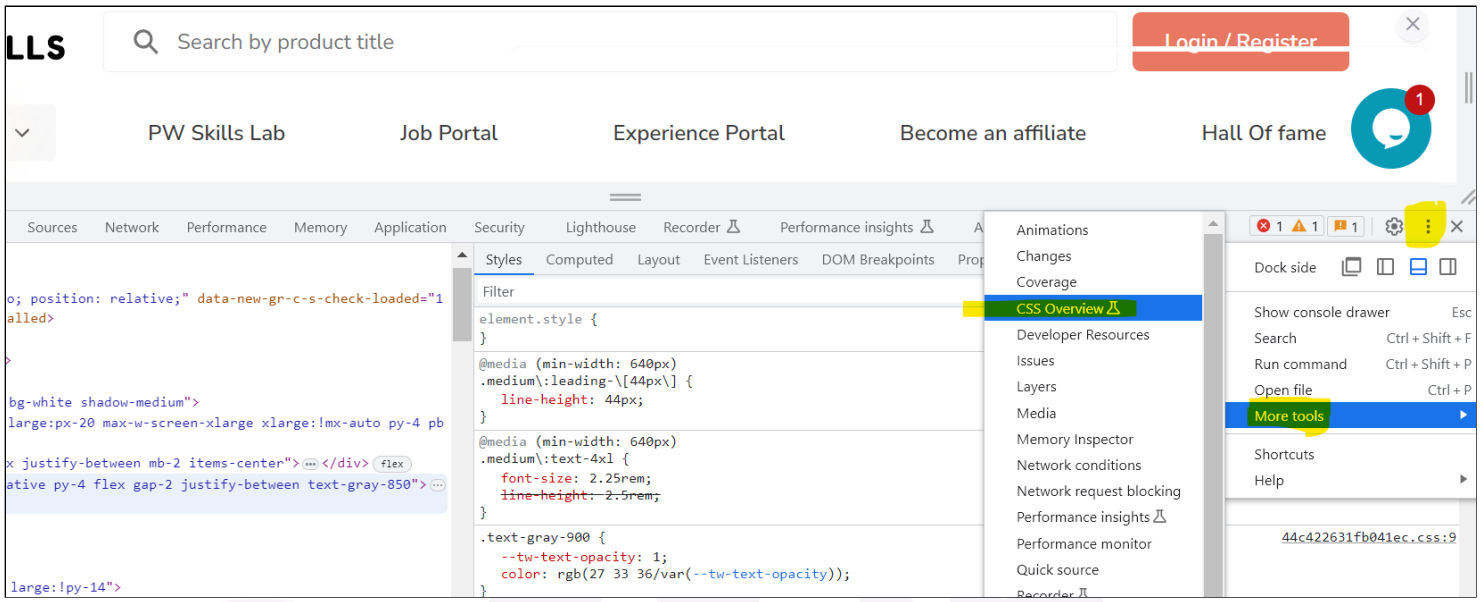


Open the CSS overview panel

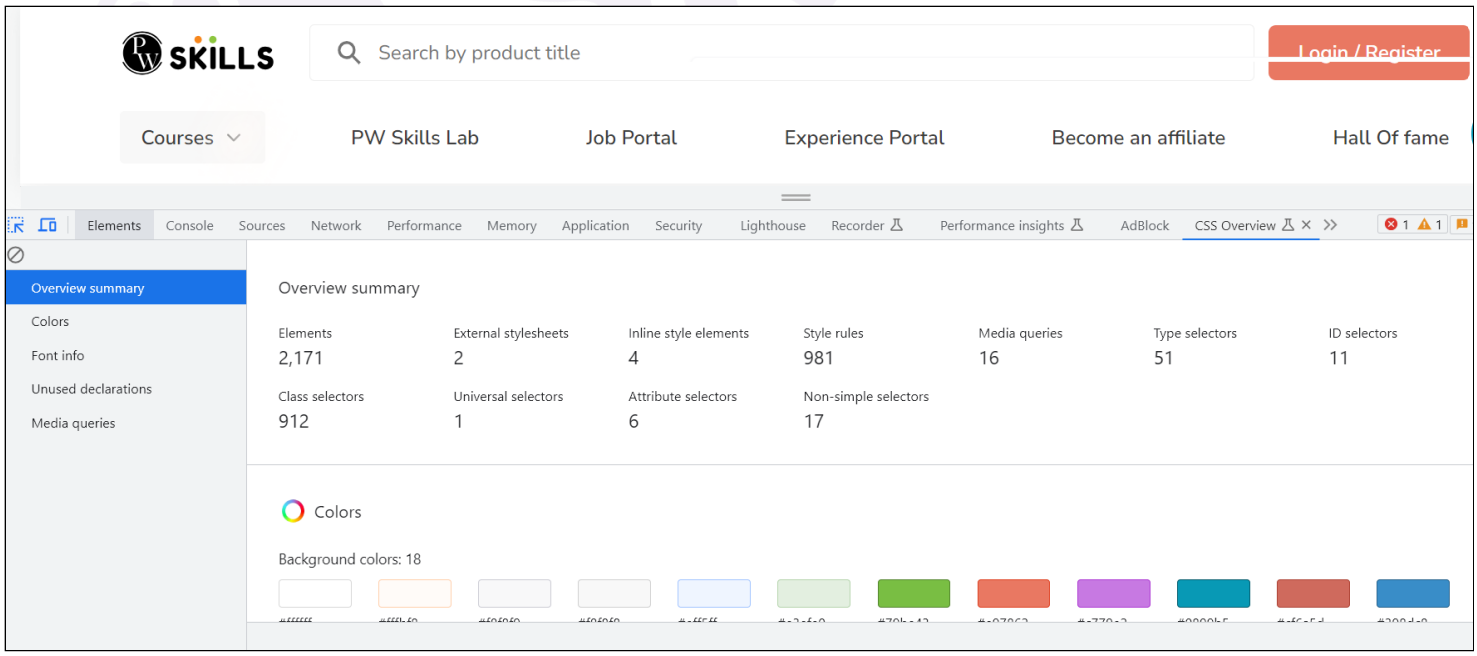
The “CSS overview” tool in Chrome Devtools provides a high-level summary of the CSS on a webpage, helping developers to identify potential performance issues.

Inspect>three do(right top corner of the chrome dev tool)>more tools>CSS Overview.

i.e.



This is the CSS overview panel



Inspecting and selecting the HTML elements

The HTML element can be inspected by right-clicking on the particular HTML element and selecting the inspect option on the web pages.

For example -
We can change the pwskills Hero section text i.e “upscaling made” to any other text
Go to pwskills>select the element to e change>right click and select inspect

Result of what we got after entering the inspection mood

Now double clicking of the h1 value from the above picture. The yellow underline word can be replaced.

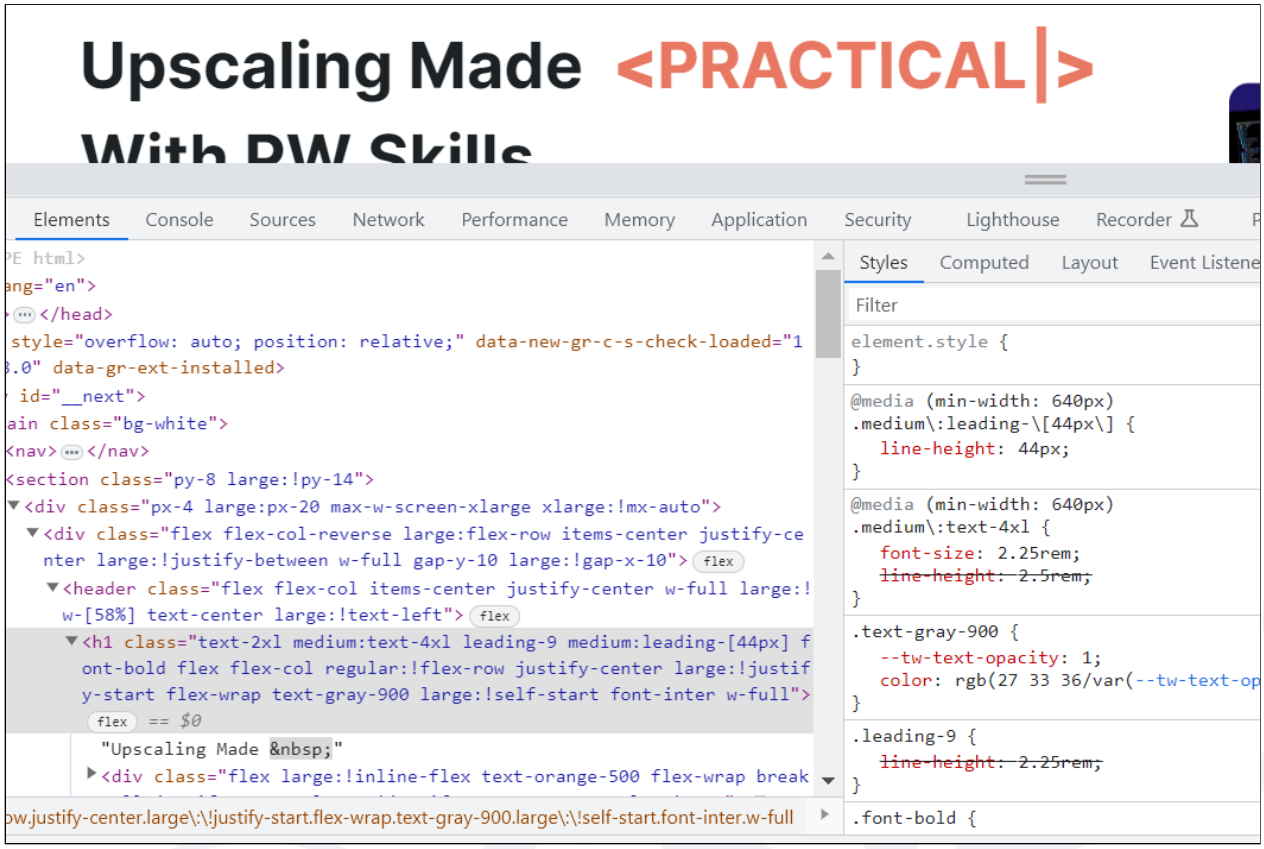
Applying temporary CSS changes for testing

Now let's see how can we change the colour of the above example temporarily for testing. It will not be saved permanently.

After selecting the element that is the h1 HTML element tag, the colour of the text can be changed in the style panel for CSS.

i.e

Before -



After - temporarily changing the CSS style (color to red)

Using box model tools for layout and positioning

The box model is a CSS concept that describes the size and position of an HTML element. It consists of four parts:

- Content: This is the actual content of the element, such as text, images, or other media.
- Padding: This is the space between the content and the border of the element.
- Border: This is the line that surrounds the element.
- Margin: This is the space between the border of the element and other elements on the page.

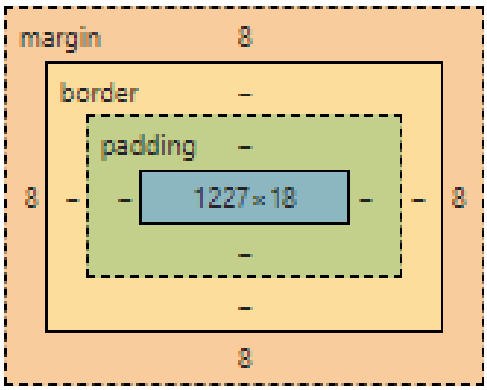
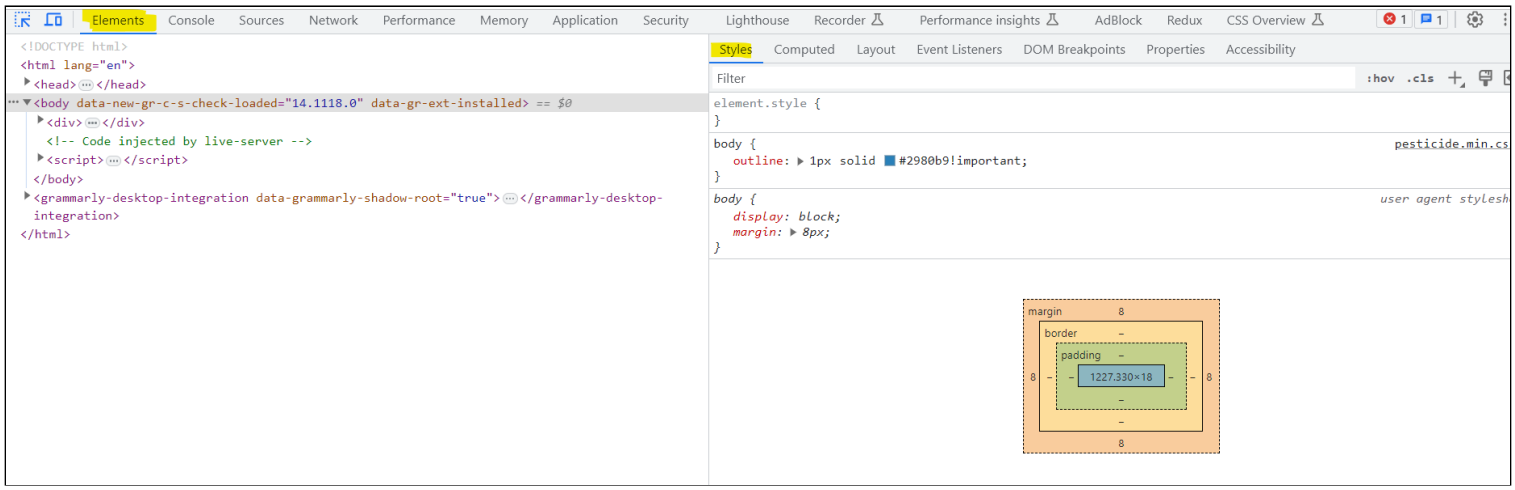


Fig. Box model figure

The box model can be used to control the layout and positioning of HTML elements in a number of ways. For example, you can use the padding property to add space around an element, the border property to create a border around an element, and the margin property to space an element away from other elements by double-clicking on the respective property.

Box model can be accessed by opening the Chrome dev tools, selecting the element panel and then clicking the style panel and scrolling down through the bottom the Box model will be available.

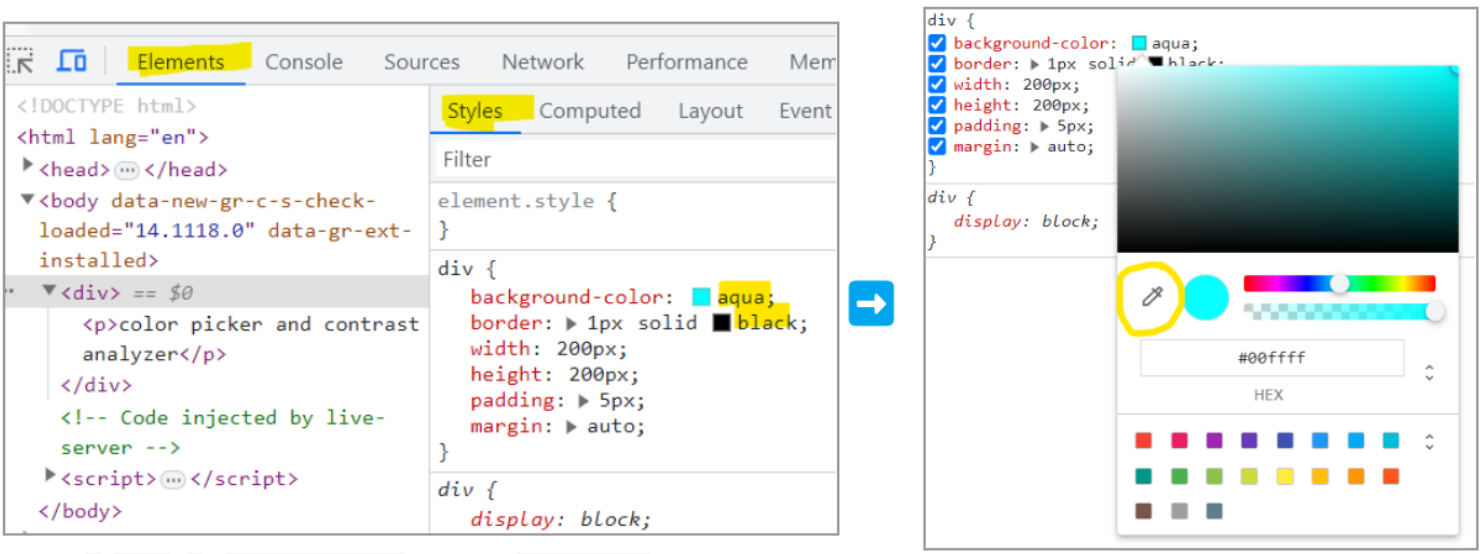
i.e.



Using the Color Picker and Color contrast analyzer Color Picker -

A color picker is a tool that allows you to select specific colors from a visual interface, typically using a graphical representation like a color wheel or a spectrum. Designers and developers use color pickers to choose colors for various elements in their projects, such as website backgrounds, text, buttons, and more. Color pickers often provide different formats for color representation, including hexadecimal (e.g., #RRGGBB), RGB (red, green, blue), and HSL (hue, saturation, lightness).
Color Picker can be accessed from the Chrome dev tools inside the style panel. i.e.

Open Chrome dev tools > select the element > navigate to the styles panel > double click on the color box prefix to any color name than the color panel will be shown along with the color Picker.



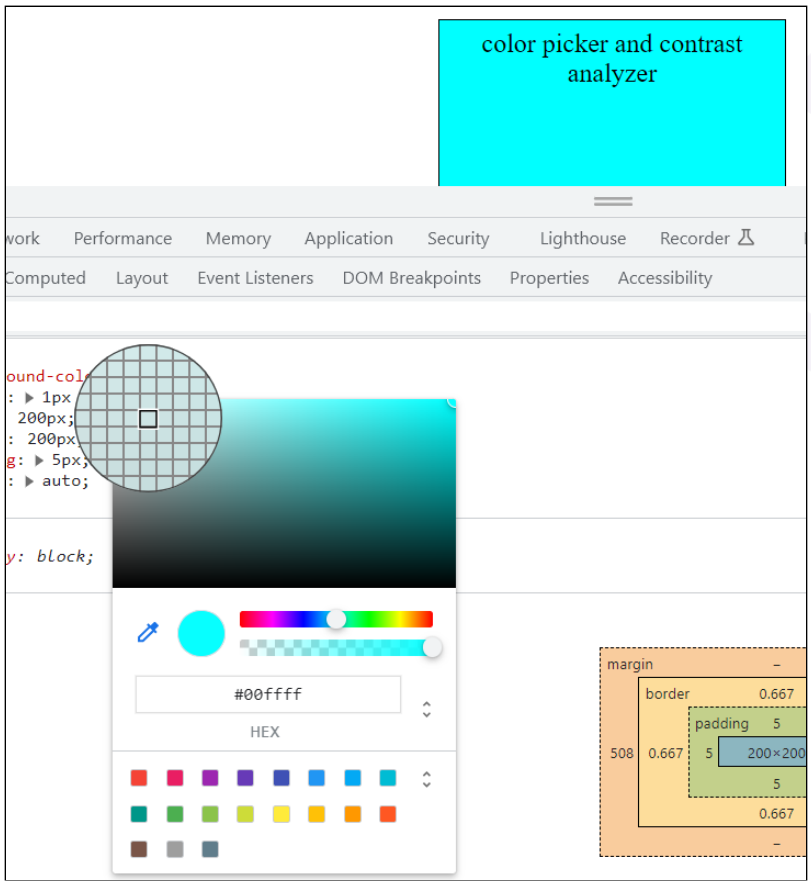
The pen-like icon circle with yellow colour is the colour picker.

Example of Color CSS picker (to change the background color of the div container to different color i.e. aqua blue to other different colors. Using color picker inside the Chrome dev tool)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <style>
      div {
        background-color: aqua;
        border: 1px solid black;
        width: 200px;
        height: 200px;
        padding: 5px;
        margin: auto;
      }
```

```
      div p {
        text-align: center;
        margin: auto;
      }
    </style>
  </head>
  <body>
    <div>
      <p>color picker and contrast analyzer</p>
    </div>
  </body>
</html>
```

This is how it works after selecting color picker any color the can be copied and used to change the background color temporary.

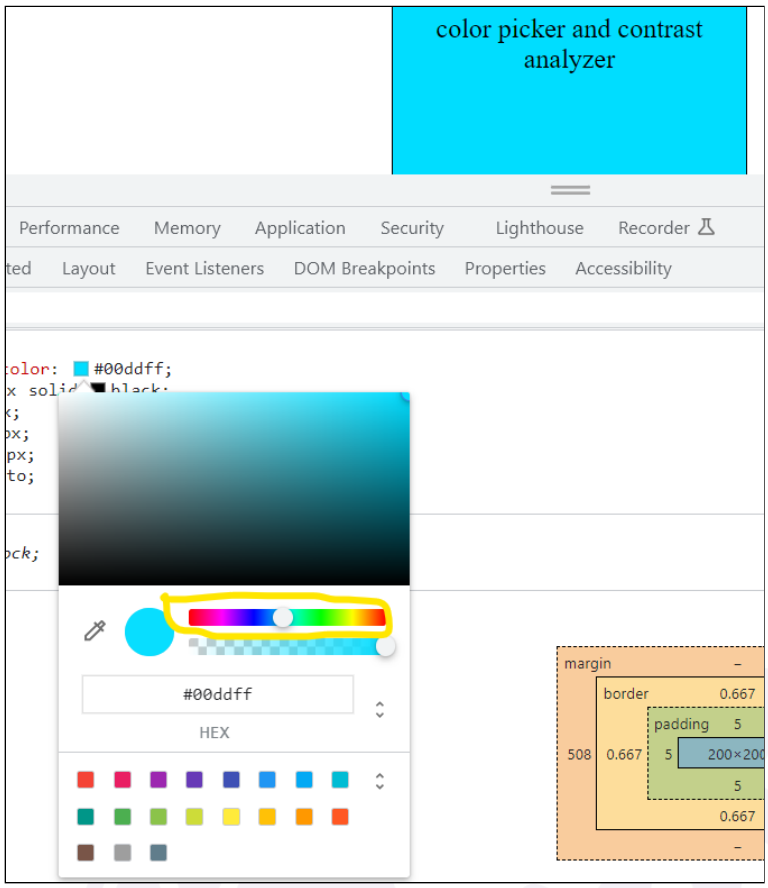


Color contrast –

A color contrast analyzer is a tool used to evaluate the contrast between two colors, typically a foreground color (such as text) and a background color. Contrast is essential for ensuring the readability and accessibility of content, especially for individuals with visual impairments. The Web Content Accessibility Guidelines (WCAG) set forth guidelines for minimum contrast ratios between text and background colors.

The Color contrast can be accessed the same way as the color picker.

Example - change the background color of the above example

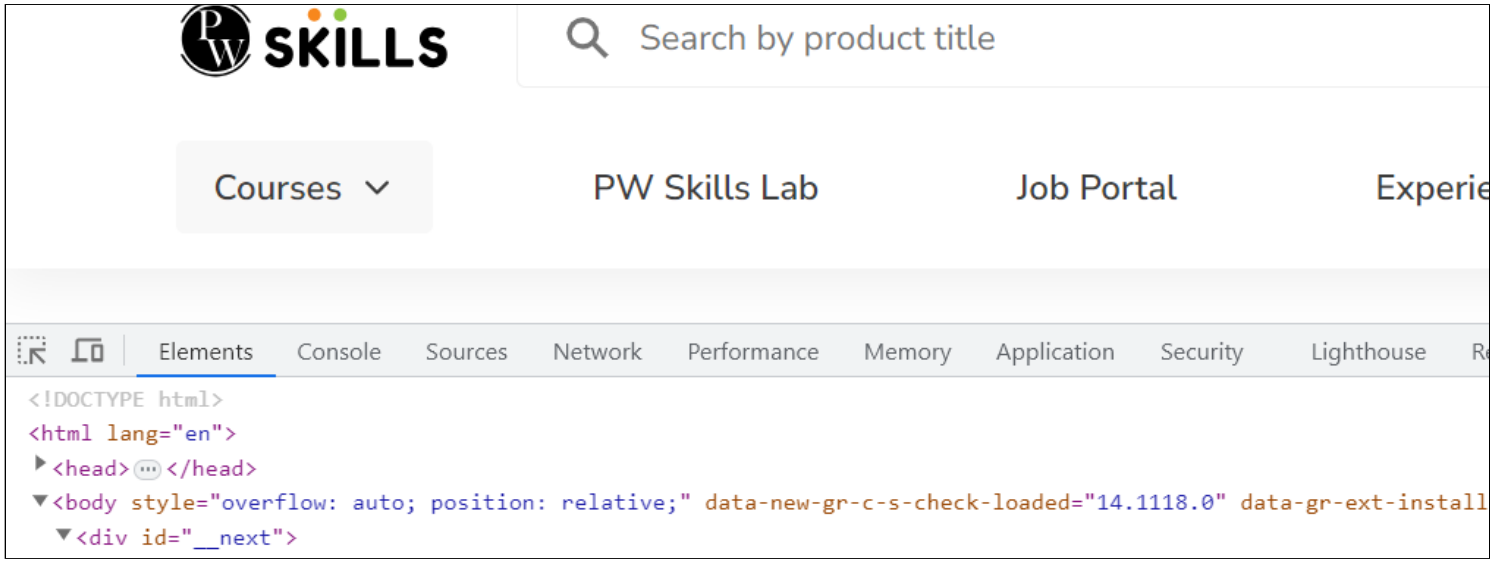


The yellow line circle, can be scrolled horizontally to change the color contrast, resulting in a change in background color temporary.

Working with media Query and responsive designs

Working with media queries and testing responsive designs in Chrome DevTools is a fundamental part of web development. Media queries allow you to apply specific CSS styles based on the characteristics of the user's device or screen, enabling you to create responsive designs that adapt to various screen sizes and orientations.
Here's how one can work with media queries and test responsive designs using Chrome DevTools.
We will use the pwkills.com website to demonstrate it.

Open Chrome Devtools
Visit the pwkills.com website, right-click and select the inspect option then chrome dev tool will open.
i.e

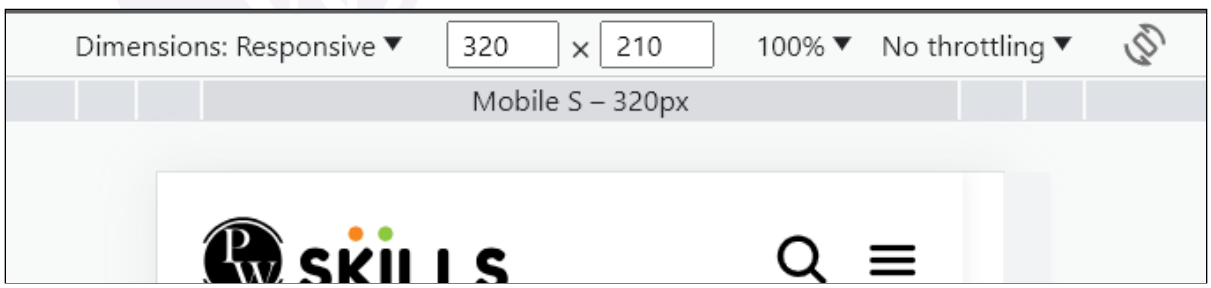


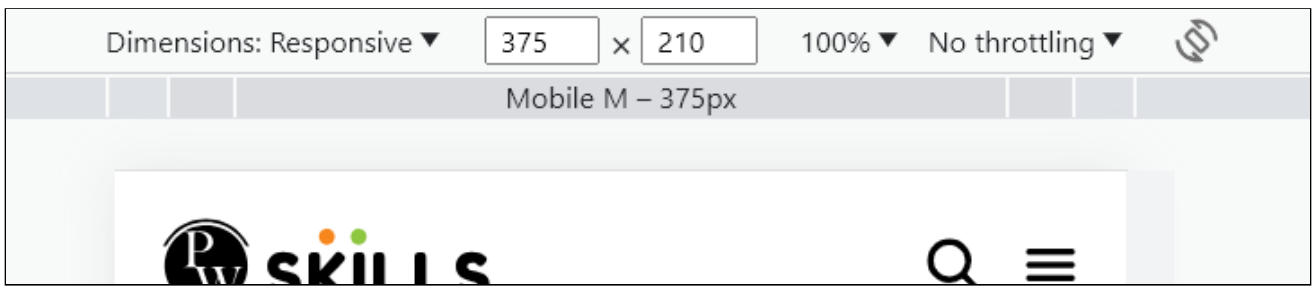
Toggle Device Toolbar

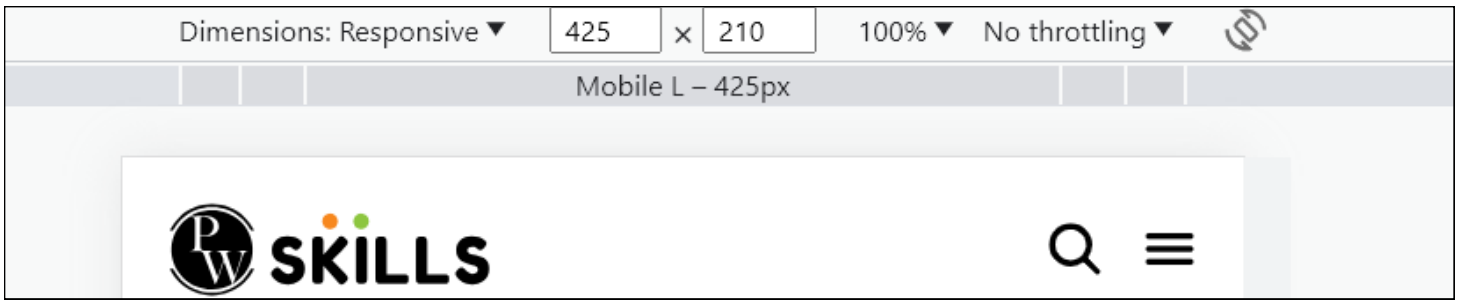
The Device Toolbar in Chrome DevTools allows you to simulate different devices and screen sizes. To toggle the Device Toolbar:

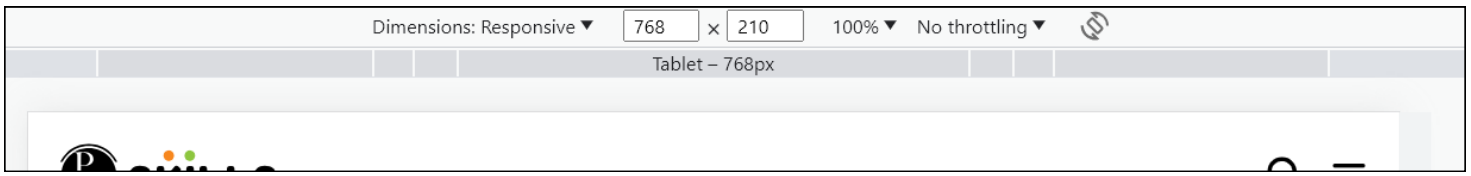
Select or click the yellow circle icon to activate the toggle device toolbar

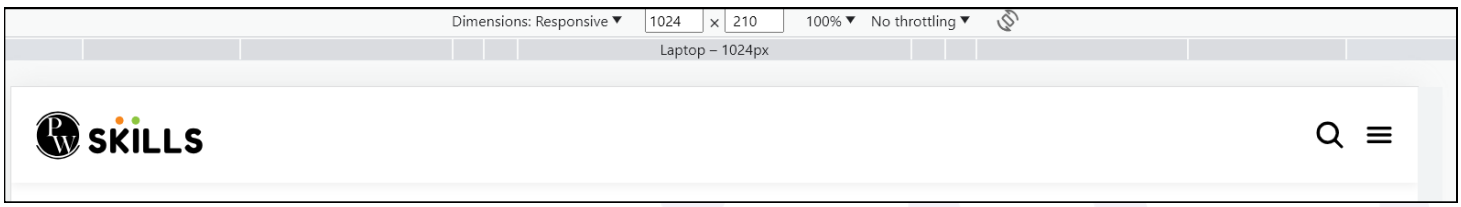
The different types of screen sizes available are -

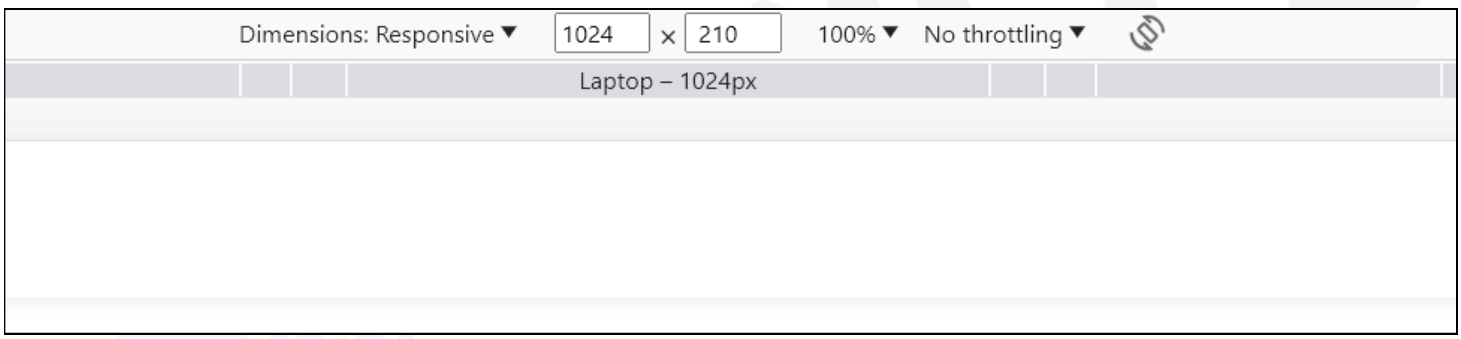










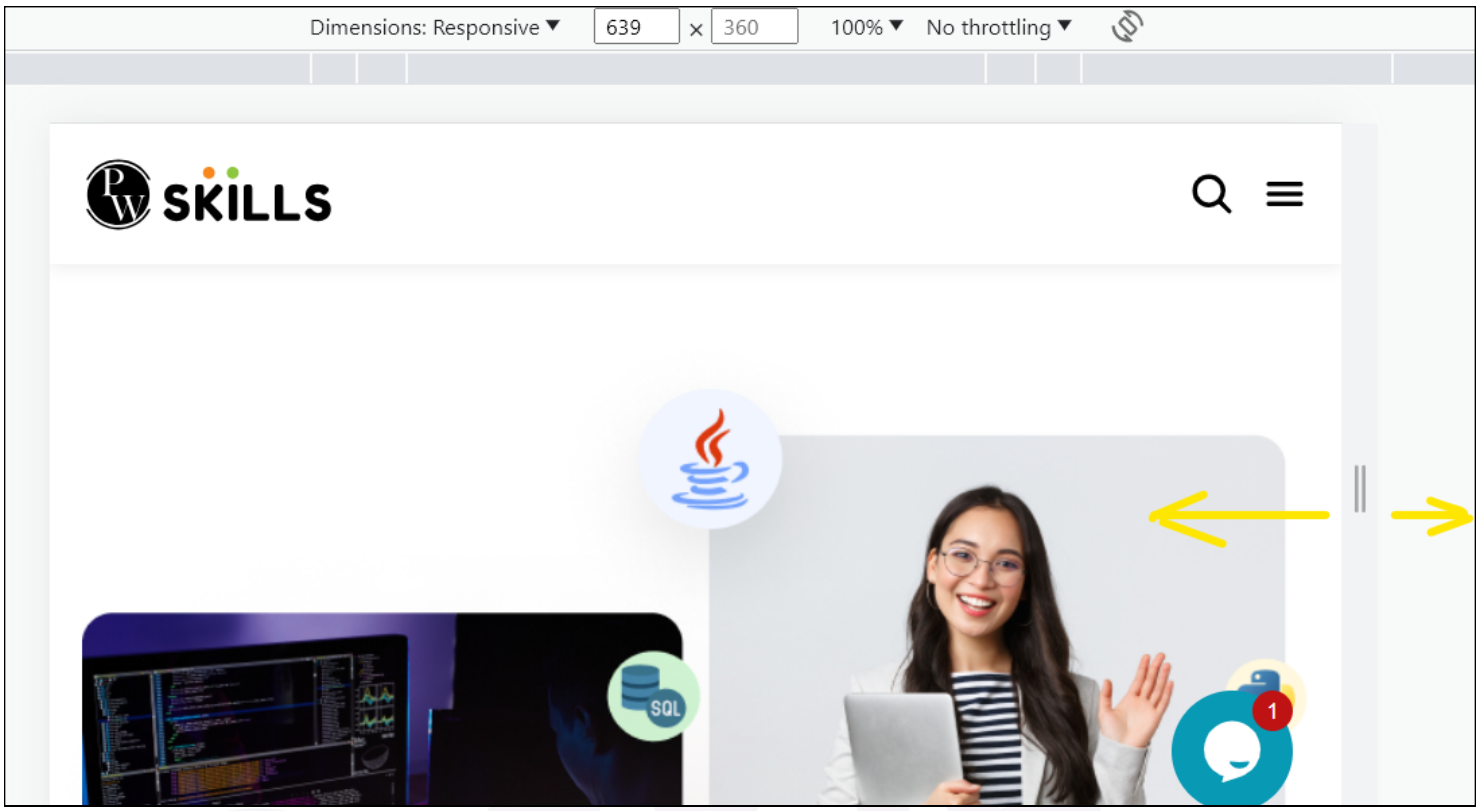


Choose a Device or Set Custom Dimensions
The device can be set to Custom Dimension just above the predefine device size.
Or by dragging horizontally the width size of the page.

Set Custom dimension

In here the different types of Dimension value can be added.

Drag the screen size horizontally -



Testing the CSS flexbox and grid layout

Testing CSS Flexbox and Grid layouts in Chrome DevTools is an excellent way to fine-tune your designs and understand how your layout is behaving on different screen sizes. Both Flexbox and Grid are powerful layout techniques that enable you to create flexible and responsive designs

Testing CSS Flexbox

index.html

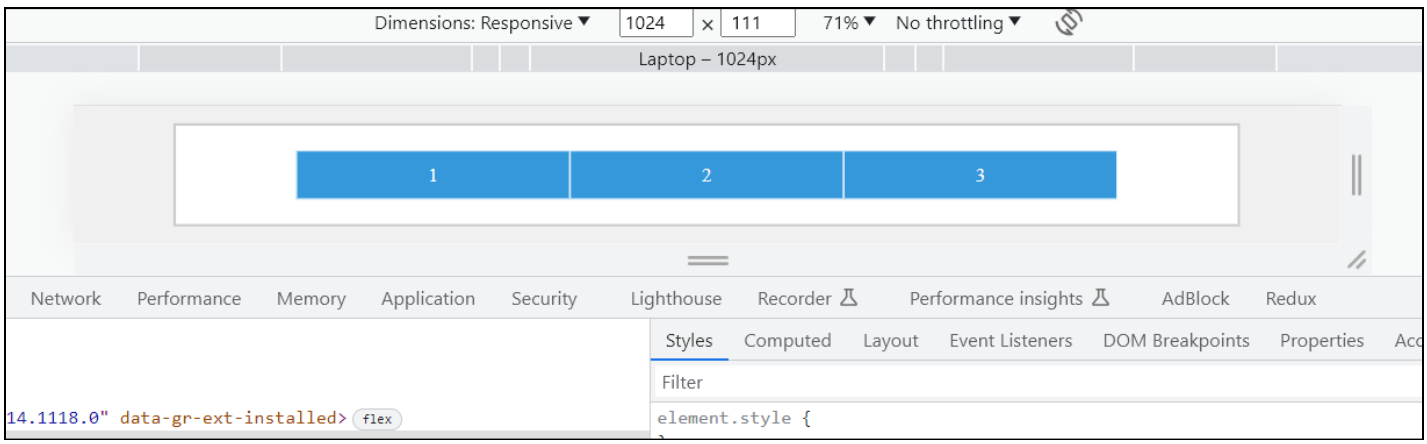
```
 <body>
      <div class="container">
        <div class="box">1</div>
        <div class="box">2</div>
        <div class="box">3</div></div>
</body>
```

style.css

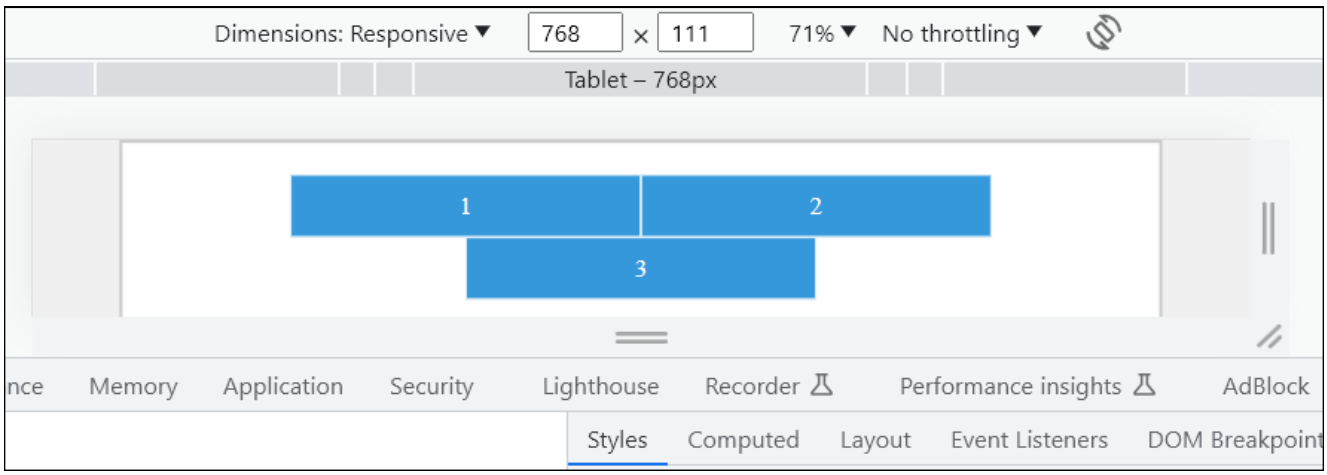
```
body {
        margin: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        min-height: 100vh;
        background-color: #f0f0f0;
      }
      .container {
        display: flex;
        flex-wrap: wrap;
        justify-content: center;
        width: 80%;
        background-color: #fff;
        padding: 20px;
        border: 2px solid #ccc;
      }

      .box {
        width: 200px;
        background-color: #3498db;
        color: #fff;
        text-align: center;
        border: 1px solid;
        padding: 10px; }
```

Browser output (Testing in flexbox in chrome dev tool) -

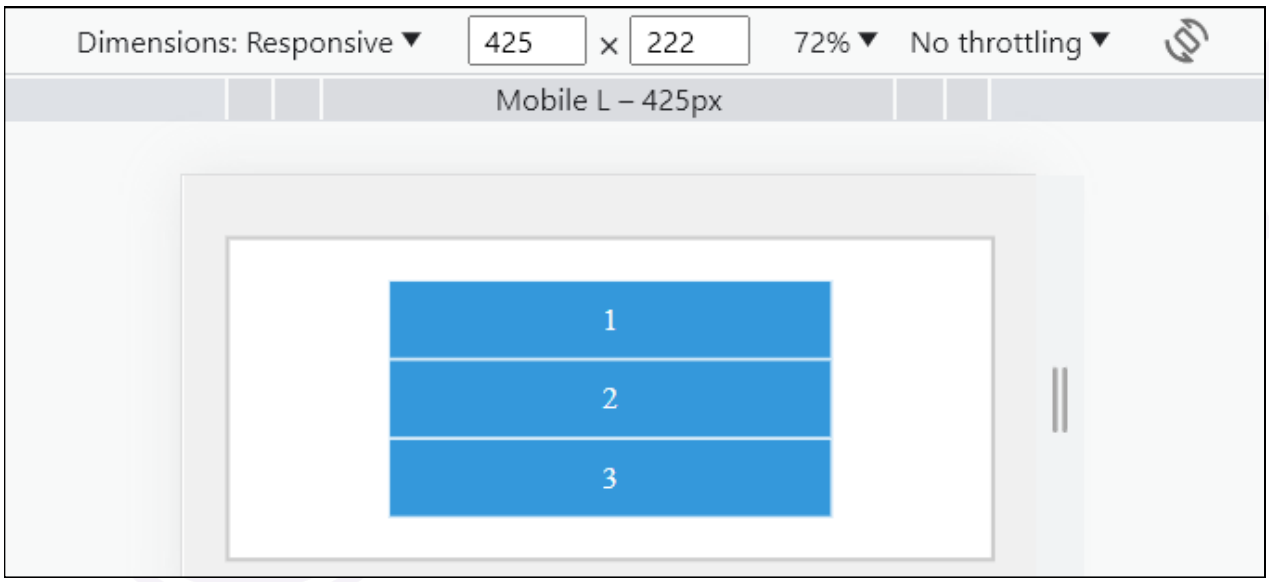
Screen size - laptop - 1024px, Here the flex items are in a single row



Tablet-size -768px, Here are the flex items in a two-row which stack on each other since the screen does not fit it in one row.



Mobile - 425px, Here the flex items are in one single column



The above example shows the responsiveness of the HTML elements using CSS flexbox.

Testing CSS Grid layout

index.html

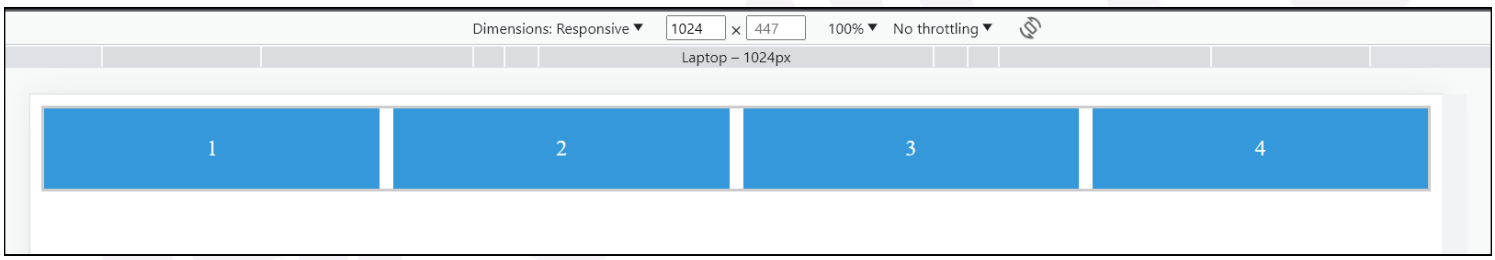
```
<body>
    <div class="grid-container">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
      <div class="item">4</div>
    </div>
  </body>
```

style.css

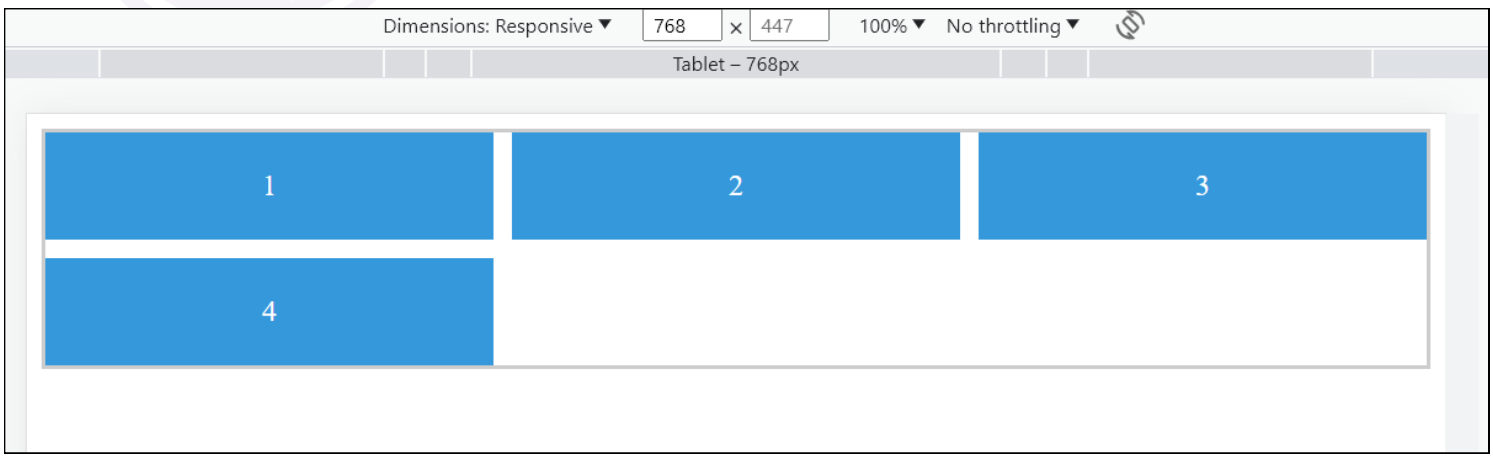
```
.grid-container {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    gap: 10px;
    background-color: #fff;
    border: 2px solid #ccc;
  }

  .item {
    background-color: #3498db;
    color: #fff;
    text-align: center;
    padding: 20px;
  }
```

Browser output (Testing in Grid layout in chrome dev tool) -

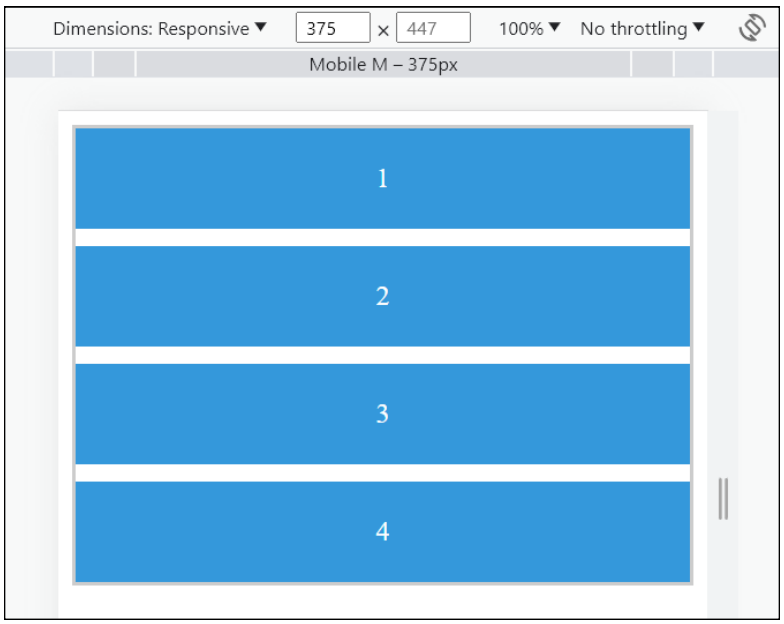
Screen size - laptop - 1024px, Here the Grid items are in a single row



Screen size - tablet-768px, Here the Grid items are in a two-row which stack on each other since the screen does not fit it in on row.



Mobile - 425px, Here the flex items are in one single column



The above example shows the responsiveness of the HTML elements using CSS Grid layout.