Module (CSS and CSS 3) -2

• 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 advantages of CSS –

- **CSS saves time** – You can write CSS 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.
- **Easy maintenance** – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **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.
- **Platform Independence** – The Script offer consistent platform independence and can support latest browsers as well.

• What are the 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.

- **What is the difference between CSS2 and CSS3?**

  - CSS3 is split into many various documents known as Modules. each module adds new capability or extends options outlined in CSS2 over conserving backward compatibility. Work on CSS3 started around the time of publication of the initial CSS2 recommendation.
  - The CSS3 version supports more browsers than CSS2.
  - CSS3 introduces several new selectors. Those new selectors square measure largely in an exceeding type of pseudo-elements and pseudo-categories.
  - The new addition of General relation Combinator will be wont to match relation parts of a given part through diacritic (~) combinator.
  - CSS3 introduces several properties attended with new values and units. It facilitates styling of backgrounds, borders, boxes, etc…, that permits the USA to stay most of the styling at intervals the computer network and HTML standards and our document, while not a necessity for all those proprietary third-party package packages.
  - New values and new units square measure introduced to support all those new properties. for example, Angle units deg, grad, rad, and switch or Time units s and ms.

- **Name a few CSS style components**

  CSS consists of two components:

  1. Properties: These are human-readable identifiers that indicate which stylistic features you want to modify. For example, font-size , width , background-color ,color, text align.
  2. Values: Each property is assigned a value. This value indicates how to style the property.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p >Me too!</p>
<p>And me!</p>

```

```
19.</body>
20.</html>
```

Every paragraph will be affected by the style.

Me too!

And me!

- **What do you understand by CSS opacity?**

The opacity property specifies the opacity/transparency of an element.

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  opacity: 0.6
  ;
}
</style>
</head>
<body>


<img src="https://media.istockphoto.com/id/1409131170/photo/diasy-border-on-blue-wood.jpg?s=2048x2048&w=is&k=20&c=Hxmp_19Ch-hQloizFDIjO2VWQW3vuDQa5ZKPnzcBcFc=" alt="Forest" width="170" height="100">
</body>
</html>
```

- **How can the background color of an element be changed?**

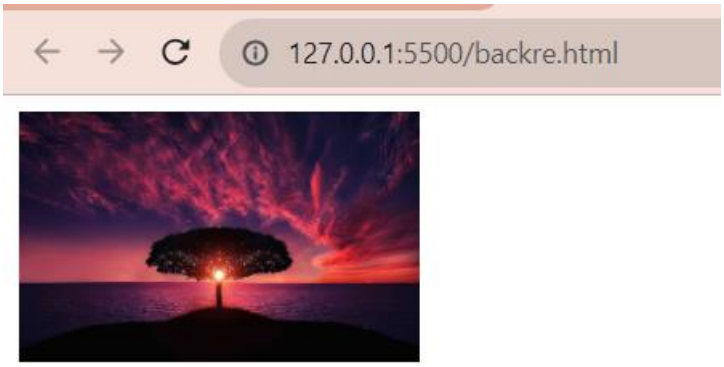
```
<!DOCTYPE html>
 <html lang="en">
 <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=h, initial-scale=1.0">
    <title>Document</title>
    <style>
        h1{
            background-color: aqua;
        }
    </style>
 </head>
 <body>
    <h1>this is first heading</h1>
 </body>
 </html>
```

this is first heading

- **How can image repetition of the backup be controlled?**

```
<!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>
```

```
        .d1{
            height: 200px;
            width: 200px;
            background-image:
url(https://cdn.pixabay.com/photo/2015/04/23/22/00/tree-736885_1280.jpg);
            background-size: contain;
            background-repeat: no-repeat;
        }

    </style>
</head>
<body>
    <div class="d1" ></div>
</body>
</html>
```



- **What is the use of the background-position property?**

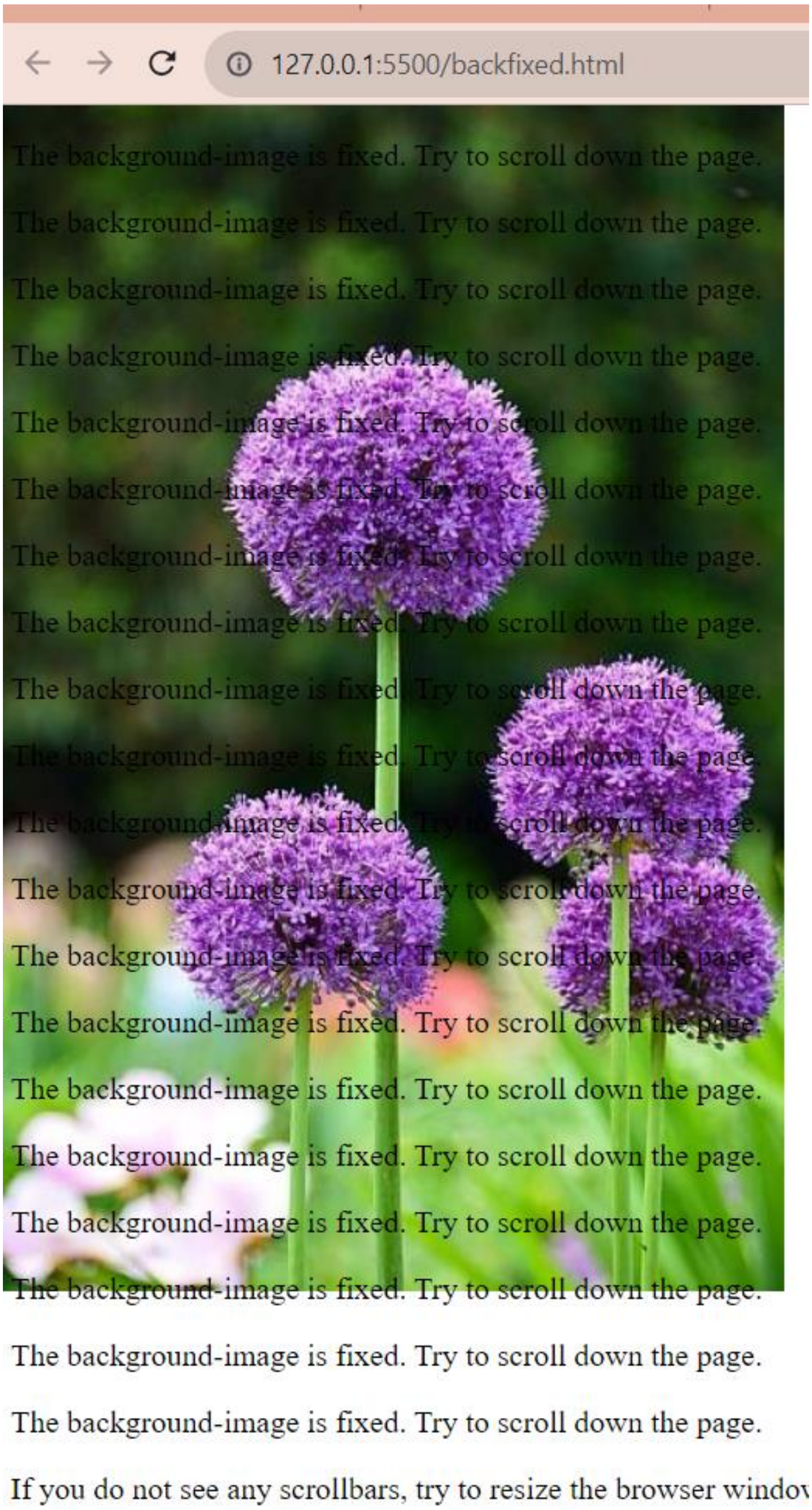
The `background-position` property sets the starting position of a background image.

- **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.

```
<!DOCTYPE html>
<html>
<head>
<style>
body  {
  background-image: url("https://hips.hearstapps.com/hmg-prod/images/types-of-
garden-flowers-purple-allium-1674847068.jpeg");
  background-repeat: no-repeat;
  background-attachment: fixed;
}
```

```
</style>
</head>
<body>

<h1>The background-attachment Property</h1>

<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>If you do not see any scrollbars, try to resize the browser window.</p>

</body>
</html>
```



• 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.

- **How to centre block elements using CSS1?**

Using margin

```
<!DOCTYPE html>
<html lang="en">

<head>
    <style>
        *{
            margin:0px;
            padding:0px;
            box-sizing: border-box;
        }

        body {
            background: brown;
        }

        #box {
            background: black;
            color:white;
            text-align: center;
            width:300px;
            margin-left: 500px;
            margin-top: 150px;
        }
    </style>
</head>

<body>
    <div id="box">
        <h3>this is a box</h3>
    </div>
</body>

</html>
```

- **How to maintain the CSS specifications**?

- **What are the ways to integrate CSS as a web page?**

CSS may be added to HTML in three different ways. To style a single HTML element on the page, use Inline CSS in a style attribute. By adding CSS to the head section of our HTML document, we can embed an internal stylesheet. We can also connect to an external stylesheet that separates our CSS from our HTML.

Inline css

```
<!DOCTYPE html>
<html>
<body>

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

Internal css

```
<!DOCTYPE html>
```

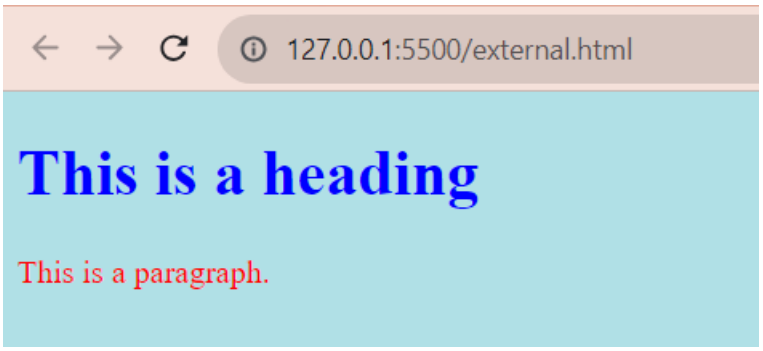
```html
<html>
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>
<body>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```

External CSS

```html
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style1.css">
</head>
<body>

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

</body>
</html>
```

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

```
}
```



• What is embedded style sheets?

Embedded style sheets is another name of inline css.

• 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.

• What is the meaning of the CSS selector?

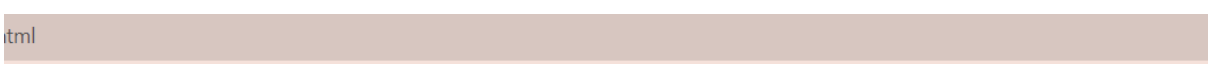
CSS selectors are used to "find" (or select) the HTML elements you want to style. We can divide CSS selectors into five categories: Simple selectors (select elements based on name, id, class) Combinator selectors (select elements based on a specific relationship between them)

```
<!DOCTYPE html>
<html>
```

```
<head>
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
</body>
</html>
```


Hello World!

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```



Red and center-aligned heading

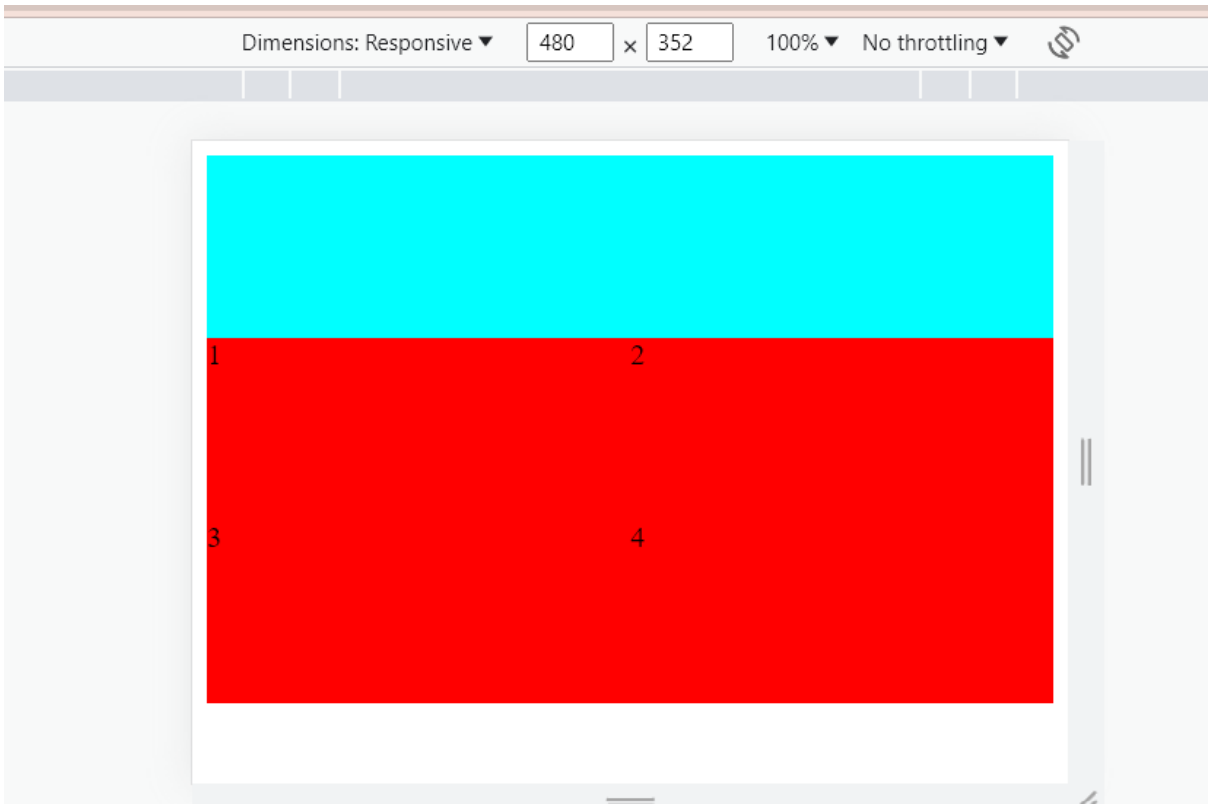
Red and center-aligned paragraph.

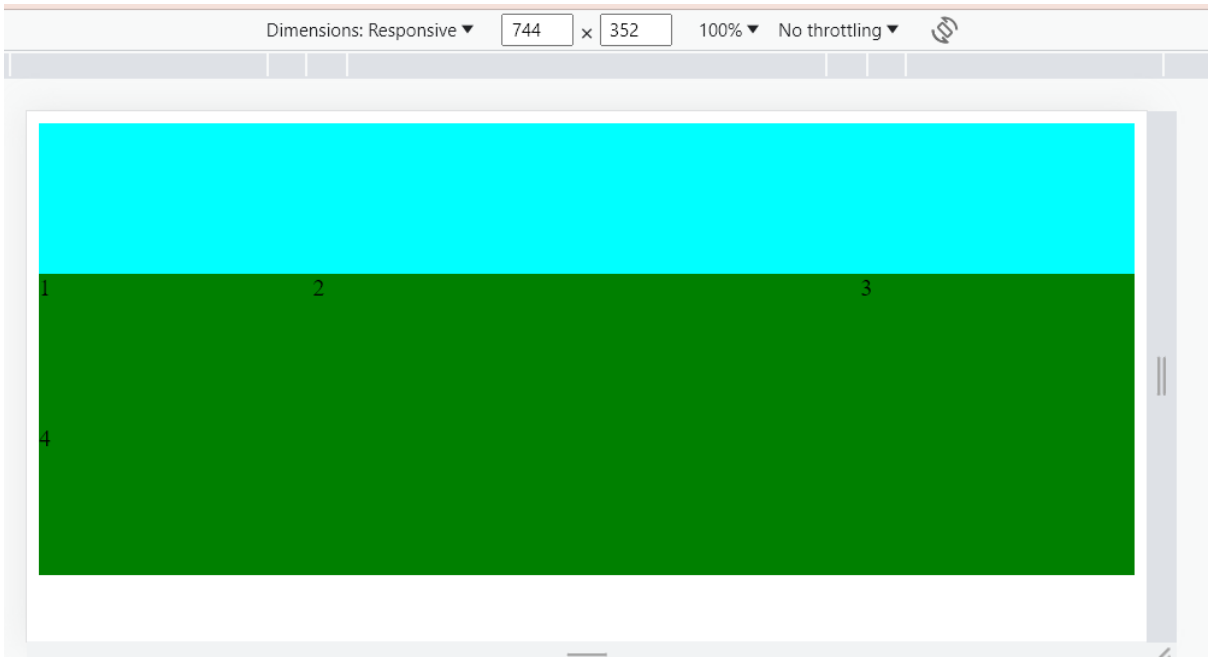
- **What are the media types allowed by CSS?**

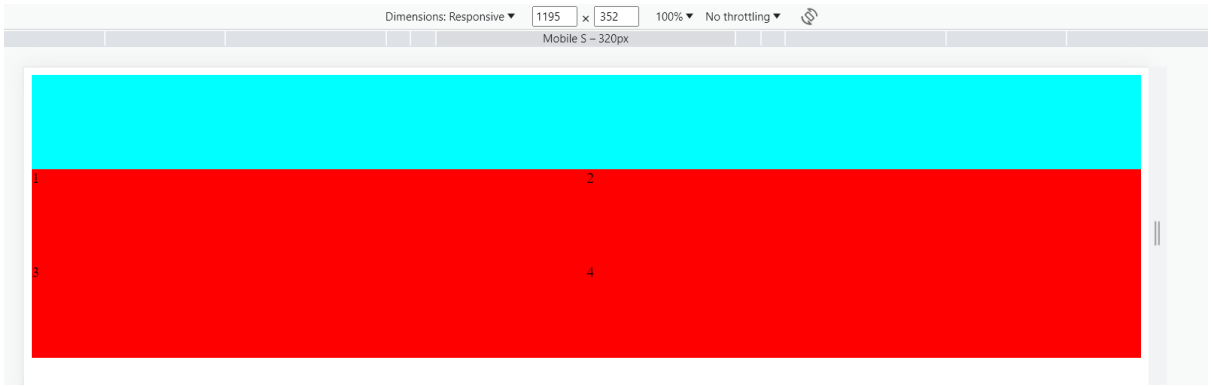
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        .header{
            height: 100px;
            background-color: aqua;
        }
        .main{
            display: flex;
            height: 200px;
            background-color: brown;
        }
        @media (min-width:320px) and (max-width:480px) {
            .main{
                display: flex;
                flex-wrap: wrap;
                background-color: yellow;
            }
            #two-1{
                width: 100%;
            }
            #two-2{
                width: 100%;
            }
            #two-3{
                width: 100%;
            }
            #two-4{
                width: 100%;
            }
        }
        @media  (max-width:1200px) {
            .main{
                display: flex;
                flex-wrap: wrap;
                background-color:red;
            }
            #two-1{
                width: 50%;
            }
            #two-2{
                width: 50%;
```

```
            }
            #two-3{
                width: 50%;
            }
            #two-4{
                width: 50%;
            }
        }
        @media (min-width:481px) and (max-width:768px) {
            .main{
                display: flex;
                flex-wrap: wrap;
                background-color:green;
            }
            #two-1{
                width: 25%;
            }
            #two-2{
                width: 50%;
            }
            #two-3{
                width: 25%;
            }
            #two-4{
                width: 100%;
            }
        }
        @media (min-width:769px) and (max-width:1024px){
            .main{
                display: flex;
                flex-wrap: wrap;
                background-color:blue
            }
            #two-1{
                width: 30%;
            }
            #two-2{
                width: 70%;
            }
            #two-3{
                width: 30%;
            }
            #two-4{
                width: 70%;
            }
        }
    </style>
</head>
```

```
<body>
    <div class="header">  </div>
        <div class="main">
            <div class="one" id="two-1">1</div>
            <div class="one" id="two-2">2</div>
            <div class="one" id="two-3">3</div>
            <div class="one" id="two-4">4</div>
        </div>

</body>
</html>
```







- **What is the rule set?**

A table of instructions used by a controlled interface to determine what data is allowable and how the data is handled between interconnected systems.

- **Create Layouts**

```
<!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>
        *{
            padding: 0%;
            margin: 0%;
            box-sizing: border-box;
        }
        div{
            display: grid;
            grid-template-columns: repeat(3);
            gap: 10px;
            background-color: aqua;

        }
        .one{
            border: 2px solid black;
            padding: 100px;
            background-color: antiquewhite;
        }
        #two-1{
            grid-row: 1/2;
            grid-column: 1/2;
        }
        #two-2{
            grid-row: 1/2;
            grid-column: 2/3;
        }
        #two-3{
            grid-row: 1/2;
            grid-column: 3/4;
        }
    </style>
</head>
<body>
    <div>
        <div class="one" id="two-1">1</div>
        <div class="one" id="two-2">2</div>
```

```
        <div class="one" id="two-3">3</div>
        <div class="one" id="two-4">4</div>
        <div class="one" id="two-5">5</div>
        <div class="one" id="two-6">6</div>

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