



An Article on Css

[By: Megha Shrivastava]

# Cascading Style Sheets

## Introduction

Cascading Style Sheet Generally Known As **CSS**.

As Name Identifies Used for Giving Style for Webpage .if we break the meaning of CSS. Then we can easily understand the use and need of CSS. Let's we do it below:

**Cascading**: An arrangement of a lightweight fabric in folds falling one over Another in Random or zigzag fashion.

**Style**: Method, or Approach, or Way, or Manner, or Fashion, or Technique, or Mode. Anyone you can take

**Sheet**: Piece, or Page, or Leaf, or Slip, or Pane. Anyone you can take Now together and technically we can say CSS is basically a sheet which gives style to webpage in a cascading manner.

## History

Creator of Csst is Hakon Wium Lie (working for Opera Software as Chief Technology Officer) and Bert Bos (working for The World Wide Web Consortium on style sheets and math.) authored the original CSS1 Specification.

**Versions of Csst**: we have basically three versions of Csst. Each Version of CSS

**CSS 1**: An official W3C Recommendation is CSS I, published in December 1996. CSS1 is a simple style sheet mechanism that allows authors and readers to attach style colors and spacing to HTML documents. The CSS1 language is just like a simple English language.

One of the fundamental features of CSS is that style sheets cascade; authors can attach a preferred style sheet, while the reader may have a personal style sheet to adjust for human or technological handicaps.

## Features Supported by CSS1 for:

- Font
- Colors
- Spacing between words, letters, and lines of text.
- Alignments of text, images, tables and other elements.
- Margin, border, padding, and positioning for most elements.
- Unique identification and generic classification of groups of attributes.

## CSS 2:

A superset of CSS1, CSS 2 was published as a Recommendation in May 1998, included number of new capabilities.

### Features Supported by CSS2:

- **Absolute Relative Fixed positioning of elements:** it allows Font selection, Tables (allow you to define any element as a table element (and all the related table elements)), Bidirectional text, Cursors (now you can define how you want your cursor to respond to various actions.)
- **The concept of media types:** allow you to specify different style rules depending upon how your document is going to be displayed. There are many different types you can specify, including: aural, Braille, handheld, screen, print, and TV
- **Paging:** it is possible to define how pages should be displayed or printed. This means that you can specify the size of the page to be printed, add things like crop marks and register marks, or how the page should layout on double- and single-sided printings.
- **Aural:** if a customer comes to your Web page with a screen reader that is CSS2 enabled, you can define how your page will sound. And this isn't just useful for blind customers. And many more features.

## CSS 3

A superset of CSS1 and CSS 2, included number of new capabilities now under construction.

## Browser Support

To be master of the full power of Cascading Style Sheets, you need to understand the CSS—browser support. No other Web-related technology has been more limited by poor browser implementation than Cascading Style Sheets.

For understanding why browser is not compatible with Ccss first we take a look how browser deal with Ccss document when browser encounters in a Ccss code it has three choice

- if it is programmed to understand the functionality, it will work as written

- if it is not programmed to understand the functionality, it ignore that code and no response
- This option is the least desirable the browser can do wrong things this is the main cause of our problems. But page will still be usable and the content accessible. Because we've separated presentation from content, our content should be able to seen but not as stylish way as we designed.

Now we take a look in the resigns on incompatibility of web-browsers with Css the major issues are listed below:

- Launched before Css: some older browser which are launched before Css launched so they totally don't understand the functionality that's why they totally ignore the Css code .e.g. Netscape 3
- Some browsers are support only subset of Css means some properties of Css. Example of this kind of browser are Microsoft's WebTV and EmacSpeak.
- The third category of browser are worst category .the basic difference between older browser and this kind of browser is that the older browser doesn't try to go through in Css they simply avoid the Css but these kind of browser tried and fail horribly. I think this the most dangerous category of browsers. Example of this category is Internet Explorer 3.
- At the time there are no browsers that completely support all of the Cascading Style Sheets Level 2 specification. But we are improving day by day hopefully in ear future we'll have browsers that completely support Css and we can enjoy working with Css.

## Advantages of using CSS

- Reusability: for using Css there is no need to type code again and again, write that in a class and applied that class for particular element .u can also inherit the class with in another class. It time saving also.
- Faster Download: less code is the resign for faster downloading.
- Maintenance: for changing the style of a document, u just changes it in Css file in one place .there is no need to change it for each element.

## Ways to applying CSS

We have four ways to put CSS in our document

- You can create a separate CSS file with extension .css. and add this file to your HTML page
- `<link rel="stylesheet" href="xxx.css" type="text/css">` where xxx.css is your file name.
- You can also apply CSS properties with style tag on a particular element
  - `<p style="color: blue">CSS Style on a paragraph. </p>`
- You can also embed CSS style into an HTML document using the 'style' tag
  - `<style type="text/css" >`
    - `P {color: green ;}`
    - `h1 {color: red ;}`
  - `</style>`
- You can also import another CSS file in the `<head>...</head>` tags of HTML documents that you want to add
  - `@import "xxx.css";`
  - `@import URL ("xxx.css");`

**After this brief introduction of CSS, I just wanted to share some problem which I faced while working in a content management project and I got very simple solution by using CSS.**

- **Problem I:** As I wrote above while working on content management project which was under construction so most of the time we have to design new pages on the site with having same look and field with different functionality so. It's very different to edit page for designing issues.

Using Web standards also makes it extremely easy to maintain visual consistency throughout the site. Since pages use the same CSS document for their layout, they are all formatted the same. And I can easily maintain some standard CSS files having global standard of the site designs, and individual CSS file for each web page. On which we can easily inherit and modify the CSS issues without affecting the global CSS. And easily use global issues with local modifications. Use of CSS here gives me freedom of design and functionality with consistency, clarity in less timing.

**Problem II:** The second major problem I faced in this project while working with tables. Problem I faced using <table> tag of HTML is below:

- Mixes presentational data in with your content.

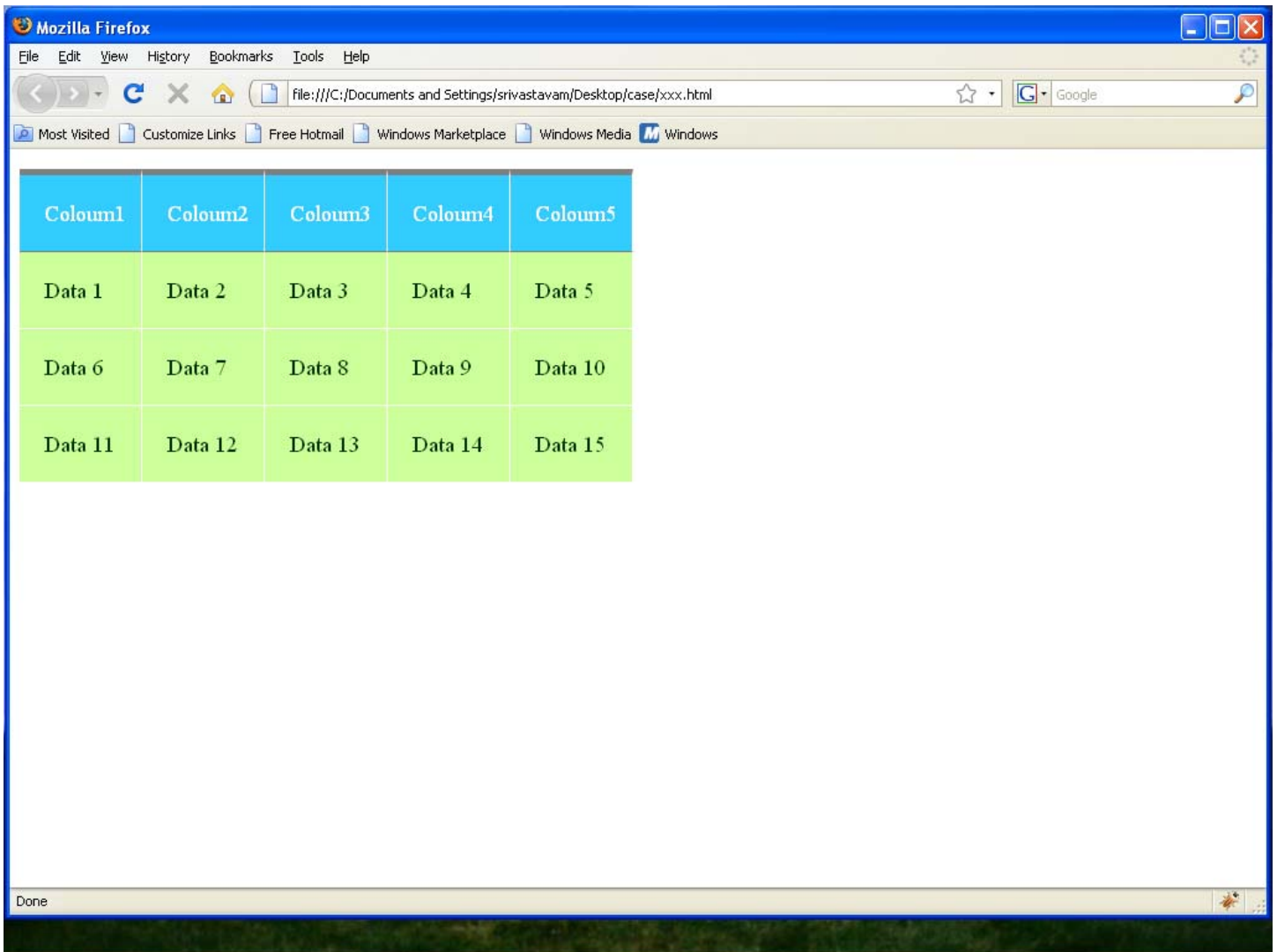
This makes the file sizes of your pages **unnecessarily large**, as users must download this presentational data for *each* page they visit.

- Bandwidth isn't free.

This makes redesigns of existing sites and content extremely **labor intensive** (and **expensive**).

- It also makes it **extremely hard (and expensive) to maintain visual consistency** throughout a site.
- Table-based pages are also **much less accessible** to users with disabilities and viewers using cell phones and PDAs to access the Web.

Creating table through Css <ul> tag short out all the above problems one by one and make project so much easy to design and flexible to change. Here I paste screen shot of the table which I used.



## HTML for this page is:

1. <html>
2. <head>
  - o <link rel="stylesheet" type="text/Css" href="xxx.css" />
3. </head>
4. <body>
  - o <div id="tabs" >

```
5.         <dl id="tab-links">
6.             <dt>Coloum1</dt>
7.             <dt>Coloum2</dt>
8.             <dt>Coloum3</dt>
9.             <dt>Coloum4</dt>
10.            <dt>Coloum5</dt>
11.        </dl><br>
12.        ■ <dl id="tab-links1">
13.            <dt>Data 1</dt>
14.            <dt>Data 2</dt>
15.            <dt>Data 3</dt>
16.            <dt>Data 4</dt>
17.            <dt>Data 5</dt>
18.        ■ </dl>
19.        ■ <dl id="tab-links1">
20.            <dt>Data 6</dt>
21.            <dt>Data 7</dt>
22.            <dt>Data 8</dt>
23.            <dt>Data 9</dt>
24.            <dt>Data 10</dt>
25.        </dl>
26.        ■ <dl id="tab-links1">
27.            <dt>Data 11</dt>
28.            <dt>Data 12</dt>
29.            <dt>Data 13</dt>
30.            <dt>Data 14</dt>
31.            <dt>Data 15</dt>
32.        </dl>
33.    </div>
```

</body>

</html>

### CSS for this page is:

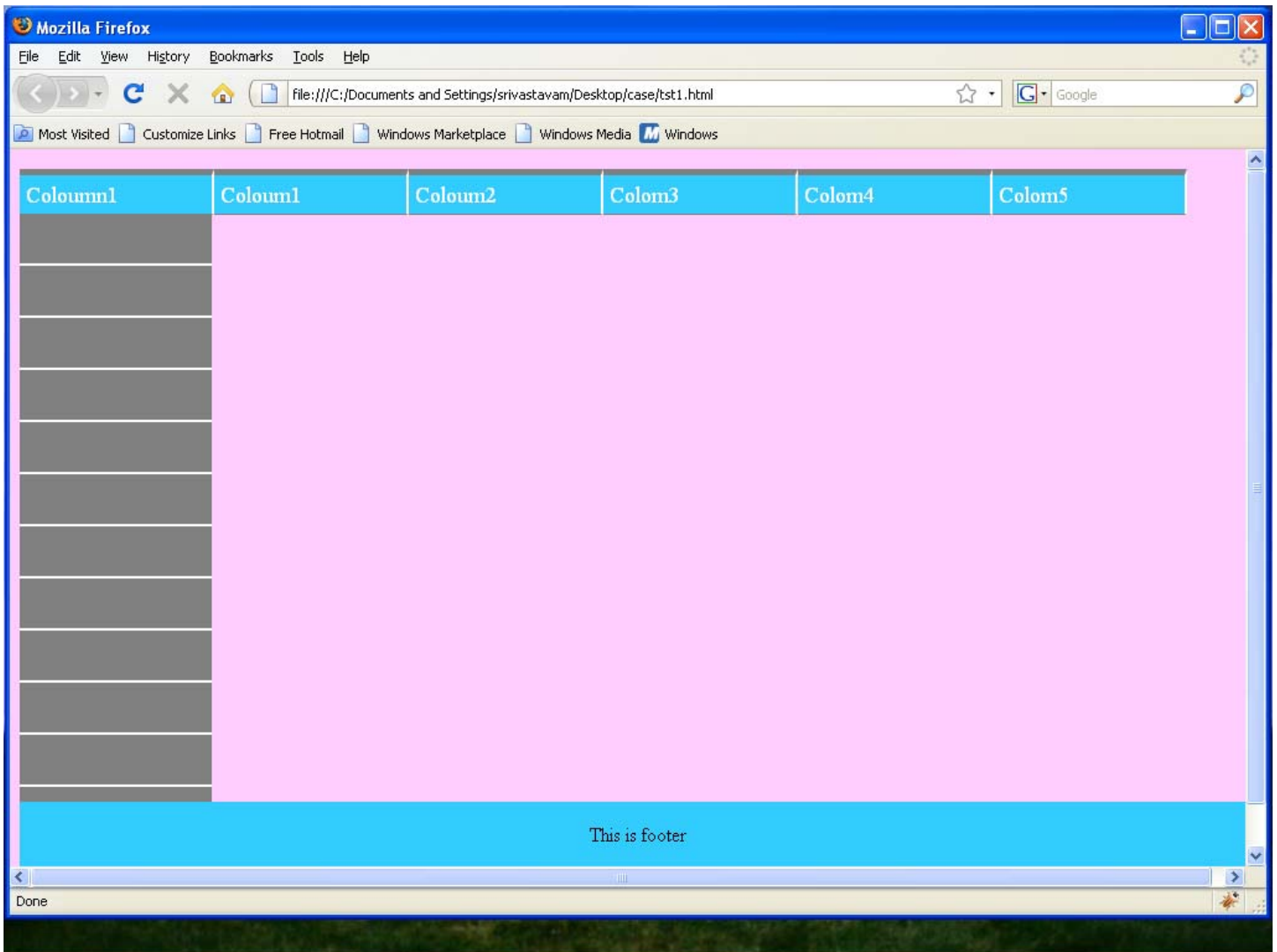
1. #tab-links dt
2. {
3. float: left;
4. padding: 20px;
5. color: white;

```
6.     background-color: #33CCFF;
7.     border-right: 1px solid white;
8.     border-bottom: 1px solid gray;
9.     border-top: 5px solid gray;
10.    font-size: 18px;
11.    font-color: white;
12.    width : 58px;
    }

13.    #tab-links1 dt
14.    {
15.        float: left;
16.        padding: 20px;
17.        color: #003300;
18.        background-color: #CCFF99;
19.        border-right: 1px solid white;
20.        border-bottom: 1px solid white;
21.        font-size: 18px;
22.        font-color: blue;
23.        width : 58px;
24.    }

25.    #tabs
26.    {
27.        width: 500px;
28.        align: center;
29.    }
```

- Another web page which I designed using Css



## HTML for this page is:

1. <html>
  - o <head>
    - <link rel="stylesheet" type="text/css"href="tst1.css" />
  - o </head>
  - o <body bgcolor = "#FFCCFF">
    - <div id="tabs" >
2. <dl id="tab-links">
3. <dt >Coloumn1</dt>

```
4.          <dt>Coloum1</dt>
5.          <dt>Coloum2</dt>
6.          <dt>Colom3</dt>
7.          <dt>Colom4</dt>
8.          <dt>Colom5</dt>
9.          </dl><br>
           ▪ </div>
           ▪ <div id = "tab-left" >
             • <dt></dt>
10.         <dt></dt>
11.         <dt></dt>
12.         <dt></dt>
13.         <dt></dt>
14.         <dt></dt>
15.         <dt></dt>
16.         <dt></dt>
17.         <dt></dt>
18.         <dt></dt>
19.         <dt></dt>
20.         <dt></dt>
21.         <dt></dt>
22.         <dt></dt>
23.         <dt></dt>
           ▪ </div>
24.
           ▪ <div id = "footer" >
             • <P> This is footer</p>
           ▪ </div>

</body>
```

## CSS for this page is

#tab-links dt

{

1. float: left;
2. padding: 5px;
3. color: white;
4. background-color: #33CCFF;

```
5. border-right: 2px solid white;
6. border-bottom: 1px solid gray;
7. border-top: 5px solid gray;
8. font-size: 18px;
9. font-color: white;
10. width : 145px;

}
```

#tab-left dt

```
{

11. padding: 20px;
12. color: white;
13. background-color: gray;
14. border-bottom: 2px solid white;
15. font-size: 18px;
16. font-color: white;
17. width : 115px;
18. margin-left: 50cm
19. height : 107cm;

}
```

#footer

```
{

20. bottom: 0;
21. position: absolute;
22. background-color: #33CCFF;
23. color: #000;
24. width: 100%;
25. text-align: center;

}
```