# Cross-Browser Compatibility Issues

Very few people realize that besides Google Chrome, there are many other web browsers like Internet Explorer, Mozilla Firefox, Safari etc. Moreover, each browser has different versions creating innumerable variations of browsers that people could be using to visit your website.

Just because your website looks good on the browser you use, it does not necessarily mean that it seems the same across multiple browsers, various operating systems and mobile devices used to visit your site. Inconsistent views across multiple browsers can create cross-browser compatibility issues.

Different web browsers parse website code differently. It is vital to check your website for any cross-browser compatibility issues as a design final step. Cross-browser compatibility issues are caused by errors within the website code. This means major browsers like Google Chrome reads and displays your website differently than Internet Explorer or Mozilla Firefox.

You must fix compatibility issues or risk not being able to reach potential customers. However, if you are not sure what to look for, it will be hard to fix the problem. A website code validator is an excellent place to start if you are developing your own website from scratch. If you are using a theme-based website maker like Squarespace or WordPress, it contains PHP and can be difficult to diagnose if you are not experienced.

**What is Cross Browser Compatibility?**

**Making your website behave consistently across browsers is called cross-browser compatibility.** Whether it is the look and feel or its behavior, a site can be created to function similarly across browsers with cross-browser compatibility and is an important step in the web design process. Many web designers tend to bypass this step due to time constraints.

Here is an example:
You have a form on your website that is crucial to lead generation. When you tested the site before launching it, the form worked perfectly on your browser. Let's say 20% of the visitors to your website use a particular browser on which the form does not work; it either looks misaligned and bad or clicking the submit button does not result in an actual submission of the form. Imagine the impression this 20% of visitors will carry about your website and business. Also, you must consider the loss of potential leads from this 20% of visitors.

**Four major things that contribute to cross-browser compatibility issues:**
A) While every browser implements W3C standards, they do not necessarily do it the same way. For instance, each browser treats HTML, CSS, and JavaScript differently.

B) Besides the basic implementation, each browser handles enhancements or deviations from W3C standards in their own way.

C) The operating system on which the browser runs handles graphics and visual elements in its own way creating differences.

D) Add to all of that the different screen resolutions, both desktop and mobile devices, and you have a number of variables that can impact not just the look of your website on the browser but how it behaves.

**Tips to Avoid Cross Browser Compatibility Issues**

Depending on the complexity of your website, it can sometimes be almost impossible to achieve 100% compatibility, but a web designer or web developer can make sure the site looks and behaves consistently across browsers by following these steps:

**Understand Browser Differences**: The first thing a web designer needs to know before designing a website is the differences between browsers and how they handle elements. This awareness alone will go a long way in ensuring the design does not lead to cross-browser compatibility issues that cannot be corrected.

**W3C Compliance:** After the website is coded, the web designer needs to try and achieve W3C compliance. This can be done at https://validator.w3.org/. Achieving W3C compliance may not be easy, mainly if your site uses features that are considered an enhancement over W3C standards. However, the closer you come to a W3C compliant code, the lesser the chances of the site differing in performance and look in different browsers.

**Cross-Browser Testing:** Test your website in as many different browsers as possible before releasing it. There are many free and paid cross-browser testing tools available to check all versions of multiple browsers.

**Understanding the Audience:** Make sure you understand the browsers and devices you're targeting. Is your page mostly for mobile users? Are you working within a company that still uses legacy browsers? Test accordingly.

**Use Fewer CSS Features:** Keeping the website code simple will reduce the amount of possible cross-browser issues. Not all websites need to be dazzling; simpler is better sometimes.

**Style Sheets:** Using Cascading Style Sheets (CSS) allows you to manage the look and feel of elements on your web pages. However, not all style sheets may be compatible with all versions of every browser. Web designers need to be aware of this and select styles that will have the least impact on cross-browser compatibility.

**Test Difficult Web Browsers First:** The most difficult browsers like Internet Explorer or Firefox should be tested first so that cross-browser compatibility issues show up quickly. If you test the easier, simpler web browsers first, then your errors may not show up until the end.

**Use a Framework:** If you use a well-established framework, whether for styling (eg Bootstrap) or a JavaScript framework (such as Angular or React), generally this means someone else has taken care of a lot of the cross-browser compatibility work for you.

**Flash:** It is essential for web developers to know that browsers designed for the visually impaired and most mobile browsers do not support flash. While Flash can add to the visual appeal of the site, it can cause cross-browser compatibility issues, especially on hand-held devices. Flash is also not the most search engine friendly of web technologies.

**Image Links:** When you link images unless you use an ALT attribute for the images, they will pose the same problems as Flash on mobile browsers and browsers for the visually impaired.

In spite of all these efforts, a website may not be 100% cross-browser compatible. However, minor differences in the look of a site do less damage than elements that do not work on certain browsers.

# Introduction to HTML

HTML stands for Hyper Text Markup Language
HTML describes the structure of Web pages using markup
HTML elements are the building blocks of HTML pages
HTML elements are represented by tags
Browsers do not display the HTML tags, but use them to render the content of the page

## HTML Versions

Since the early days of the web, there have been many versions of HTML:

| VERSION | YEAR |
|---|---|
| HTML | 1991 |
| HTML 2.0 | 1995 |
| HTML 3.2 | 1997 |
| HTML 4.01 | 1999 |
| XHTML | 2000 |
| HTML5 | 2014 |

# First Example

```
<html>
 <head>
      <title>First HTML Program</title>
 </head>
 <body>
      <h1>This is the Main Heading</h1>
      <p>This text might be an introduction to the rest of
           the page. And if the page is a long one it might
            be split up into several sub-headings.</p>
      <h2>This is a Sub-Heading</h2>
      <p>Many long articles have sub-headings so to help
           you follow the structure of what is being written.
           There may even be sub-headings (or lower-level headings).</p>
      <h2>Another Sub-Heading</h2>
      <p>Here you can see another sub-heading.</p>
 </body>
</html>
```

An HTML **element** is usually made up of two **tags** - an opening tag (start tag) and a closing tag (end tag) – with content inserted in between. (The closing tag has an extra forward slash in it.)

```
<tagname>Content goes here...</tagname>
```

The HTML element is everything from the start tag to the end tag. Each HTML element tells the browser something about the information that sits between its opening and closing tags.

The opening <html> tag indicates that anything between it and a closing </html> tag is HTML code.

You will often see a <head> element which contains contains meta information i.e. information about the page (rather than information that is shown within the main part of the browser window).

You will usually find a <title> element inside the <head> element. The contents of the <title> element are either shown in the top of the browser, above where you usually type in the URL of the page you want to visit, or on the tab for that page (if your browser uses tabs to allow you to view multiple pages at the same time).

The <body> element defines the document body. The <body> tag indicates that anything between it and the closing </body> tag should be shown inside the main browser window.

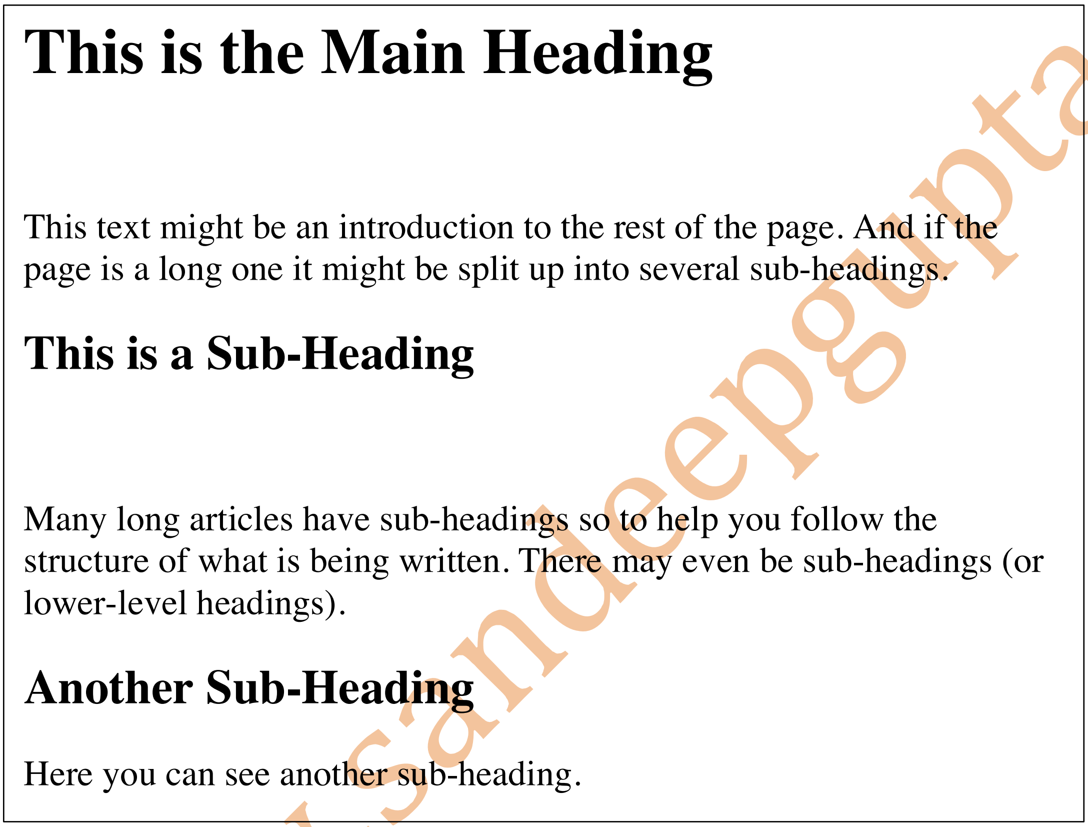
HTML headings are defined with the <h1> to <h6> tags. HTML has six "levels" of headings. <h1> is used for main headings <h2> is used for subheadings. If there are further sections under the subheadings then the <h3> element is used, and so on...

Browsers display the contents of headings at different sizes. The contents of an <h1> element is the largest, and the contents of an <h6> element is the smallest. The exact size at which each browser shows the headings can vary slightly. Users can also adjust the size of text in their browser. You will see how to control the size of text, its color, and the fonts used when we come to look at CSS. In above example, words between <h1> and </h1> are part of main heading.

HTML paragraphs are defined with the <p> tag. A paragraph of text appears between these <p> and </p> tags. By default, a browser will show each paragraph on a new line with some space between it and any subsequent paragraphs.

HTML tags are not case sensitive: <P> means the same as <p>.
The HTML5 standard does not require lowercase tags, but W3C recommends lowercase in HTML, and demands lowercase for stricter document types like XHTML.

**Output:**



# This is the Main Heading

This text might be an introduction to the rest of the page. And if the page is a long one it might be split up into several sub-headings.

## This is a Sub-Heading

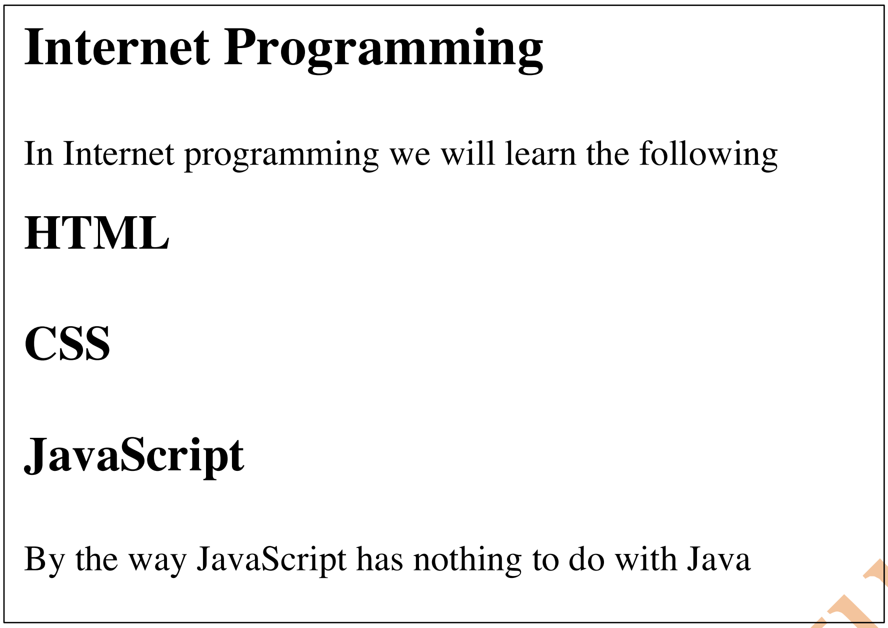
Many long articles have sub-headings so to help you follow the structure of what is being written. There may even be sub-headings (or lower-level headings).

## Another Sub-Heading

Here you can see another sub-heading.

***Do it youself***

**Write an html program to create the following:**



# Internet Programming

In Internet programming we will learn the following

## HTML

## CSS

## JavaScript

By the way JavaScript has nothing to do with Java

**Creating an HTML file**

**W3C Validators**

# HTML Formatting

HTML defines special elements for formatting text.
For example, elements like <b> for bold and <i> for italic text.

| | HTML Element | Purpose |
|---|---|---|
| 1 | <b> | Makes text **bold** |
| 2 | <i> | Makes text *italic* |
| 3 | <strong> | The use of the <strong> element indicates that its content has strong importance. The words contained in this element might be said with strong emphasis by screen readers. By default, browsers will show the contents of a <strong> element in bold. |
| 4 | <em> | The <em> element indicates emphasis that subtly changes the meaning of a sentence. By default browsers will show the contents of an <em> element in italic. |
| 5 | <sup> | The <sup> element is used to contain characters that should be superscript such as the suffixes of dates or mathematical concepts like raising a number to a power such as $2^4$. |
| 6 | <sub> | The <sub> element is used to contain characters that should be subscript. It is commonly used with foot notes or chemical formulas such as $H_2O$. |
| 7 | <del> | The <del> element can show text that has been deleted from document. the content of a <del> element usually has a line through it. |
| 8 | <ins> | The <ins> element can be used to show content that has been inserted into a document. The content of a <ins> element is usually underlined. |
| 9 | <mark> | The HTML <mark> element defines marked or highlighted text.<br>**HTML5** |
| 10 | <small> | The HTML <small> element defines smaller text. |

### White Space Collapsing

In order to make code easier to read, web page authors often add extra spaces or start some elements on new lines. However, when the browser comes across two or more spaces next to each other, it only displays one space. Similarly if it comes across one or more lines, it treats that as a single space too. This is known as white space collapsing.

**HTML Code:**

```
<p>The moon is drifting away from Earth.</p>
<p>The moon                 is drifting away from              Earth.</p>
<p>The moon is drifting away from

Earth.</p>
```

**Output:**

```
The moon is drifting away from Earth.

The moon is drifting away from Earth.

The moon is drifting away from Earth.
```

### Line Breaks (New Line) and Horizontal Rules (Drawing Line)

As you have already seen, the browser will automatically show each new paragraph or heading on a new line. But if you wanted to add a line break inside the middle of a paragraph you can use the line break tag <br>. Similarly you may use the element <hr> to draw a line.

### Empty Elements

There are a few elements that do not have a closing tag. They are known as empty elements. <br> and <hr> are empty elements.

Empty elements can be written like this also: <br /> i.e. before the closing angled bracket of an empty element give be a space and a forward slash character.

In fact it is a good programming habit to write an empty element like this: <br />

### The HTML <pre> Element

The HTML <pre> element defines preformatted text.

The text inside a <pre> element is displayed in a fixed-width font (usually Courier) and it preserves both spaces and line breaks:

### Escape Characters

There are some characters that are used in and reserved by HTML code. For example, the left and right angled brackets.

Therefore, if you want these characters to appear on your page you need to use what are termed "escape" characters (also known as escape codes or entity references). For example, to write a left angled bracket, you can use either &lt; or &#60;. For an ampersand, you can use either &amp; or &#38;.

There are also special codes that can be used to show symbols such as copyright and trademark, currency symbols, mathematical characters, and some punctuation marks. For example, if you want to include a copyright symbol on a web page you can use either &copy; or &#169;.

Some common escape characters

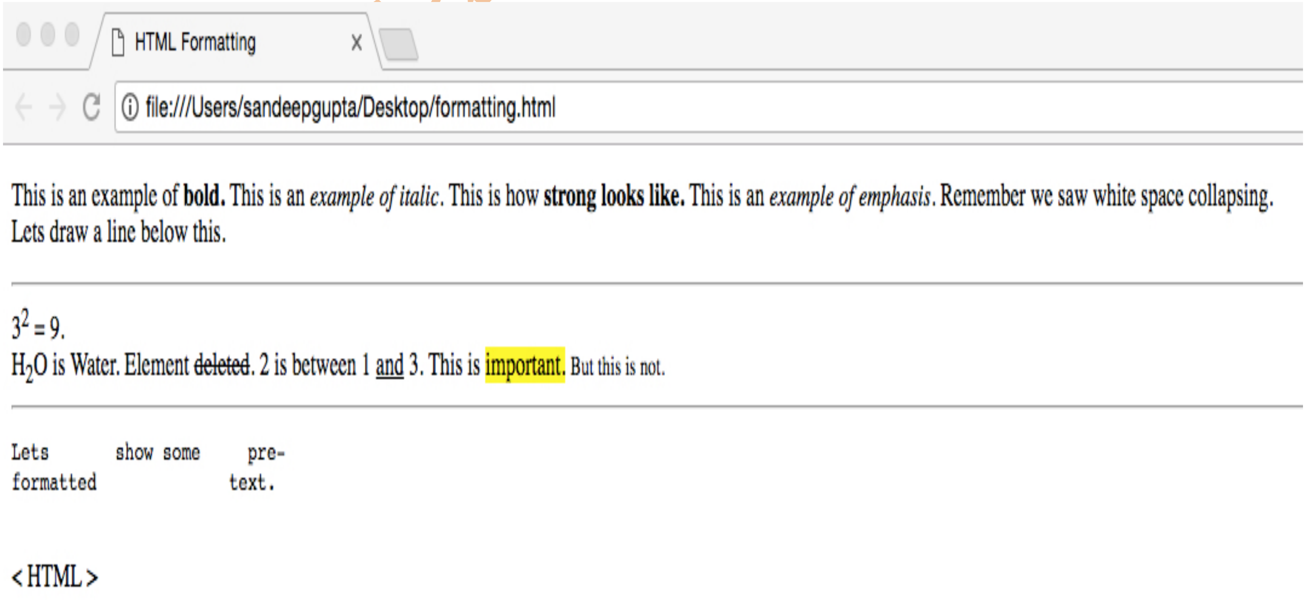
| | |
|---|---|
| **<** | **&lt; / &#60;** |
| **>** | **&gt; / &#62;** |
| **&** | **&amp; / &#38;** |
| **"** | **&quot; / &#34;** |
| **©** | **&copy; / &#169;** |
| **®** | **&reg; / &#174;** |

***formatting.html***

```
<!DOCTYPE html>
<html>
  <head>
        <title>HTML Formatting</title>
  </head>
  <body>
         <p>This is an example of <b>bold. </b>
                         This is an <i>example of italic. </i>
                         This is how <strong>strong looks like. </strong>
                         This is an <em>example of emphasis. </em>
                         Remember       we
                         saw                      white space
                         collapsing.<br />
                         Lets draw a line below this.<hr>
                         3<sup>2</sup> = 9.
                         <br>
                         H<sub>2</sub>O is Water.
                         Element <del>deleted</del>.
                         2 is between 1 <ins>and</ins> 3.
                         This is <mark>important.</mark>
                         <small> But this is not. </small>
                         <hr />
                         <pre>Lets     show some    pre-
formatted                text.</pre>
                         <br />
                         &lt; HTML &gt;

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

<!--The very first line is required for HTML5. We use it since we are using mark element. Also note the syntax of comment -->



This is an example of **bold.** This is an *example of italic*. This is how **strong looks like.** This is an *example of emphasis*. Remember we saw white space collapsing. Lets draw a line below this.

$3^2 = 9$.
$H_2O$ is Water. Element ~~deleted~~. 2 is between 1 and 3. This is important. But this is not.

```
Lets     show some    pre-
formatted            text.
```

< HTML >

# HTML Attributes

All HTML elements can have **attributes.**
Attributes provide **additional information** about an element.
Attributes are always specified in **the start tag**.
Attributes usually come in name/value pairs like: **name="value"**

**Example:**
<p title="I am a tooltip"> This is a paragraph. </p>

<p lang="en-us">Paragraph in English</p>

The attribute name indicates what kind of extra information you are supplying about the element's content. It should be written in lowercase.

The value is the information or setting for the attribute. It should be placed in double quotes. Different attributes can have different values.

In second example, an attribute called lang is used to indicate the language used in the paragraph element. The value of this attribute on this page specifies it is in US English.

HTML5 allows you to use uppercase attribute names and omit the quotemarks, but this is not recommended.

The majority of attributes can only be used on certain elements, although a few attributes (such as lang) can appear on any element.

Most attribute values are either pre-defined or follow a stipulated format. We will look at the permitted values as we introduce each new attribute.

*Refer attributes.html*

## id attribute

Every HTML element can carry the id attribute. It is used to uniquely identify that element from other elements on the page. Its value should start with a letter or an underscore (not a number or any other character). It is important that no two elements on the same page have the same value for their id attributes. The id attribute is known as a **global attribute** because it can be used on any element.

## class attribute

Every HTML element can also carry a class attribute. Sometimes, rather than uniquely identifying one element within a document, you will want a way to group several elements as part of one particular group. For example, you might have some paragraphs of text that contain information that is more important than others and you may want to put these paragraphs in one group. The class attribute of HTML helps you do this. If you would like to indicate that an element belongs to several classes, you can separate class names with a space.

id and class attributes are especially useful when HTML is used with CSS & Javascript.

*Refer selectors_css.html (covered later)*

## HTML Fonts

The <font> tag specifies the font face, font size, and color of text.
The <font> tag is not supported in HTML5. Use CSS instead.
In HTML5 we make use of CSS to work with fonts.

*Refer Fonts.html*

## HTML Anchor Tag/ Creating Links or Hyperlinks on a Webpage

Links are an important feature of the web because they allow you to move from one web page to another — enabling the very idea of browsing or surfing. You will commonly come across the following types of links:

- Links from one website to another
- Links from one page to another on the same website
- Links from one part of a web page to another part of the same page

In HTML, links are defined with the <a> tag also called Anchor tag.

General Format: <a href="*url*">*link text*</a>

The href attribute specifies the destination address of the link. For example,
*https://www.study-circle.org*

The link text is that part of tag which the user is going to see. Clicking on the link text will take you to the specified address. For example,
*Click here to to know more about us.*

Example: <a href="*http://www.study-circle.org/Batches/3.pdf* ">*Download Third Year Batches Timetable*</a>

**Link Colour**
By default, in all browsers

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

You can change the default colors, by using CSS.

**Linking to Other Sites**

When you link to a different website, the value of the href attribute will be the full web address for the site, which is known as an ***absolute URL.***

URL stands for Uniform Resource Locator. Every web page has its own URL. This is the web address that you would type into a browser if you wanted to visit that specific page.

An absolute URL starts with the domain name for that site, and can be followed by the path to a specific page. If no page is specified, the site will display the homepage.

Example of absolute URL:
<a href="*http://www.study-circle.org*">*Study Circle Official Website*</a>

### Linking to Other Pages on the same site

When you are linking to other pages within the same site, you do not need to specify the domain name in the URL. You can use a shorthand known as a relative URL.

If all the pages of the site are in the same folder, then the value of the href attribute is just the name of the file.

If you have different pages of a site in different folders, then you can use a slightly more complex syntax to indicate where the page is in relation to the current page.

Examples:

```
<a href="downloads.html">Downloads</a>

<a href="music/listings.html">Listings</a>
```

### Creating Bookmark / Linking to a specific part on the same page

At the top of a long page you might want to add a list of contents that links to the corresponding sections lower down. Or you might want to add a link from bottom of the page back to the top of it to save users from having to scroll back to the top.

Before you can link to a specific part of a page, you need to identify the points (also called bookmarks) in the page that the link will go to. You do this using the id attribute (which can be used on every HTML element).

### The target Attribute

The target attribute in <a> specifies where to open the linked document.

The target attribute can have one of the following values:

1. _blank - Opens the linked document in a new window or tab
2. _self - Opens the linked document in the same window/tab as it was clicked (this is default)
3. _parent - Opens the linked document in the parent frame
4. _top - Opens the linked document in the full body of the window
5. framename - Opens the linked document in a named frame

### Link Title

The title attribute specifies extra information about an element.
The information is most often shown as a tooltip text when the mouse moves over the element.

*Refer Link1.html*

***Links2.html***

```
<html>
  <head>
        <title>HTML Links</title>
  </head>
  <body>

   <h1> <a href="http://www.study-circle.org"> Worlds Best Website </a> </h1>

   <h1> <a href="#spa"> 1. SPA </a> </h1>
   <h1> <a href="#ds"> 2. DS & AOA</a> </h1>
   <h1> <a href="#oopm"> 3. OOPM </a> </h1>
   <h1> <a href="#ip"> 4. INTERNET PROGRAMMING </a> </h1>
   <h1> <a href="#ocjp"> 5. OCJP </a> </h1>
   <h1> <a href="#android"> 6. ANDROID APP DEVELOPMENT </a> </h1>

   <h1 id="spa">1. SPA</h1>
   <p> SPA is a subject of sem2 and contains the C programming language.
         It is the foundation of programming. </p>

   <h1 id="ds">2. DS & AOA</h1>
   <p> DS & AOA are subjects of sem 3 and  4 respectively and
         very important subjects for placement in dream companies.  </p>

   <h1 id="oopm">3. OOPM</h1>
   <p> OOPM is a subject of sem 3 and contains Java Programming.
        It is the first subject where you will learn java programming. </p>

   <h1 id="ip">4. INTERNET PROGRAMMING</h1>
   <p> This is a subject of sem 5. It is one of the best subjects of MU.
        Here you will learn HTML, CSS, Javascript. </p>

   <h1 id="ocjp">5. OCJP</h1>
   <p> This is not a subject of MU syllabus. It is an international certification exam on
Java.
        Many students do this certification to get an edge over others. </p>

   <h1 id="android">6. ANDROID APP DEVELOPMENT</h1>
   <p> Another very good subject in sem 7. Need not tell you its importance </p>

  </body>
</html>
```

## Block Elements & Inline Elements

Some elements will always appear to start on a new line in the browser window. These are known as block level elements. Examples of block elements are <h1>, <p>, <ul>, and <li>.

Some elements will always appear to continue on the same line as their neighbouring elements. These are known as inline elements. Examples of inline elements are <a> and <img>.

# Introduction to CSS ( Cascading Style Sheets )

HTML tells the web browser **WHAT** it should display on the web page.
CSS describes **HOW** HTML elements should be displayed.

CSS allows you to create rules that specify how the content of an html element should appear. For example, you can specify that the background of the page is cream, all paragraphs should appear in gray using the Arial font, or that all level one headings should be in a blue, italic, Times font.

HTML was never intended to contain tags for formatting a web page. HTML was created to describe the content of a web page. So, When tags like <font> and color attributes were added to the HTML 3.2 specification, it resulted into a chaos for web developers.

Development of large websites, where fonts and color information were added to every single page was a tedious process using HTML. To solve this problem, the World Wide Web Consortium (W3C) created CSS. CSS removed the style formatting from the HTML page.

Nowadays, the style definitions are normally saved in external .css files. With an external stylesheet file, you can control and change the look of an entire website by using just one CSS file.

# CSS Syntax

CSS works by associating ***rules*** with HTML elements. These rules govern how the content of specified html elements should be displayed.

A CSS rule contains two parts: a selector and a declaration.

**p { font-family: Arial; }**

The above CSS rule indicates that all <p> elements should be shown in the Arial font.

**p** in the above example is a selector which indicates the HTML element to whom the rule applies to. The same rule can apply to more than one element if you separate the element names with commas.

**{ font-family: Arial; }** in the above example is called declaration or declaration block. Declarations indicate how the elements referred to in the selector should be styled. Declarations are split into two parts - a property and a value - and are separated by a colon.

You can specify several properties in one declaration, each separated by a semi-colon.

**h1, h2, h3 { font-family: Arial; color: yellow; }**

This rule indicates that all <h1>, <h2> and <h3> elements should be shown in the Arial typeface, in a yellow color.

Properties indicate the aspects of the element you want to change. For example, color, font, width, height and border. Values specify the setting you want to use for the chosen properties. For example, if you want to specify a color property then the value is that color you want the text in these elements to be.

# Using CSS

There are three ways of using CSS:

- External style sheet
- Internal style sheet
- Inline style

## 1) External style sheet

This technique uses two documents: the HTML file and a separate CSS file.

The CSS file should not contain any html tags. The style sheet file must be saved with a .css extension.

The HTML document must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section. The <link> element indicates where the CSS file is located.

**<link>**
The **<link>** element can be used in an HTML document to tell the browser where to find the CSS file which will be used to style the page. It is an empty element (meaning it does not need a closing tag) and it lives inside the **<head>** element. It should use three attributes:

**href**
This specifies the path to the CSS file (which is often placed in a folder called **css** or **styles**).

**type**
This attribute specifies the type of document being linked to. The value should be **text/css**.

**rel**
This specifies the relationship between the HTML page and the file it is linked to. The value should be **stylesheet** when linking to a CSS file.

An HTML page can use more than one CSS style sheet. For example, one CSS file could control size of text and another its colour. To do this the HTML file could have a **<link>** element for every CSS file it uses.

ext_html.html

```
<html>

  <head>
        <title>Using External CSS</title>
        <link  href="css/styles1.css"  type="text/css"  rel="stylesheet" />
        <link  href="css/styles2.css"  type="text/css"  rel="stylesheet" />
  </head>

  <body>

        <h1>Important Programming Subjects: </h1>


        <h2>1. SPA</h2>
        <p> SPA is a subject of sem 2 and contains the C programming language.
              It is the foundation of programming. </p>

        <h2>2. DS & AOA</h2>
        <p> DS & AOA are subjects of sem 3 and 4 respectively and
              very important subjects for placement in dream companies.  </p>

       <h2>3. OOPM</h2>
       <p> OOPM is a subject of sem 3 and contains Java Programming.
             It is the first subject where you will learn java programming. </p>

  </body>

</html>
```

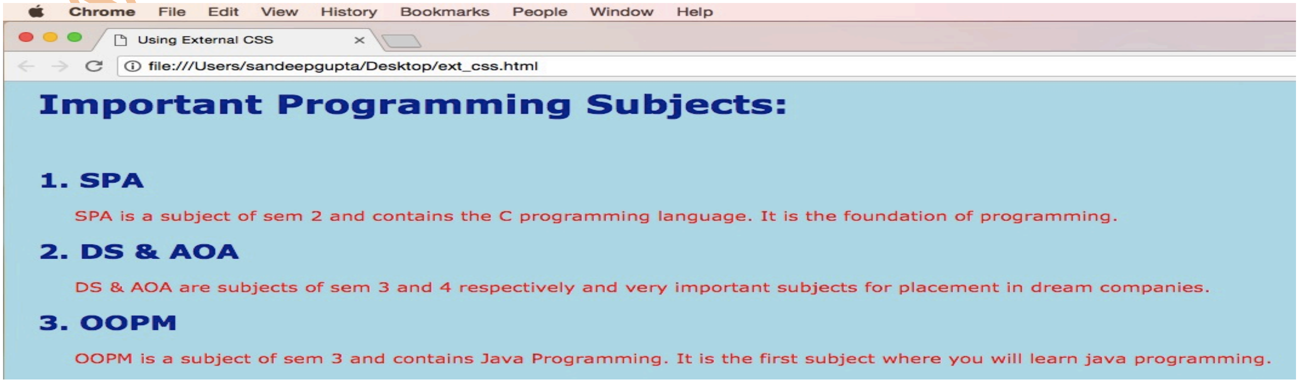
styles1.css

```
body {
    background-color: lightblue;
    font-family: verdana;
}
```

styles2.css

```
h1, h2 {
    color: darkblue;
    margin-left: 20px;
}

p {
         color: red;
         margin-left: 50px;
}
/* No space should be given between
20(or 50) and px */
```



Important Programming Subjects:

1. SPA

SPA is a subject of sem 2 and contains the C programming language. It is the foundation of programming.

2. DS & AOA

DS & AOA are subjects of sem 3 and 4 respectively and very important subjects for placement in dream companies.

3. OOPM

OOPM is a subject of sem 3 and contains Java Programming. It is the first subject where you will learn java programming.

**Note:**

1. If we put

```
body {
        background-color: white;
        font-family: verdana;
}
```

   in styles2.css then background colour will be white since in <link> styles2.css is defined after styles1.css.

2. We could have used only one style sheet here.

## 2) Internal Style Sheet

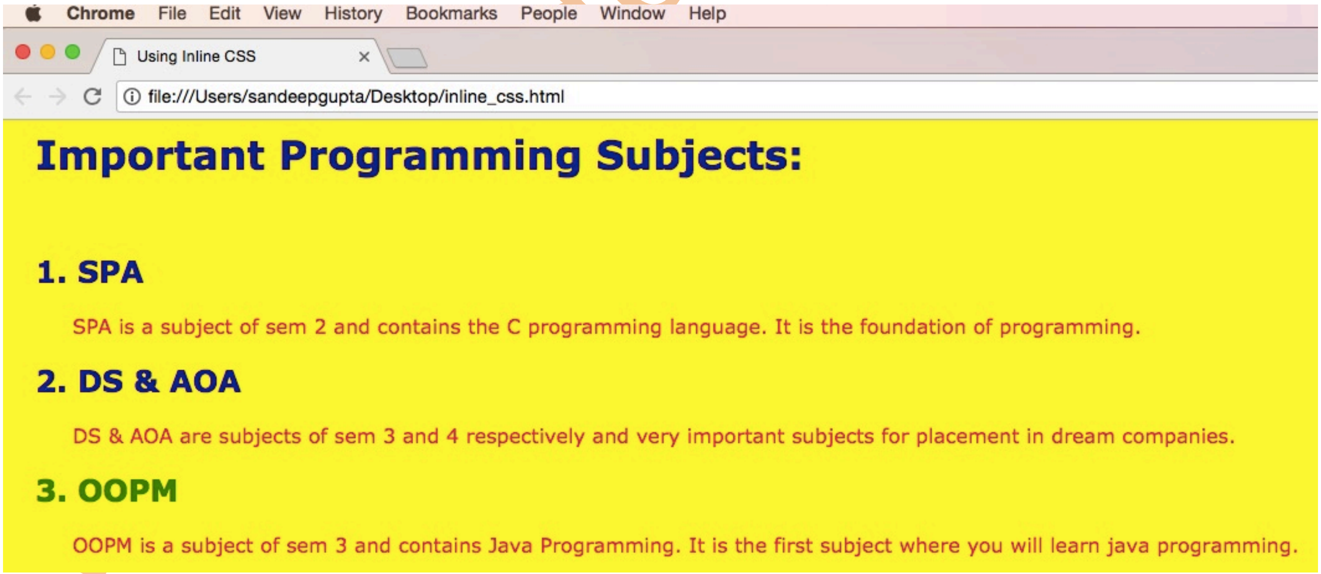
Instead of creating an external style sheet, you can also include CSS rules within an HTML page by placing them inside a **<style>** element, which usually sits inside the <head> element of the page.

The <style> element should use the type attribute to indicate that the styles are specified in CSS. The value should be **text/css**.

*Refer int_css.html*

## 3) Inline Style

An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.


Chrome File Edit View History Bookmarks People Window Help
Using Inline CSS
file:///Users/sandeepgupta/Desktop/inline_css.html

Important Programming Subjects:

1. SPA

SPA is a subject of sem 2 and contains the C programming language. It is the foundation of programming.

2. DS & AOA

DS & AOA are subjects of sem 3 and 4 respectively and very important subjects for placement in dream companies.

3. OOPM

OOPM is a subject of sem 3 and contains Java Programming. It is the first subject where you will learn java programming.

***inline_css.html***

```
<html>

  <head>

        <title>Using Inline CSS</title>

        <style type="text/css">

          body {
                background-color: yellow;
                font-family: verdana;
          }

          h1, h2 {
                color: darkblue;
                margin-left: 20px;
          }

          p {
                color: red;
                margin-left: 50px;
          }

         </style>

        </head>

  <body>

        <h1>Important Programming Subjects: </h1>


        <h2>1. SPA</h2>
        <p> SPA is a subject of sem 2 and contains the C programming language.
          It is the foundation of programming. </p>

        <h2>2. DS & AOA</h2>
        <p> DS & AOA are subjects of sem 3 and 4 respectively and
          very important subjects for placement in dream companies.  </p>

        <h2  style = "color:green;">3. OOPM</h2>
        <p> OOPM is a subject of sem 3 and contains Java Programming.
         It is the first subject where you will learn java programming. </p>

  </body>

</html>
```

**Note:**
Inline is given more importance than internal. We call this cascading.

## When to use external CSS ?

The following are the advantages of creating a separate CSS file When building a website.

1. All of your web pages can share the same style sheet. This is achieved by using the <link> element on each HTML page of your site to link to the same CSS document. This means that the same code does not need to be repeated in every page (which results in less code and smaller HTML pages). Therefore, once the user has downloaded the CSS stylesheet, the rest of the site will load faster.

2. If you want to make a change to how your site appears, you only need to edit the one CSS file and all of your pages will be updated. For example, you can change the style of every <h1> element by altering the one CSS style sheet, rather than changing the CSS rules on every page.

3. The HTML code will be easier to read and edit because it does not have lots of CSS rules in the same document. It is generally considered good practice to have the content of the site separated from the rules that determine how it appears.

## When to use internal CSS ?

1. If you are just creating a single page, you may put the css rules in the same file to keep everything in one place.

2. If you have one page which requires a few extra rules (that are not used by the rest of the site), you might consider using CSS in the same page.

**In general, keeping all CSS rules in a separate file is considered to be a good programming practice.**

## Cascading Rules

Sometimes it is possible that more than one css style is specified for one particular HTML element. *(Refer inline_css).* In that case, the following priority is used:

- Inline style (inside an HTML element)
- External and internal style sheets (in the head section)
  The css rule specified later is given more importance
- Browser default