Table of Contents

Contents

[1Web Development in 1 Hour 2](#__RefHeading___Toc1681_1575552429)

[1.1Components of Web Development 2](#__RefHeading___Toc1683_1575552429)

[1.1.1Intro 2](#__RefHeading___Toc1685_1575552429)

[1.1.2HTML 2](#__RefHeading___Toc1687_1575552429)

[1.1.3CSS 3](#__RefHeading___Toc1689_1575552429)

[1.2Structuring HTML Tags 3](#__RefHeading___Toc1691_1575552429)

[1.2.1<!DOCTYPE HTML> : 3](#__RefHeading___Toc1693_1575552429)

[1.2.2<html> : 3](#__RefHeading___Toc1695_1575552429)

[1.2.3<head> : 4](#__RefHeading___Toc1697_1575552429)

[1.2.4<body> : 4](#__RefHeading___Toc1699_1575552429)

[1.3Various HTML Tags 4](#__RefHeading___Toc1701_1575552429)

[1.3.1Heading Tags 4](#__RefHeading___Toc1703_1575552429)

[1.3.2Paragraph Tag 4](#__RefHeading___Toc1705_1575552429)

[1.3.3Line Break Tag 4](#__RefHeading___Toc1707_1575552429)

[1.3.4Preserve Formatting 4](#__RefHeading___Toc1709_1575552429)

[1.3.5Non Breakable Space 5](#__RefHeading___Toc1711_1575552429)

[1.3.6Horizontal Lines 5](#__RefHeading___Toc1713_1575552429)

[1.3.7Division Tag 5](#__RefHeading___Toc1715_1575552429)

[1.3.8Image Tag 5](#__RefHeading___Toc1717_1575552429)

[1.3.9HTML Tables 5](#__RefHeading___Toc1719_1575552429)

[1.3.10HTML Lists 5](#__RefHeading___Toc1721_1575552429)

[1.3.11Links 6](#__RefHeading___Toc1723_1575552429)

[1.3.12HTML Forms 6](#__RefHeading___Toc1725_1575552429)

[1.4Semantic Tags 7](#__RefHeading___Toc1727_1575552429)

[2CSS 7](#__RefHeading___Toc1750_1575552429)

[2.1Advantages of CSS 8](#__RefHeading___Toc1016_1575552429)

[2.1.1CSS saves time 8](#__RefHeading___Toc1752_1575552429)

[2.1.2Pages load faster 8](#__RefHeading___Toc1754_1575552429)

[2.1.3Easy maintenance 8](#__RefHeading___Toc1762_1575552429)

[2.2Types of CSS 8](#__RefHeading___Toc1760_1575552429)

[2.2.1External CSS 8](#__RefHeading___Toc1020_1575552429)

[2.2.2Internal CSS 8](#__RefHeading___Toc1022_1575552429)

[2.2.3Inline CSS 9](#__RefHeading___Toc1024_1575552429)

[2.3Selector 9](#__RefHeading___Toc1026_1575552429)

[3.CSS Box Model 10](#__RefHeading___Toc1028_1575552429)

[4960 Grid System 11](#__RefHeading___Toc1030_1575552429)

[4.2Advantages 11](#__RefHeading___Toc1032_1575552429)

[4.3Why 960 11](#__RefHeading___Toc1758_1575552429)

# Web Development in 1 Hour

## Components of Web Development

### Intro

What we see in a website is most probably the effect of HTML and CSS working together. The web browser interprets the HTML and CSS code to create the page that you see. Whether it is text, colors, images, audio, video etc are all structured on the page using these two technologies. So now the question arises what exactly is HTML and CSS.

### HTML

HTML stands for HyperText Markup Language. Let’s understand it word by word.

* HyperText is the method by which you move around on the web — by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.
* Markup is what HTMLtags do to the text inside them. They mark it as a certain type of text.

HTML consists of a series of codes typed into a specific syntax within angular brackets. These codes in angular brackets are called ‘Tags’.Tags are what separates normal text from HTML code.Tags are interpreted by the browser and each tag has a specific property that it signifies. That property is impacted on the text that is mentioned within it. Tags are themselves not displayed on the web page but browser interprets the property attached with the tags and that property is then reflected on the content within the tags.

Tags also contain various attributes which are used to change the default functioning of the tags. These attributes can be used to get the desired property onto the content within the tags.

### CSS

CSS stands for Cascading Style Sheets. As the name suggests it is used to ‘cascade styles’ on our web page. It can be used to define styles which cannot be defined by using HTML tags and their attributes. It also does the work of separating the style codes with the content codes so that if we want to repeat a particular style for multiple elements, code duplication can be avoided. This makes our code shorted, cleaner and easy to debug.

## Structuring HTML Tags

HTML tags follow a parent child relationship. This means that tags can have parent tags, child tags and siblings. They can also be nested together one inside other. The basic structure followed by all the HTML5 pages is:

**<!DOCTYPE HTML >**

**<html>**  
 **<head>  
    <title>My First HTML Page</title>  
 </head>  
  
 <body>  
  
    My text goes here.  
  
 </body>  
  
</html>**

Let’s look at the components of this skeleton format one by one.

### <!DOCTYPE HTML> :

This part tells the browser that the document that it is currently handling is a HTML5 document. It basically is used to determine the version of the document so that several tags can be identified.

### <html> :

This is the parent tag of all the tags in the HTML document( except the DOCTYPE tag). All the data within this tag is interpreted by the browser to produce the outcome of the child tags to be displayed on the web page. This is also called as the root tag of HTML document.

### <head> :

Head tag contains the data which is not meant to be displayed on the web page(except title) but is needed by the browser to interpret the rest of the page data. It can contain links to any external document, some custom styling and scripting elements and also the metadata about the body elements.

### <body> :

This is the part of the web page that contains data that is to be displayed on the screen or the web page body. It can contain texts, images, audios, videos etc…The various body tags which are used in HTML are discussed later in this blog.

## Various HTML Tags

### Heading Tags

Heading tags are used to create heading(as the name suggests) and can also be used to create important texts which appear to stand out from the rest of the plain text. They are by default bold and have a line gap above and below it.

There are 6 headings in HTML namely <h1>, <h2>,<h3>, <h4>,<h5>,<h6>, h1 being the largest and h6 being the smallest.

### Paragraph Tag

Again as the name suggests, this tag creates a paragraph. It is basically used to divide the content into different structured paragraphs to ease reading purposes. The text written within this tag is grouped together and is separated from rest of the text by leaving a line gap before and after it.

It is created using <p> … </p> syntax.

When 2 paragraphs are written one after the other, the line gap between each of them is cushioned together and instead of displaying 2 line gaps(one gap by each of the paragraphs) only a single line gap is visible.

### Line Break Tag

When we enter text in HTML documents, content within tags is treated as a part of a single line. Even after pressing enter key, line break is not applied. So, to provide line breaks this tag is used. It created a line gap between text above and below it.

Its syntax is <br>.

### Preserve Formatting

Again the story from break tag continues, when entering text within tags no multiple spaces and breaks are interpreted. So to allow text to be written following a particular design and format, this tag is used. It displays the content exactly as written within the tags.

Its syntax is <pre> ---Content---</pre>

### Non Breakable Space

Non breakable space is used where we don’t want to split text into a different line. Words separated by a non-breakable space will stick together and won’t split into a new line.

It is represented as &nbsp;

### Horizontal Lines

Horizontal lines are used to visibly break the sections of a document. It created line from the current position in the document to the right margin of the document.

It is written as <hr>

### Division Tag

This tag is used to create divisions in our text. It basically acts as the entity which differentiates different divisions of a document. Visibly it is similar to a paragraph tag but it indicates a meaning to the coder that it is a different division in a document.

Its syntax is <div> ---content---</div>

### Image Tag

Image tag allows us to display images on our web page. We can display any image by providing its source address with src attribute of image tag. We can even provide an alternate text in case due to some conflict image is not displayed.

Its syntax is: <img src=” ” alt=” ” >

### HTML Tables

We can also create tables with different rows and columns it our web page. In these tables we can arrange different types of data like images, links, forms etc…

Table is created using <table> tag. To create a row we use <tr> tag which corresponds to table row. For a column we can rather use <th> tag or <td> tag. <th> stands for table heading and <td> stands for table data. We can also merge 2 or more rows or columns using colspan and rowspan attributes for rows and columns respectively.

### HTML Lists

We can display information point to point using lists. HTML offers three types of lists:

#### Ordered Lists:

These lists are numbered lists. The numbers can vary from numeric to roman numerals to different styles.

They are created using <ol> tag and for each list item <li> tag is used.

#### Unordered Lists

These lists are not numbered. Instead they display data using bullet points. The bullets may be of several types like circles, squared, ring etc…

They are created using <ul> and similarly for each list item <li> is used.

#### Definition Lists

They are also called description lists. They display data in the form of terms along with description of each term in the list.

They are created using <dl> tag. For each definition term we use <dt> and for each description <dd> is used.

### Links

Links are used to create paths from on page to another or to join different entities into a single page.

These may include an address to external documents.

Links are created using anchor tag. Along with anchor ‘href’ attribute is passed which contains address to the link.

e.g.: <a href=[http://www.google.co.in](http://www.google.co.in/)> Google </a>

### HTML Forms

Forms are used to collect data from the user. Forms are used to gather information from the web page and then send it to a backend application for processing purposes. One of the children of form is input element.

Input element is of many types namely text, email, password, submit, button etc… Depending upon the type of input element the display and functioning is affected. For example text displays a text field and password displays a text field in which text is displayed as dots or asterisk.

Submit button is one of the most important element of a form without which data cannot be sent to the back end for evaluation. Target is also specified within the form wherein the target server is pointed. We can also specify the method with which the data is to be transferred. It can either be Get or Post. Get method sends data by appending the data values onto the link. This poses a problem of security because data can be copied from the URL. That is where Post comes into action. Post doesn’t display data in the URL. The data inputted into the URL is kept hidden from external elements. Thus it can be used to send classified data.

The various elements which can be included in a form are text field, drop down list created using <select> tag and each drop down item is created using <option> tag, radio buttons which can be grouped such that only one of them is true or even more than one, check box which are somewhat similar to radio buttons.

## Semantic Tags

Most tags which we have studied till now are non semantic tags. They are called so because they don’t tell us much about what’s inside them and what they convey. HTML5 included some semantics which define some structural format and also throw an insight on their purpose.

The various semantic tags are:

<header>

<nav>

<section>

<article>

<aside>

<footer>

They actually don’t do anything special. They just tell the coder about the use of the code that they cover. This makes reading easy and also debugging is also made time efficient. The basic structure of semantic tags is represented in the diagram.

This was some insight into HTML. Now we have the content for our website ready but styling is still missing. HTML only creates content of the web page but CSS is needed to make our page attractive and provide a good look and feel.

# CSS

CSS stands for Cascading Style Sheet. **Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the presentation of a document written in HTML.CSS is designed primarily to enable the separation of presentation and content, including aspects such as the [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface).This separation can improve content [accessibility](https://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

## Advantages of CSS

### **CSS saves time**

You can write CSS once and then reuse 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.

### Pages load faster

If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

### Easy maintenance

To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

## Types of CSS

1. External CSS

2. Internal CSS

3. Inline CSS

### External CSS

The <link> element can be used to include an external stylesheet file in your HTML document.

An external style sheet is a separate text file with**.css** extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.

Its syntax is:

<head>

<link type = "text/css" href = "..." media = "..." />

</head>

### Internal CSS

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document.

The syntax for Internal CSS is:

<head>

<style type = "text/css" media = "all">

body {

background-color: linen;

}

h1 {

color: maroon;

margin-left: 40px;

}

</style>

</head>

### Inline CSS

You can use *style* attribute of any HTML element to define style rules. These rules will be applied to that element only.

Its syntax is:

<element style = "...style rules....">

## Selector

In CSS, *selectors* declare which part of the markup a style applies to by matching tags and attributes in the markup itself.

Selectors may apply to:

1. all elements of a specific type, e.g. the second-level headers [h2](https://en.wikipedia.org/wiki/HTML_element" \l "Basic_text)
2. elements specified by [attribute](https://en.wikipedia.org/wiki/HTML_attribute), in particular:
   * *id*: an identifier unique within the document
   * *class*: an identifier that can annotate multiple elements in a document
3. elements depending on how they are placed relative to others in the [document tree](https://en.wikipedia.org/wiki/Document_Object_Model).

Classes and IDs are case-sensitive, start with letters, and can include alphanumeric characters and underscores. A class may apply to any number of instances of any elements. An ID may only be applied to a single element.

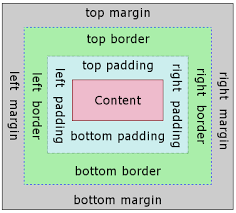
# CSS Box Model

When laying out a document, the browser's rendering engine represents each element as a rectangular box according to the standard CSS box model. CSS determines the size, position, and properties (color, background, border size, etc.) of these boxes.

Every box is composed of four parts (or areas), defined by their respective edges: the content edge, padding edge, border edge, and margin edge.

The content area, bounded by the content edge, contains the "real" content of the element, such as text, an image, or a video player. Its dimensions are the content width (or content-box width) and the content height (or content-box height). It often has a background color or background image.

The **padding area**, bounded by the padding edge, extends the content area to include the element's padding. Its dimensions are the padding-box width and the padding-box height. When the content area has a background, it extends into the padding.  
The **border area**, bounded by the border edge, extends the padding area to include the element's borders. Its dimensions are the border-box width and the border-box height.  
The margin area, bounded by the margin edge, extends the border area to include an empty area used to separate the element from its neighbors. Its dimensions are the margin-box width and the margin-box height.



# 960 Grid System

The 960 Grid System is an effort to streamline web development workflow by providing commonly used dimensions, based on a width of 960 pixels. There are two variants: 12 and 16 columns.The 12-column grid is divided into portions that are 60 pixels wide. The 16-column grid consists of 40 pixel increments. Each column has 10 pixels of margin on the left and right, which create 20 pixel wide gutters between columns.

## Advantages

* Provides a "framework" that's designed to look good on all monitors.
* Streamlines the design process by defining exact measurements.
* Reduces development time by providing pre-coded HTML/CSS.
* In a perfect world, it helps designers and developers communicate better - smoothing out the process of moving from Design to Coding.

Ultimately, helps establish some basic guidelines for content columns, while still providing designers with full control over their designs.  
960.gs is based around putting all of your site's elements in a 960px wide container, and dividing that container into 12, 16, or 24 equally sized columns. They provide a tool to use alternate widths, but 960 is what the entire system is developed for and it happens to look pretty sharp.

## Why 960

Because 960px is a width that is suited for the wide number of platforms on which we browse the web. It essentially allows for a 1024px wide monitor to show the site accurately and without horizontal scrolling, accounting for the width of the browser chrome, scrollbars, and a bit of padding for legibility. There is always a 10px margin placed at the right and left of the main content column, which means that smaller browsers will always be able to read the farthest left content without the text butting but against the browser window.

