



# Web Site Design & Development

## Introduction

# What is Website?

Website is a location on web and is hosted on a web server. It is a set of related web pages. It is accessed using Internet address known as Uniform Resource Locator

# How to access Websites?

When we type a certain URL in a browser search bar, the browser requests the page from the Web server and the Web server returns the required web page and its content to the browser. Now, it differs from how the server returns the information required in the case of static and dynamic websites.



# URL Components

- **The protocol** used to access the website, which in this case is http, meaning port 80. It can also be https; port 443.
- **The subdomain** which by default is www.
- **The domain name**; domain names are normally chosen to have a meaning. Like in our case “**google**”, we can understand that this website offers tutorials.
- **The suffix** name which can be .com, .info, .net, .biz, or country specific. For detailed information, please refer the following Wikipedia link:  
*[https://en.wikipedia.org/wiki/List\\_of\\_Internet\\_top-level\\_domains](https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains)*.



# Website folder structure

## Website Folder Structure

All .html files go here in top-level of the folder.  
index.html is the name of the homepage

INNER FOLDERS

CSS

All .css files go here

images

All images go here.  
examples: .jpg, .jpeg,  
.gif, .png

javascript  
or  
js

All .js files go here

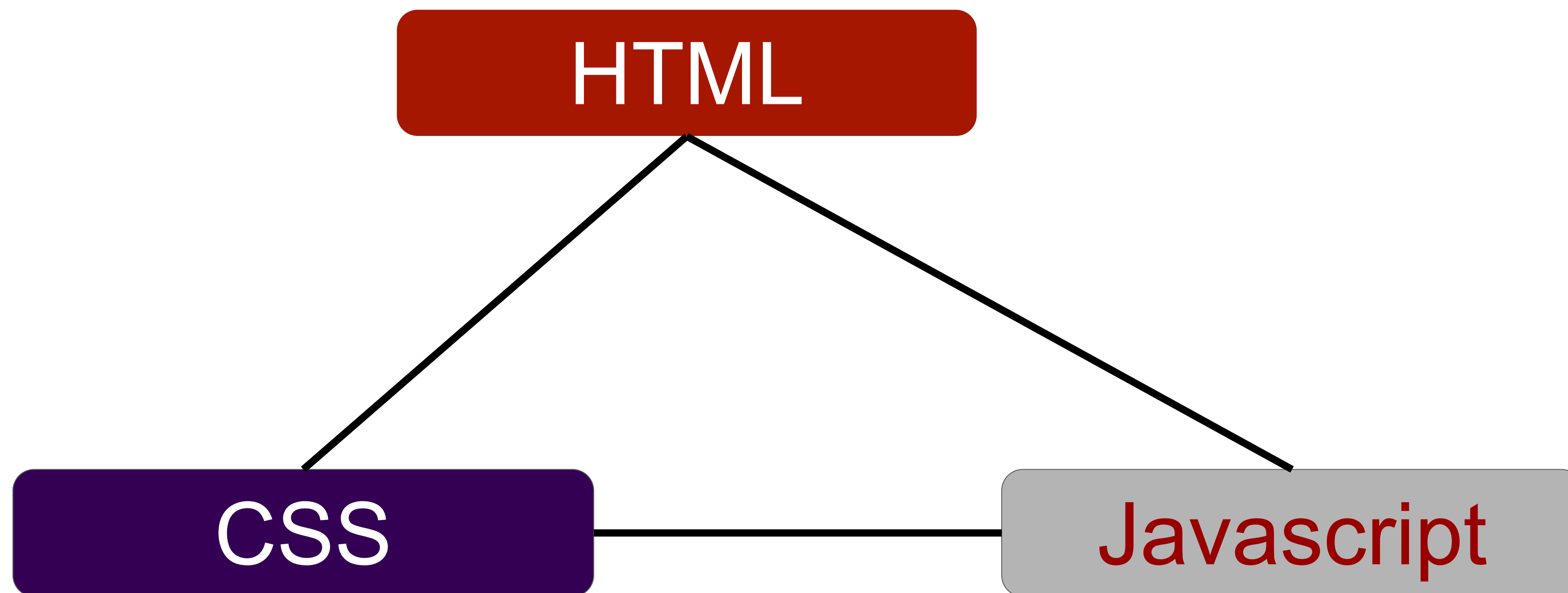
- **Project (Root) Folder:** This is the main folder for the project and contains ALL other files and folders. On the web it's called the root folder. It's not shown in the image above; consider it the grey box around everything.
- **HTML files:** these are all of the files that end with `.html`. These should be directly in the Project/Root folder and not in any sub folders. Doing that makes it easy to see all of your files and makes all of the paths to resources (images, css files) the same. The only required file is `index.html` but your sites will all have more `.html` files.
- **CSS Folder:** all `.css` files go here. In this class we will not write css either in style elements on the web page or in attributes of elements. Instead all CSS will go in external stylesheets and those will be put in the css folder. You will need to have at least one stylesheet called either `main.css` or `style.css`. I will generally use `main`, but some of my tutorials and others use `style.css` so it's OK if you use that. But don't use both. Other CSS files should have names referring to what they are styling.
- **Images Folder:** all images go here (`.jpg`, `.png`, `.gif`, etc). You can name the folder `images` or `img`.
- **JavaScript Folder:** all JavaScript files go here. They end in `.js`. You can name the folder `javascript` or `js`.
-

# Website Authoring tools?

HTML text editors are used to create and modify web pages. HTML codes can be written in any text editors including the notepad. One just needs to write HTML in any text editor and save the file with an extension “.html”. Some of the popular HTML text editors are given below:

- Notepad
- Notepad++
- Sublime Text 3
- Atom
- GeeksforGeeks IDE
- Visual Studio

# The Web Programming Triangle



# What is HTML?

**HTML** stands for **HyperText Markup Language**. It is used to design web pages using the **markup language**. HTML is the combination of **Hypertext** and **Markup language**. Hypertext defines the link between the web pages and markup language defines the text document within the tag that define the structure of web pages.



# What is HTML used for ?

HTML is used to create the structure of web pages that are displayed on the World Wide Web (www). It contains Tags and Attributes that are used to design the web pages. Also, we can link multiple pages using Hyperlinks.

# HTML Basic Format Page Structure

The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.

## HTML Page Structure

```
<!DOCTYPE html>    ← Tells version of HTML
<html>             ← HTML Root Element

<head>             ← Used to contain page HTML metadata
  <title>Page Title</title> ← Title of HTML page
</head>

<body>             ← Hold content of HTML
  <h2>Heading Content</h2> ← HTML heading tag
  <p>Paragraph Content</p> ← HTML paragraph tag
</body>

</html>
```

# HTML Basic Format Page Structure

- **<DOCTYPE! html>** – A doctype or document type declaration is an instruction that tells the web browser about the markup language in which the current page is written. It is not an element or tag. The doctype declaration is not case-sensitive.
- **<html>** – This tag is used to define the root element of HTML document. This tag tells the browser that it is an HTML document. It is the second outer container element that contains all other elements within it.
- **<head>** – This tag is used to define the head portion of the HTML document that contains information related to the document. Elements within the head tag are not visible on the front-end of a webpage.
- **<body>** – The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front end.



**Example 1:** This is the basic example of HTML that display the heading and paragraph content.

```
<!DOCTYPE html>
<html>

  <!-- Head Section content -->
  <head>

    <!-- Page title -->
    <title>Basic Web Page</title>
  </head>

  <!-- Body Section content -->
  <body>

    <!-- Used to display heading content -->
    <h1>Welcome to GeeksforGeeks</h1>

    <!-- Used to display paragrapg content -->
    <p>A computer science portal for geeks</p>
  </body>
```

# HTML Elements

## (tags & attributes)

An HTML element is the collection of start and end tags with the content inserted in between them.

Tags and attributes are the basis of HTML. They work together but perform different functions .

### What Are HTML Tags?

**Tags** are used to **mark up the start of an HTML element** and they are usually enclosed in angle brackets. An example of a tag is: `<h1>`.

Most tags must be opened `<h1>` and closed `</h1>` in order to function.

### What are HTML Attributes?

**Attributes** contain **additional pieces of information**. Attributes take the form of an opening tag and additional info is **placed inside**.

An example of an attribute is:

```

```

In this instance, the image source (src) and the alt text (alt) are attributes of the `<img>` tag.

### Golden Rules To Remember

1. The vast majority of tags must be **opened** (`<tag>`) and **closed** (`</tag>`) with the element information such as a title or text resting between the tags.
2. When using multiple tags, the tags must be **closed in the order in which they were opened**. For example:



# Formatting Tags

Tag	Description
<code>&lt;b&gt; &lt;/b&gt;</code>	Specifies the text as bold. Eg. this is bold text
<code>&lt;em&gt; &lt;/em&gt;</code>	It is a phrase text. It specifies the emphasized text. Eg. <i>Emphasized text</i>
<code>&lt;strong&gt; &lt;/strong&gt;</code>	It is a phrase tag. It specifies an important text. Eg. this is strong text
<code>&lt;i&gt; &lt;/i&gt;</code>	The content of italic tag is displayed in italic. Eg. <i>Italic text</i>
<code>&lt;sub&gt; &lt;/sub&gt;</code>	Specifies the subscripted text. Eg. X <sub>1</sub>
<code>&lt;sup&gt; &lt;/sup&gt;</code>	Defines the superscripted text. Eg. X <sup>2</sup>
<code>&lt;ins&gt; &lt;/ins&gt;</code>	Specifies the inserted text. Eg. The price of pen is now 2015.
<code>&lt;del&gt; &lt;/del&gt;</code>	Specifies the deleted text. Eg. The price of pen is now 2015.
<code>&lt;mark&gt; &lt;/mark&gt;</code>	Specifies the marked text. Eg. It is raining
<code>&lt;cite&gt;&lt;/cite&gt;</code>	The <code>&lt;cite&gt;</code> and <code>&lt;/cite&gt;</code> tag pair usually indicates that the enclosed text is a bibliographic citation like a book or a magazine. By convention, the citation is displayed in italic
<code>&lt;code&gt;&lt;/code&gt;</code>	The <code>&lt;code&gt;</code> tag is used to tell the browser that the enclosed text is a program code. The browser renders the enclosed text in a monospaced teletype-style font like courier.

# Table Tags

Tag	Description
<code>&lt;table&gt; &lt;/table&gt;</code>	Specifies a table.
<code>&lt;tr&gt; &lt;/tr&gt;</code>	Specifies a row in the table.
<code>&lt;th&gt; &lt;/th&gt;</code>	Specifies header cell in the table.
<code>&lt;td&gt; &lt;/td&gt;</code>	Specifies the data in an cell of the table.
<code>&lt;caption&gt; &lt;/caption&gt;</code>	Specifies the table caption.
<code>&lt;colgroup&gt; &lt;/colgroup&gt;</code>	Specifies a group of columns in a table for formatting.

# List Tags

Tag	Description
<code>&lt;ul&gt; &lt;/ul&gt;</code>	Specifies an unordered list.
<code>&lt;ol&gt; &lt;/ol&gt;</code>	Specifies an ordered list.
<code>&lt;li&gt; &lt;/li&gt;</code>	Specifies a list item.
<code>&lt;dl&gt; &lt;/dl&gt;</code>	Specifies a description list.
<code>&lt;dt&gt; &lt;/dt&gt;</code>	Specifies the term in a description list.
<code>&lt;dd&gt; &lt;/dd&gt;</code>	Specifies description of term in a description list.

# Form Tags

Forms are used to input the values. These values are sent to the server for processing. Forms uses input elements such as text fields, check boxes, radio buttons, lists, submit buttons etc. to enter the data into it.

The following table describes the commonly used tags while creating a form:

Tag	Description
<code>&lt;form&gt; &lt;/form&gt;</code>	It is used to create HTML form.
<code>&lt;input&gt; &lt;/input&gt;</code>	Specifies the input field.
<code>&lt;textarea&gt; &lt;/textarea&gt;</code>	Specifies a text area control that allows to enter multi-line text.
<code>&lt;label&gt; &lt;/label&gt;</code>	Specifies the label for an input element.

# CSS

CSS is used to control the style of a web document in a simple and easy way.

CSS is the acronym for "Cascading Style Sheet". This tutorial covers both the versions CSS1, CSS2 and CSS3, and gives a complete understanding of CSS, starting from its basics to advanced concepts.

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

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, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.



# Common CSS Terms

CSS syntax includes selectors, properties, values, declarations, declaration blocks, rulesets, at-rules, and statements.

- A **selector** is a code snippet used to identify the web page element or elements that are to be affected by the styles.
- A **property** is the aspect of the element that is to be affected. For example, color, padding, margin, and background are some of the most commonly used CSS properties.
- A **value** is used to define a property. For example, the property color might be given the value of red like this: color: red;.
- The combination of a property and a value is called a **declaration**.
- In many cases, multiple **declarations** are applied to a **single selector**. A **declaration block** is the term used to refer to all of the declarations applied to a single selector.
- A single selector and the declaration block that follows it in combination are referred to as a ruleset.
- **At-rules** are similar to rulesets but begin with the @ sign rather than with a selector. The most common at-rule is the @media rule which is often used to create a block of CSS rules that are applied based on the size of the device viewing the web page.
- Both rulesets and at-rules are CSS statements.

# An Example of CSS Syntax

Let's use a block of CSS to clarify what each of these items is.

```
h1 {  
  color: red;  
  font-size: 3em;  
  text-decoration:  
underline;  
}
```

In this example, `h1` is the selector. The selector is followed by a declaration block that includes three declarations. Each declaration is separated from the next by a semicolon. The tabs and line breaks are optional but used by most developers to make the CSS code more human-readable.

By using `h1` as the selector, we are saying that every level 1 heading on the web page should follow the declarations contained in this ruleset.

The ruleset contains three declarations:

```
color:red;  
font-size: 3em;  
text-decoration: underline;
```

`color`, `font-size`, and `text-decoration` are all properties. There are literally [hundreds of CSS properties](#) you can use, but only [a few dozen are commonly used](#).

We applied the values `red`, `3em`, and `underline` to the properties we used. Each CSS property is defined to accept values formatted in a specific way.

For the `color` property we can either use a [color keyword](#) or a color formula in Hex, RGB, or HSL format. In this case, we used the color keyword `red`. There are a few dozen color keywords available in CSS3, but millions of colors can be accessed with the other color models.

We applied the value of `3em` to the property `font-size`. There are [a wide range of size units](#) we could have used including pixels, percentages, and more.

Finally, we added the value `underline` to the property `text-decoration`. We could have also used `overline` or `line-through` as values for `text-decoration`. In addition, CSS3 allows for the use of the line-styles solid, double, dotted, dashed, and wavy as well the specification of text-decoration colors. We could have applied all three values at once by using a declaration like this:

```
text-decoration: blue double underline;
```

That rule would cause the `h1` in our initial example to be underlined with a blue double line. The text itself would remain red as defined in our `color` property.

# Ways of Linking CSS Rules to an HTML Document

There are three ways of adding CSS rules to a web page:

1. **Inline styles:** Inline styles are applied to specific HTML elements. The HTML attribute `style` is used to define rules that only apply to that specific element. Here's a look at the syntax for writing inline styles.

```
<h1 style="color:red; padding:10px; text-decoration:underline;">Example Heading</h1>
```

That code would cause just that heading to render with red underlined text and 10 pixels of padding on all sides.

2. **Internal stylesheets:** An internal stylesheet is a block of CSS added to an HTML document head element. The style element is used between the opening and closing head tags, and all CSS declarations are added between the style tags.
3. **External stylesheets:** External stylesheets are documents containing nothing other than CSS statements. The rules defined in the document are linked to one or more HTML documents by using the link tag within the head element of the HTML document.

**THANK  
YOU!**