

# Chapter 1 — A Brief History and Creating Your First Page

## Introduction

### ***Learning Outcomes for this Chapter***

Upon completion of the reading of this chapter and assignments found at the end, a student should be able to:

- explain the HTML, as a base technology of the Web, is an ever evolving and changing language,
- recall that HTML was originally created in the late 1980's by a scientist, Tim Berners-Lee,
- differentiate the difference between null or void tags and tags with a required ending,
- implement basic tag nesting,
- recall and use the required parts of a properly formed HTML tag.

### ***What is HTML?***

HTML, or Hypertext Markup Language, is a computer language that is used to structure the content of a webpage. Hypertext refers to the system of linking one piece of information to another, and markup refers to the actual "marking up" of the information in order to organize it into something that is readable by a computer program. It is useful to think of learning HTML as learning an entirely new language. Becoming fluent in HTML will allow you to read, decipher, and write web-pages.

Just as with any living language, HTML has evolved over the years<sup>1</sup> and continues to change. HTML was originally created in 1989 by Tim Berners-Lee while at CERN, the European Organization for Nuclear Research, as a way to share linked documents in an easy to access manner.<sup>23</sup> The first website created, though simplistic to our modern standards, succeeded in achieving the intended outcome.<sup>4</sup>

### **A Brief History of HTML**

- HTML was created in 1989 by Tim Berners-Lee at CERN
- HTML— Published June 1993 as an Internet Engineering Task Force (IETF) working

1 <http://www.w3.org/MarkUp/>

2 <http://www.w3.org/People/Berners-Lee/>

3 <http://home.web.cern.ch/about>

4 <http://info.cern.ch/>



draft by Tim Berners-Lee and Daniel Connolly 5

- HTML 2.0 — IETF RFC1866 November 1995 by Berners-Lee and Connolly — this defined much of the structure that we still use today <sup>5</sup>
- HTML 3.2 – January 1997 — First standard published by the w3.org. Added tables and many of the features that IBM, Microsoft, Netscape and others had added to their browsers <sup>6</sup>
- HTML 4.01 – December 1999 — Standardized scripting and introduced style sheets to HTML <sup>7</sup>
- XHTML 1.0 – January 2000 — Reformulated HTML 4.0 into XML <sup>8</sup>
- XHTML 1.1 – November 2010 — Removed many of the HTML 3 and HTML 4 presentation tags and moved them to CSS <sup>8</sup>
- HTML5.2 — December 2017 — HTML5 "Recommendation" of the w3.org<sup>9</sup>.
- HTML Living Standard – May 2019 – W3C and WHATWG agree on a continually changing and evolving standard<sup>10</sup> that is called the Living Standard.

## What is CSS?

Though we will not be going over CSS until you have a solid understanding of basic HTML, it is useful to be able to know and recognize CSS when you see it. CSS stands for Cascading Style Sheet, and is a language that is used to describe the style of a website's appearance to the user. "Cascading" refers to the hierarchy of importance of the code, with the lower inputs taking precedence over those above. As with HTML, CSS has grown over time and has become the standard way to control the appearance of a webpage.<sup>10</sup>

The w3.org maintains the CSS standard.

## What is JavaScript?

Though we will not be discussing JavaScript in this volume, it is useful to know and understand the importance of its role in a webpage. JavaScript is a full featured object-oriented programming language which is used to control and manipulate HTML, CSS, and even the browser itself. JavaScript is a common part of a web browser, and you have most likely already

5 <http://www.rfc-editor.org/rfc/rfc1866.txt>

6 <http://www.w3.org/TR/REC-html32>

7 <http://www.w3.org/TR/html401/>

8 <http://www.w3.org/TR/xhtml11/>

9 <https://www.w3.org/TR/html52/>

10 <http://www.w3.org/Style/LieBos2e/history/Overview.html>



experienced using it at some point. Have you ever clicked on the check-out button during a purchase in an online store or filled out a form on a webpage? That is JavaScript at work!

Though it may seem complicated at first, it is helpful to think of the relationship between HTML, CSS, and JavaScript in simple categories. HTML organizes the information, CSS makes it look presentable in a web browser, and JavaScript makes the interactive portions work.

JavaScript and its standard is managed by ECMA, originally known as the European Computer Manufacturers Association. The standard ECMA-262 actually calls the language ECMAScript.<sup>11</sup>

## How do they work together?

An HTML document may contain HTML, CSS, and JavaScript code. It may also tell the browser to load external CSS and JavaScript files. These files may be on a local drive or a web server.

A website is usually made up of many HTML pages, CSS files, and JavaScript functions. When they are sent to the browser, by the server, additional dynamic content may be included into the pages. The web server may process forms submitted, generate custom content, and even query and update databases. There are dozens of dynamic extensions and languages. They include: PHP, Perl, Java, NODE.js, and others.

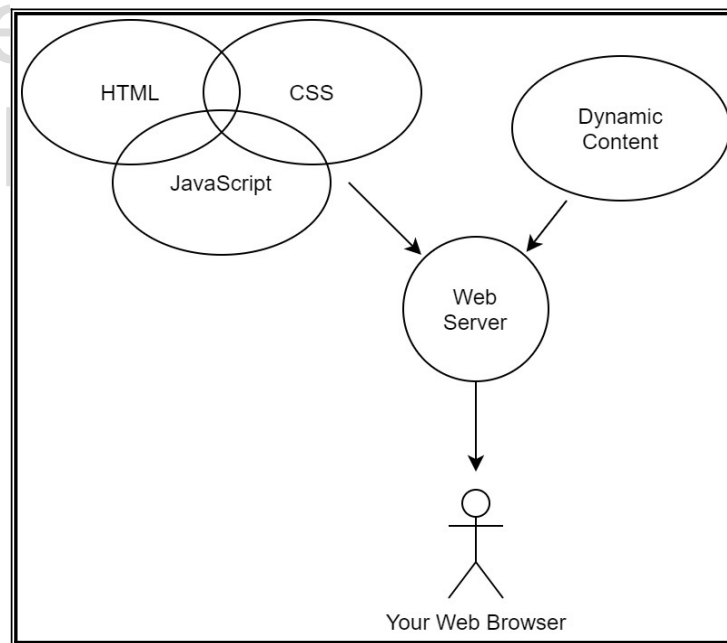


Illustration 1: What goes into a Website

<sup>11</sup> <https://www.ecma-international.org/publications/standards/Ecma-262.htm>



## What Makes up an HTML Document?

HTML is a "plain-text" document made up of tags with content and attributes. This means that an HTML document consists of human readable text that has minimal formatting mixed with tags that tell the browser what the text means and suggests how to display the text. A tag is the "Markup" in HTML and communicates to the web browser what the information means. For example, the `<p> . . . </p>` tag tells the browser that the text and markup following it should be treated as a paragraph.

## HTML Tags

A tag is easy to recognize as it begins with a `<` and ends with a `>`. These tag opening symbols are the "less-than" and "greater-than" symbols (also known as chevrons). If the tag contains content, such as the text in a `<p> . . . </p>` tag, it will be followed by an end tag that is marked with a `/`. For example, this is the proper way to mark up a paragraph:

```
<p>Example text.</p>
```

Some tags, also known as "empty tags" or "void tags", do not have an end tag. For example, the `<hr>` tag creates a Hard Rule (line) across the page. Since the `<hr>` tag does not contain anything, it does not require an end tag.

## HTML Attributes

In a typical HTML document, tags need to be differentiated from each other and will need to have special properties. We call those "attributes" of a tag. This is done by specifying the name of the attribute and then the value of that attribute in the opening tag. We should not need to specify them again in the closing of the tag. For instance, the empty tag `<img>` which displays images typically has two attributes:

```

```

In this example, the attribute **src** (short for "source") defines the file name or URL of the image, and **alt** (short for "alternate name") describes the image to users who cannot see the image. We will learn, in a future chapter, how to use the `<img>` tag.

In order for CSS and JavaScript to easily find specific tags, the **id** and **class** attributes can be added to any tag. An **id** is a unique identifier (ID) that may be assigned to one and only one tag on a page. The **class** attribute may be shared between several tags, and a tag may belong to multiple classes. The **id** or **class** name must begin with a letter and can then be made up of



letters, numbers, hyphens (-), underscores (\_), colons (:), and periods (.).<sup>12</sup> For example, observe this line:

```
<p id="gettysburgAddress">Four score and seven years ago...</p>
```

In this example, the tag `<p>...</p>` has an `id` of "gettysburgAddress", and can then be easily singled out in CSS and JavaScript. Classes function similarly, however it is important to remember that while a `class` may be shared between many tags in an HTML document, an `id` must only be used once.

## Understanding Our First HTML Document

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Chapter 1 Example 1</title>
  </head>
  <body>
    <h1>My first webpage & example.</h1>
    <p>A paragraph of text. Asa sareha tubedit nelurie cerietatesa
iferitam? Teta tas ho sihut cakek usu ties getef. Odemopi lazeme
hienie rorelop nefuva resit. Ibotum nedili ga yave me taru rud ge
yeg. Nogo nu sir fenene his utanieses ci. Tenedac mo co banevu boce
cenale tete. Rani nasat lec. Inodan hen usava reni refos. Ola buh
mo men: Ecuxode asemiewu tayopi sete oteyavep ganes liheyu
lecipef.</p>
  </body>
</html>
```

<sup>12</sup> <http://www.w3.org/TR/html401/types.html#type-name>



## My first web page & example.

A paragraph of text. Asa sareha tubedit nelurie cerietatesa iferitam? Teta tas ho sihut cakek usu ties getef. Odemopi lazeme hienie rorelop nefuva resit. Ibotum nedili ga yave me taru rud ge yeg. Nogo nu sir fenene his niger utanieses ci. Tenedac mo co banevu boce cenale tete. Rani nasat lec. Inodan hen usava reni refos. Ola buh mo men: Ecuxode asemiewu tayopi sete oteyavep inenat ganes liheyu lecipef.

*Illustration 2: My First Webpage*

Though the above may seem confusing to a first time viewer, the pieces can be broken down quite simply. Before you learn to write HTML it is important to be able to recognize the most common tags. Keep in mind the hierarchy of tags as you read. We often refer to hierarchy in a webpage as the "nesting" of tags, or how one tag is inside another tag.

- `<!DOCTYPE html>` — This will be the first line of any page that you create. It is used to communicate to the web browser which version of the HTML language it is written.<sup>13</sup> The current "Living Standard" is represented with the lowercase `html`. Previous standards have other specifiers.
- `<html> . . . </html>` — The "html" tag is the root of the page and tells the browser that it is reading an HTML document. All of the HTML document will reside between the open `<html>` and close `</html>` tags.
- The `<head> . . . </head>` of the document contains information (called metadata) that will, aside from the title, be mostly invisible to the viewer. It may be useful to think of the head as a brain that does most of the behind-the-scenes work. As with a human being, the head must come before the body. The standard requires that a HTML document must have one `<head> . . . </head>` directly inside the `<html> . . . </html>` tag. It can not be inside of another tag.
- `<meta charset="utf-8">` — This important tag communicates how the text is

<sup>13</sup> <http://www.w3.org/QA/Tips/Doctype>



encoded to the web browser. Utf-8 is short for Unicode 8 bit format and allows for 2,097,152 different characters from all languages of the world to be viewable in your HTML document, ensuring that your page reaches the maximum number of potential viewers.<sup>14</sup>

- `<title>...</title>` — The title of a webpage is a short, stand-alone description of what the page is about. The title will often display in the title bar of a web browser, the title in search engines, and the name of the page shortcut. The title is required on all HTML documents and must be in the `<head>...</head>`.<sup>15 16</sup>
- `<body>...</body>` — The "body" tag contains all the content of the webpage that will be displayed in the browser. Anything that you want your viewers to see should go inside this tag/section.  
The standard requires that a HTML document must have one `<body>...</body>` directly inside the `<html>...</html>` tag. It can not be inside of another tag.
- `<h1>...</h1>` — Is used to mark a section header of the highest rank.<sup>17</sup> There are six header levels (ranging from **h1** to **h6**). It is not valid for a header tag to be inside a paragraph.  
For example, you would wrap the page title in an `<h1>...</h1>` as it is the highest level in the hierarchical scale. Titles of subsections would be in `<h2>...</h2>` tags. Titles of sub-subsections would be in an `<h3>...</h3>` tag.
- `<p>...</p>` — The text or other HTML inside this tag will be grouped as a paragraph. By default, this element displays as a block across the full browser window, meaning that it will appear below other elements rather than beside another element. It will shrink or stretch to fit the entire width of the page regardless of size and words will automatically wrap.<sup>18</sup>

## Semantic Nature of Tags in HTML

At this point, we need to stress that HTML tags are used to describe the content of our pages, not to format the output on the browser. Semantics is the study of the meaning of language.<sup>19</sup> In HTML we use the tags to mark what something means. We use CSS to change how things appear.

14 <https://html.spec.whatwg.org/multipage/semantics.html#the-meta-element>

15 [http://www.unicode.org/faq/utf\\_bom.html](http://www.unicode.org/faq/utf_bom.html)

16 <http://en.wikipedia.org/wiki/UTF-8>

17 <https://html.spec.whatwg.org/multipage/sections.html#the-h1,-h2,-h3,-h4,-h5,-and-h6-elements>

18 <https://html.spec.whatwg.org/multipage/grouping-content.html#the-p-element>

19 <https://www.dictionary.com/browse/semantics>



## Nesting of Tags

Now that you understand the essential components of a webpage, we can begin talking about the hierarchy of tags, or more commonly referred to as "nesting". You can think of nesting like Russian Matryoshka dolls<sup>20</sup>, which contain smaller and smaller dolls that fit inside of each other.<sup>20</sup> The smallest one fits in to the next largest, and so on until you reach the largest doll, which would be the `<html> . . . </html>` tag. If a tag begins inside of another tag, it must end inside of that same tag.



Illustration 3: Russian Matryoshka dolls

For example, a paragraph inside of the body must be written as such:

```
<body>  
  <p>This is a paragraph</p>  
</body>
```

Note how the `<p> . . . </p>` is nested within the `<body> . . . </body>` tag. Improper nesting can lead to your web-pages rendering incorrectly in the same way that incorrect parentheses in a math problem will give incorrect answers. The web browser can only read and speak HTML. Make sure to structure your sentences, or code, correctly.

Here are some examples of **incorrectly nested** tags:

---

<sup>20</sup> [https://en.wikipedia.org/wiki/Matryoshka\\_doll](https://en.wikipedia.org/wiki/Matryoshka_doll)





```
<h1>The Best Ice-Cream  
<p>A paragraph about the best frozen goodness I have ever eaten.  
</h1>  
</p>
```

The example above is incorrect because the `<h1> . . . </h1>` ends after the `<p> . . . </p>` begins. You should not put any other tag inside a header tag.

```
<h1>The Best Ice-Cream  
<p>A paragraph about the best frozen goodness I have ever eaten.</p>  
</h1>
```

This example is incorrect because the `<p> . . . </p>` is nested inside of the header. The `<h1> . . . </h1>` tag is not a container, but should be a description of the tags that follow it.

Here is a correct example of nesting:

```
<h1>The Best Ice-Cream</h1>  
<p>A paragraph about the best frozen goodness I have ever eaten.</p>
```

Remember, proper nesting is one of the most important aspects of writing an HTML document.

## HTML Entities — Displaying Special Characters

In an HTML document `<`, `>`, and `&` have a special meaning. You should not include them in the text of your pages as they will potentially confuse the web browser.<sup>21</sup> However, if you need to add special symbols and Unicode characters to your page, you may do so using HTML Entities. The table below shows a few of the character entity mark-ups:

Entity	Character
<code>&amp;lt;</code>	<
<code>&amp;copy;</code>	©

<sup>21</sup> [http://www.tutorialspoint.com/html5/html5\\_entities.htm](http://www.tutorialspoint.com/html5/html5_entities.htm)



<code>&amp;gt;</code>	>
<code>&amp;euro;</code>	€
<code>&amp;amp;</code>	&
<code>&amp;pound;</code>	£
<code>&amp;quot;</code>	"

## ***Making Sure Your HTML is Correct***

Complete web efficiency comes from ensuring that your HTML document is using the language correctly. Though it is often difficult to spot your own mistakes, running your page through the web validator that the w3 provides will give you a quick and thorough report of any mistakes that you may have made. The free validator can be found at:

<http://validator.w3.org/>

## ***Base HTML Document***

To be consistent in your HTML pages, it is recommended that you start with a base document or template that has all the required elements on it. Below you will find a very simple one of those. As you learn more tags and uses, your base document may have other tags and semantic (Chapter 3) blocks added.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>title</title>
  </head>
  <body>
    <!-- page content goes here -->
  </body>
</html>
```

The base document does also shows the HTML comment statement `<!-- message -->`. A comment statement can exist anywhere inside the HTML document and is totally ignored by the browser and search engines. It is often used as a place for developers to put their name and a



description of the page. Comments are also used, like above, to remind the developer that something is missing and where it need to be added.

### ***Advanced Topics for Exploration***

- The format of HTML4 and XHTML1.1 base pages
- Numeric representation of Entities and reference to extended list of them <sup>22</sup>

### ***Vocabulary***

- CSS
- Cascading Style Sheet
- comment
- DOCTYPE
- HTML
- HTML Entities
- Hypertext Markup Language
- JavaScript attribute
- body
- head
- id
- meta
- p
- tag
- title
- void/empty tag

---

22 <http://dev.w3.org/html5/html-author/charref>



## Exercises

### Word Search

```
S t i J a t d a m e t a  
p t e a e t t T t t t a  
a H p v n D i i t a r t  
r b d a t O e S t e g t  
a a b S i C l Y C l r r  
g t d c t T e v s S e i  
r a i r i Y e h i i S b  
a P d i e P t m e b C u  
p s d p s E v t p a v t  
h g m t e E t o a t d e  
L r H T M L M e i m y a  
h y P b o d y c l d r b
```

CSS, DOCTYPE, HTML, JavaScript, attribute, body, empty, entities, head, id, meta, paragraph, tag, title, void

### Problems

1. Using the links in the text as a starting place, write a 1-2 page history of the development of e-commerce of the Web. As part of your research find important dates in the history of e-commerce that you have used. Be sure to include links to your sources in your text.
2. The Extensible Markup Language (XML) is a way to mark up data and other information, it has a very similar structure to HTML. Write a 1-2 page answer to the question "Is HTML a subset (or complies with) the XML standard?" Be sure to include links to your sources in your text.
3. Use the base HTML document (above) and create a page that showcases all six levels of headings (<h1>...</h1> to <h6>...</h6>). Be sure to validate your HTML using the validation service at <http://validator.w3.org/>.
4. Start with the base HTML document (above) and create a simple page about you or a famous person from history. Place your name in an <h1>...</h1> at the top of the visible page. Then put a couple of <h2>...</h2> tags, with topics, below the name and a paragraph or two after each of the second level headings. Be sure to include on the page the source used for your material. Validate your HTML using the validation service at <http://validator.w3.org/>.

