Chapter 12 Introduction to HTML and Website Development

Shambhavi Roy
Clinton Daniel
University of South Florida
Manish Agrawal
University of South Florida

Follow this and additional works at: https://digitalcommons.usf.edu/dit_tb_eng

Scholar Commons Citation

This Book Chapter is brought to you for free and open access by the The Modernization of Digital Information Technology at Digital Commons @ University of South Florida. It has been accepted for inclusion in FUNDAMENTALS OF INFORMATION TECHNOLOGY: Textbook – English by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact digitalcommons@usf.edu.
## CHAPTER CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>248</td>
</tr>
<tr>
<td>What is HTML?</td>
<td>249</td>
</tr>
<tr>
<td>Structure of an HTML Page</td>
<td>251</td>
</tr>
<tr>
<td>Elements in HTML</td>
<td>252</td>
</tr>
<tr>
<td>Adding Links</td>
<td>253</td>
</tr>
<tr>
<td>Adding Images</td>
<td>254</td>
</tr>
<tr>
<td>Creating Lists</td>
<td>255</td>
</tr>
<tr>
<td>Creating Tables</td>
<td>255</td>
</tr>
<tr>
<td>Creating Forms</td>
<td>256</td>
</tr>
<tr>
<td>Cascading Style Sheets (CSS)</td>
<td>260</td>
</tr>
<tr>
<td>Website Design and Storyboarding</td>
<td>261</td>
</tr>
<tr>
<td>Structure of a Website</td>
<td>263</td>
</tr>
<tr>
<td>Hosting a Website</td>
<td>265</td>
</tr>
<tr>
<td>Website Editors</td>
<td>266</td>
</tr>
<tr>
<td>HTML Extensions</td>
<td>268</td>
</tr>
<tr>
<td>Dynamic HTML (DHTML)</td>
<td>268</td>
</tr>
<tr>
<td>Extensible Markup Language (XML)</td>
<td>269</td>
</tr>
<tr>
<td>XML and JSON</td>
<td>272</td>
</tr>
<tr>
<td>Chapter Terms and Definitions</td>
<td>274</td>
</tr>
<tr>
<td>Chapter Case: Wrenley Saves the Pets!</td>
<td>276</td>
</tr>
</tbody>
</table>
It may not always be profitable at first for businesses to be online, but it is certainly going to be unprofitable not to be online.

—Esther Dyson, investor, philanthropist

Overview

The Internet has been around since the early 1970s. In the early years, the Internet only supported plain text and using the Internet required users to learn the precise syntax of several computer commands. For example, to send an email to user jane@example.com required users to type the command: mail -s "mail subject" jane@example.com <<< "Mail body." Most users found this too challenging and the only users on the early Internet were computer science researchers at universities.

This changed in 1989 when Sir Tim Berners Lee proposed and demonstrated an information management system in 1989 based on hyperlinks. Users could click on links to navigate to other pages. By 1993, Tim Berners Lee had specified the first version of HTML, which made information sharing as easy as clicking a link on a page. During December 1992–March 1993, Marc Andreesen (a student) and Eric Bina (a graduate) of the University of Illinois at Urbana-Champaign (UIUC), both working at the National Center for Supercomputing Applications (NCSA) at Urbana-Champaign, built the Mosaic web browser. They also added the “img” tag to html to handle images and released the web browser to the Internet. The browser soon became immensely popular and acquired the status of a killer application that popularized the Internet. Sites like Yahoo offered directories for users to find content on the emerging World Wide Web.

Since then, the Internet has become increasingly user friendly as HTML has improved, adding support for images and videos. This has made the Internet and World Wide Web a powerful force for

This overview is from CERN, the birthplace of HTML and HTTP https://home.cern/science/computing/birth-web/short-history-web (accessed June 2023).


NCSA also published one of the first HTML tutorials. It is a simple document, perhaps the best introductory documents to learn HTML. An entire generation of enthusiasts, including the authors of this book learned HTML from this document. It is archived at http://seismo.berkeley.edu/~rallen/resources/html/HTMLPrimerAll.html (accessed June 2023).


for information sharing and commerce, no longer limited to specialized users familiar with exotic computer commands. Today over 5 billion people, over 60% of the world’s population, use the Internet, primarily because HTML and related technologies have made it easy, inexpensive, and meaningful for anyone anywhere in the world to access the web.

This chapter introduces HTML, the primary end-user technology used on the web.

**What is HTML?**

HTML (Hypertext Markup Language) is the standard markup language used for creating webpages and web applications. A markup language is a computer language that is used to apply layout and formatting conventions to a text document. The HTML markup language makes text on webpages more interactive and dynamic. HTML allows us to specify the placement of images, tables, links, etc. on webpages using simple text descriptions. HTML provides the structure and content of a webpage, specifying how text, images, hyperlinks, tables, and other media should be displayed on a browser.

When you visit a website, say http://www.google.com, returned is an HTML page (Figure 198). Your browser can understand the page specifications defined in HTML and is able to render it as a webpage that you can see and interact with.

Hypertext means “text within text.” In the context of the World Wide Web, hypertext refers to text with a link to another page within it. Hypertext is the Internet’s way to link two or more webpages (HTML documents) with each other.

Figure 198 — The Google homepage is an example of an HTML page.

---

163 The term “hypertext” has a Florida association. It was coined by Ted Nelson in 1965. In 1963, when he got the idea, Nelson was documenting research at the John C. Lilly Dolphin lab in Miami (https://web.archive.org/web/20150506080854/http://hyperland.com/TNvita) (accessed June 2023). Hypertext was defined as “A body of written or pictorial material interconnected in such a complex way that it could not conveniently be presented or represented on paper.” Nelson could foresee that “[hypertext] could grow indefinitely, gradually including more and more of the world’s written knowledge.” Hypertext was seen as a file system that, unlike file systems such as those used in Windows and Macs, “[allowed] file contents to be arranged any-which-way, and in any number of ways at once.” Unfortunately, Nelson’s lack of technical expertise prevented him from implementing the idea. Modern hyperlinks are seen as one-sided versions of Nelson’s proposed two-sided links. Ted Nelson’s hypertext idea may have greatly influenced the design of modern personal computers, https://archive.nytimes.com/bits.blogs.nytimes.com/2007/12/11/when-big-blue-got-a-glimpse-of-the-future/ (accessed June 2023).

To see the HTML code that defines the page, you can right click on your webpage (on most browsers such as Google Chrome or Mozilla's Firefox) and choose “View Page Source.” This will bring up text specification of that webpage. (Figure 199).

HTML has been an Internet standard since 1993, and all common browsers (e.g., Google Chrome, Apple Safari, Microsoft Internet Explorer, and Mozilla's Firefox) understand HTML. So, when you visit any page written in HTML, say http://www.google.com, on any browser, they all receive the same text, and the page looks the same on all these browsers. For example, the Google home page looks the same on all browsers.165

HTML is a crucial part of the web and forms the foundation of all webpages. HTML uses tags to define elements. Tags can have attributes to further describe elements. These tags and attributes are used to create the visual and semantic structure of a webpage. Though HTML is primarily designed for machines to interpret, as you will see in this chapter, its simplicity makes it easy even for humans to understand the content.

165 Browsers introduce minor individual tweaks, for example in their choice of default fonts. Commercial websites are tested for performance on all browsers to account for these tweaks. But for the purposes of this book, we treat all browsers as identical.
Virtually all information and tools are presented to end users as webpages. You access your school resources on the web, check email on the web, track bank accounts on the web, your teachers enter your grades on the web, and so on. All these webpages are created using HTML. Therefore, HTML is a foundational technology that all modern professionals will find useful.

In this chapter, we will review the basic elements you need to know to create HTML pages and review the editors that make it easy to build webpages. If you find this chapter interesting and would like to learn more about HTML and creating webpages, you should also take advantage of free online tutorials like W3Schools to learn HTML in more detail, as it is not possible to cover all tags and attributes in a chapter.

**Structure of an HTML Page**

HTML is a simple language. It essentially encloses text within tags to tell the browser how to render the enclosed text. For example, the browser will render `<b>this is bold text</b>` as **this is bold text**. `<i>this is italicized text</i>` will be rendered as *this is italicized text*. You don’t need fancy editors to create an HTML page. You could create an HTML page by typing HTML commands on a notepad and saving the file as .html (a file with an html extension). All HTML pages follow the same basic structure. They contain a **DOCTYPE declaration** at the top, followed by an HTML section. The HTML section comprises of a HEAD and BODY. Figure 200 shows a basic example.

The contents of **Figure 200** are interpreted by the browser as follows:

**DOCTYPE Declaration**: This is the first line of an HTML page, and it tells the browser what kind of document to expect, particularly, which version of HTML.

**HTML tag**: All code in HTML is declared using tags. A tag is a special keyword that is enclosed by `<>`. The HTML tag defines the start and end of an HTML document. Everything in the HTML document is nested within the opening and closing HTML tags, i.e., between `<html>` and `</html>`. Tags are closed with the “/” symbol in the tag, Thus, `</html>` indicates the end of the section that began with the `<html>` tag.

**Head tag**: The head tag contains information about the HTML document that is mostly not displayed to the user. This section includes the title of the page, any information about the page such as author information and version history (this is often called meta information), information about the look and feel (called stylesheets), and information about user interactions (specified using a language called JavaScript). You can also include additional description and relevant keywords about your webpage within the head tag, so search engines can more easily identify your page. The only content within the head that is displayed to the user is information within the title tag, which is displayed as the title of your browser tab.

**Body tag:** The body tag contains all the content that is displayed to the user. This includes text, images, links, and other HTML elements. When you build your HTML page, you will spend most of your time within the body. This is also the section we discuss the most in this chapter.

**Elements in HTML**

Like most electronic documents, webpages have text, images, tables, and links to other pages. The text may be bold, italicized, and use different font sizes. You might see ideas listed as bullet points or numbered lists. You may also see forms, with different input elements for you to fill in. You can represent each of these elements with HTML tags and build a complex HTML page that asks users to enter data and then sends the inputted data to a server for further action.

Here are a few essential tags to get you started.

- The **paragraph** `<p>` tag is used to define a block of text as a paragraph. The text within the `<p>` tag is displayed as a separate paragraph, with a line break before and after it (Figure 201). The text within the paragraph tag can include any type of content, such as text, images, links, or other HTML elements.

```
<!DOCTYPE html>
<html>
<body>
  <p>This is my new paragraph.<br>
     It is a collection of text that can be styled</p>
</body>
</html>
```

FIGURE 201 — Using the `<p>` tag creates a paragraph.

When creating a paragraph, you could use attributes to further specify the kind of paragraph you want. To create a paragraph with text in red color, try this:

```html
<p style="color:red;"> This is a red paragraph.</p>
```

- **Heading tags:** Headings are the essential mechanism to create structure within pages. HTML supports headings with six levels of headings, ranging from `<h1>` to `<h6>`. The `<h1>` tag represents the main heading on a page and is typically used for the page title, while the other headings `<h2>` to `<h6>` are used for subheadings. It is best practice to not skip heading levels, i.e., subsections of `<h1>` should use `<h2>`, not `<h3>` or `<h4>` etc. Figure 202 shows an example.

---

167 Headings have another use, they indicate the organization of the page to assistive technologies. You can read about this at [https://www.w3.org/WAI/tutorials/page-structure/headings/](https://www.w3.org/WAI/tutorials/page-structure/headings/) (accessed June 2023).
• **Bold** `<b>` tag is used to bold text.

• The **underline** `<u>` tag is used to underline text.

• The delete `<del>` or `<strike>` tag is used to visually strikethrough text (Figure 203).

• The **subscript** `<sub>` and **superscript** `<sup>` tags are used to display text as subscript or superscript (Figure 204). These tags can be helpful to display scientific formulas.

**Adding Links**

Hyperlinks are the unique distinguishing feature of HTML. Hyperlinks allow you to click on a link on a page and immediately navigate to another page that could be hosted anywhere in the world. The `<a>` tag stands for “anchor” and allows you to add hyperlinks to an HTML page.
The `<a>` tag has a required attribute to specify the target page. The target page is specified using the `href` attribute, giving the URL of the page or the resource you want to link. For example, to create a link to the homepage of the New York Times website, you can use the code shown in Figure 205:

```
<!DOCTYPE html>
<html>
<body>


</body>
</html>
```

Clicking on The New York Times will take you to the newspaper's website.

FIGURE 205 — An anchor tag allows users to link web addresses.

When this code is rendered on a web browser, the user can click on “The New York Times” to go to the website.

**Adding Images**

You can add images to your HTML page with an `<img>` tag. Like the `<a>` tag, the `<img>` tag also has a required attribute, that specified the image file to be displayed. You need to have the source (`src`) attribute to give the location of the image, so the browser can get the image and display it on the webpage. Optionally, you can also have other attributes like width and height parameters to tell the browser how large the image should be on the page (Figure 206).

Images draw the attention of visitors. So, not surprisingly, company logos, background images, and advertisements are carefully and tastefully designed to draw the attention of visitors and convey the right message about the site to them.

```
<html>
<body>

<img src="laptops_in_school.jpg" alt="Laptops in school" style="width:50%;height:50%;">

</body>
</html>
```

Image Size
Here we use the style attribute to specify the width and height of an image.

FIGURE 206 — Users can add images and adjust the image size in HTML.
Creating Lists

To create lists in HTML, such as a list of things to buy, places to visit, or people to invite, HTML offers two options—bulleted lists and numbered lists. You have the `<ul>` tag for unordered (bulleted) lists and the `<ol>` tag for ordered (numbered) lists. You can indicate the list items with the `<li>` tag and nest lists within lists (Figure 207).

```
<html>
<body>

<h2>A Nested List</h2>
<p>Movies to watch with family</p>
<ul>
  <li>Watch with Family
    <ul>
      <li>Gray Hound</li>
      <li>Saving Private Ryan</li>
    </ul>
  </li>
  <li>Watch with Friends
    <ul>
      <li>Lord of the rings</li>
      <li>Game of Thrones</li>
    </ul>
  </li>
  <li>Watch with Friends
    <ul>
      <li>13 Assassins</li>
      <li>Zombieland</li>
    </ul>
  </li>
</ul>

</body>
</html>
```

![A Nested List](image)

**FIGURE 207** — Nested lists using bulleted or numbered lists in HTML make it easy for users to identify sections.

Creating Tables

To display structured data, such as price lists, club memberships, class enrollments etc., HTML supports tables. Tables are created using the `<table>` tag. Tables have some additional necessary elements to specify rows and cells. The `<tr>` and `<td>` tags define the rows and cells of the table, respectively. You can also use the `<th>` tag to define table headers, which are typically bolded and centered. A `<table>` will have one or many `<tr>` tags, indicating the rows in the table. Each `<tr>` in turn encloses many `<td>` tags, indicating the cells within the row. The `<td>` tag represents a cell and encloses the contents of the cell (Figure 208).

```
<table>
  <tr>
    <th>Watch with Family</th>
    <td>Gray Hound</td>
    <td>Saving Private Ryan</td>
  </tr>
  <tr>
    <th>Watch with Friends</th>
    <td>Lord of the rings</td>
    <td>Game of Thrones</td>
  </tr>
  <tr>
    <th>Watch with Friends</th>
    <td>13 Assassins</td>
    <td>Zombieland</td>
  </tr>
</table>
```
Creating Forms

Forms are the principal mechanism to get user inputs on webpages. Creating forms on webpages is exciting because it allows your users to interact with the webpage and submit information for processing. Almost all your favorite webpages have forms where data can be entered. Through forms,
users can enter text, numbers, dates, and upload files. Users can also press buttons, tick checkboxes, and choose radio buttons. The Google.com homepage, probably the most famous webpage in the world, is just a simple form with an input field where you can enter your search query. The page is essentially a form with a text input element and two buttons (Figure 209).  

Forms are created using HTML code. Later in this chapter, we will see how they can be styled and customized using CSS stylesheets. When a user submits a form on an HTML page, the data is sent to a server for processing or storage in a database. For example, when you search for “Movies to watch” on Google.com, your input into the form is taken and sent to a Google server. The server then responds with the answer to your search query.

Forms are used in a variety of contexts. Common forms include login and registration forms, search forms, contact forms, feedback forms, and e-commerce checkout forms. They are an essential part of web development and enable webpages to collect data, provide a personalized user experience, and perform various actions based on user input.

Forms are created using the <form> tag. Every form also has multiple <input> tags to specify the kind of input you want from the user. These <input> elements can be text, number, file, or image. Each of these can be specified by choosing the right value for the “type” attribute of the <input> tag. The data collected in a form is sent to the server for processing using the action attribute of the form tag.  

In the example in Figure 210, we have a simple form to collect the first name and last name of the user. When the “Submit” button is pressed after entering the first and last names, the form data will be sent to the processor located at server_submit.php. The processor will respond after processing the name, for example by saving it in the database, and customizing the response with the name, etc.  

---

168 The ‘I’m feeling lucky’ button on Google’s home page has an interesting history. Most users do not use it, but it is a founder favorite. You can read about it at https://www.howtogeek.com/847170/googles-im-feeling-lucky-explained/ (accessed June 2023) and https://content.time.com/time/business/article/0,8599,685801,00.html (accessed June 2023).

169 Forms generally use the capability of HTTP (hypertext transfer protocol) to upload data from the browser to the server. This is specified using the POST method attribute of the FORM tag. We ignore this detail in this book.

170 The ids (unique Id) of elements you see in the form help the processor associate inputs with the fields. These ids should, therefore, be unique for forms to work as expected.
To create a multi-line input element, use the tag “textarea.” You can also specify how wide and tall this text box should be using “rows” and “cols” attributes.

To create a dropdown that allows users to select from a list of things, you can use the select tag (Figure 211).

There are additional form input types to handle common input scenarios. A radio button is a great choice if you want to give users multiple choices, but you want them to pick only one option. The concept of a radio button dates to radios that had mechanical buttons to select a station. You could only select one button at a time. If you pressed another button, then the previously pressed button would pop out automatically. This concept has carried over to HTML forms and such selectors are called radio buttons, tracing their lineage. So, even though you may see many options, selecting a new option will deselect your previous choice. Figure 212 shows an example.

HTML forms are similar to car radios. Both can have multiple buttons/fields to receive input.
FIGURE 212 — Similar to actual radio buttons, even HTML radio buttons can only allow one input at a time, clearing the previous options.

The “Checkbox” element in HTML derives its name from paper forms with checkboxes. You can select one or many checkboxes. Similarly, creating a “Checkbox” element in HTML will allow your users to select one or more choices (Figure 213).

FIGURE 213 — Users can create checklists in HTML by using the checkbox element.

If you want users to enter a date, you may want to use the “Date” element. Your browser will recognize this element and display a calendar (Figure 214). Once a date is selected, the browser updates the field with the selected date.
Cascading Style Sheets (CSS)

The HTML we have created thus far is functional, but bland. We now look at CSS, which is the standard way to make webpages stand out with color and design. CSS stands for Cascading Style Sheets and is used to describe how HTML elements are to be displayed on screen, paper, phones, iPads, or in other media. The design of CSS saves the developer time by allowing the layout of multiple webpages to be controlled all at once. In a few examples we have specified the height and width of the image. See Figure 206. While this was convenient for the example, imagine trying to decide how every element should look across multiple pages on your website. When you add tags like <font> and <color> attributes to HTML, you will have a nightmarish experience trying to remember the color and other attributes of other similar elements on your page. Your webpage may end up looking inconsistent if you are not careful. Besides, the HTML file will be huge and messy. It gets worse. If you decide to change the colors or font styles on your site, you would have to update the look and feel of every such element on your page.

To avoid these problems, the best practice is for developers to define all their styles in a CSS file that they then link to their HTML documents. In this way, every “Submit” button on all webpages across your site will use the same size and color and every header will be of the same font size and style.

To see how CSS works, let’s begin by trying to add some basic styling effects to our HTML document within the style tag as shown in Figure 215.
W3Schools is a great resource to learn about styles.\textsuperscript{171} You can tweak more aspects about the look and feel of your webpage than you can imagine. Once you develop some comfort with CSS, the next step to improve your productivity as a web developer is to create a separate CSS file (Figure 216) and then link it to your HTML file as shown in line 4 of Figure 217.

```
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}
```

**FIGURE 216 — CSS files include styles and attributes that affect normal HTML elements.**  
**FIGURE 217 — Users can link stylesheets to HTML to reduce clutter in the HTML text.**

### Website Design and Storyboarding

Websites are collections of individual webpages that accomplish a specific purpose. For example, the website of your school district, say https://www.hillsboroughschools.org/ or your school, say https://kingib.org/ has several pages that provide all the information and interaction students and parents expect from the school or school district.

When you visit a website (say, www.google.com), there are servers (owned and managed by a service provider) waiting for you to reach out to them. To support all their users and their requests, websites have a collection of webpages that are connected to one another through weblinks. This way, when you go from one Google webpage to another, you have a seamless experience.

For example, when you search for a book on Google, you see a collection of results. If you click on a Google sponsored ad after reviewing the results, another result page opens. The entire flow is managed by multiple Google webpages and application programs that handle these user requests.

To accomplish this, websites should be coherent, easy to navigate, and accomplish what you set out to do. The same need for coherence and ease of use applies to much larger sites where you write your blog, sell your products, or promote your company. Building a website is a resource-intensive exercise. You must plan, build, and deploy your website. Just like you wouldn’t start constructing a house without having a plan, you don’t want to build a website on the fly. Therefore, once you learn the elements of webpages including HTML and CSS and are ready to build a website, it is a good idea to start with storyboards.

\textsuperscript{171} The W3Schools CSS tutorial is at https://www.w3schools.com/css/default.asp (accessed June 2023).
Storyboards are visual representations of creative projects.\textsuperscript{172} When used with websites, storyboards help all members of the development team identify all the users of a website and their expected interactions.\textsuperscript{173} Once storyboards are developed, they are usually communicated using wireframes. Wireframes are illustrations of the webpage that focus on space allocation for the elements of the page and their intended behaviors.\textsuperscript{174} Wireframes are intended to be simple and disposable and typically do not include any styling, color, or graphics.

Storyboards and wireframes are both important tools in the design and development process of a website, but they serve different purposes and are used in different contexts. Storyboards are about the storytelling and visual components and wireframes focus on the user interface and interaction aspects of a website. For large scale projects, you should first do storyboarding before moving on to wireframing tools.

While storyboarding evolved from film and drama, the concept has found a home in website development. It is useful to understand the high-level overview and flow of the webpages and visualize the user experience. Once you and your team agree about the flow, you can then move to wireframing the design.

A wireframe outlines the structure and content of a website without including design elements such as colors, typography, or images. Wireframes are created during the planning phase to help designers and developers identify and address potential design and usability issues.

Wireframes typically include boxes and lines to represent the placement of different elements on the page, such as headers, navigation menus, content sections, and calls to action. Balsamiq\textsuperscript{175} is a popular wireframing software that lets you quickly wireframe your webpages and then share it with your team and customers to get their feedback before investing development effort to create the website (Figure 218). Wireframing tools such as Balsamiq offer templates to pick, drag, and drop different webpage elements and put them on the screen to quickly develop wireframes. You can use wireframes to design browser-based webpages or for mobile apps like iOS or Android. Building wireframes to refine application concepts is an excellent, low-effort, no-cost way to test new application ideas.

\begin{itemize}
\item \textsuperscript{172} You can read about the origins of storyboards, and how Walt Disney and his team originated the concept in 1933 to convey the story of “Three Little Pigs” at \url{https://www.instructionaldesign.org/concepts/storyboarding/} (accessed June 2023).
\item \textsuperscript{173} A good description of storyboards and their use is at \url{https://www.capterra.com/resources/what-is-a-storyboard-in-web-design/} (accessed June 2023).
\item \textsuperscript{174} The US federal government’ s usability website has a nice introduction to wireframing at \url{https://www.usability.gov/how-to-and-tools/methods/wireframing.html} (accessed June 2023).
\item \textsuperscript{175} \url{https://balsamiq.com/} has an excellent tutorial on wireframing and user interface design at \url{https://balsamiq.com/learn/courses/intro-to-ui-design/} (accessed June 2023).
\end{itemize}
Another popular wireframing tool, Figma, was acquired by Adobe in September 2022 for $20 billion.\footnote{You can read the announcement at \url{https://news.adobe.com/news/news-details/2022/Adobe-to-Acquire-Figma/default.aspx} (accessed June 2023).} Figma not only lets you wireframe and collaborate with colleagues but also automatically converts your wireframe design into a working HTML website that you can later deploy to your webhost.

\section*{Structure of a Website}

Once you learn HTML to build webpages and create storyboards and wireframes for your idea, you may be interested in creating your website so it can be viewed by anyone in the world. We now provide an overview of the elements of a working website.

If you want to create a website, your website admin (or you) will need to have a \textit{webserver} (e.g., Apache webserver\footnote{The Apache webserver, which traces its origins to the first webserver built by Tim Berners Lee at CERN, is extremely robust and easy to install. If you have access to a personal computer, installing the Apache webserver and becoming familiar with its use is delightful learning. You can learn more and download Apache from \url{https://httpd.apache.org/} (accessed June 2023).}) to host the website.\footnote{You can read more details online, including at GoDaddy's page, \url{https://www.godaddy.com/garage/how-to-build-a-website-with-html-and-css/} (accessed June 2023).} You will need to save your HTML pages and other supporting files in an organized folder structure on the webserver. We will now discuss the basic structure of a website so you can understand how all the elements of a site are integrated together.

It is a good idea to include the folder structure of your website in your wireframe, so it is easy to see how it is organized on your webserver. Figure 219 shows an example of the folder hierarchy of a website and how it ties to the different sections of the site together.
It is useful to maintain a consistent organization for the files within each folder. Figure 220 shows a popular organization structure for the files in a folder. The starting point of your application or website is typically called the “Root” folder of your site. A file called “index.html” is displayed by default if someone visits your website and does not specify a specific file name.179

Organizing folders for Images, Fonts, Common libraries, JavaScript, and Stylesheets (Figure 220) makes it easier to manage and maintain files. Search engines and users will also be able to navigate the website easily. Additionally, many websites also use a Content Management System (CMS) that provides a graphical interface to organize and manage the files and contents of websites.

You can have all your HTML pages in a separate folder, or you can choose to divide your website into different functional areas and have a separate webpage for each topic within it. In Figure 221, you can see how the submenus of a website correspond to similarly named folders on the server. For example, if a user visits your home page, the server will show the root-level index.html file. However, if the user clicks Submenu 1, then the Index.html within the Submenu 1 folder will be displayed.

FIGURE 221 — Concept and folder diagrams help users plan how to develop their websites and content.

Hosting a Website

Hosting a website means making it accessible to the public over the Internet by storing the website's files and data on a webserver. A webserver is a computer that is connected to the Internet and is designed to store and deliver webpages to users who request them through their browsers.

When you host a website, you essentially rent space on a webserver from a hosting company or provider. This allows your website to be stored on the webserver and made available for access over the Internet. The web hosting company also provides a range of services, such as server maintenance, security, and technical support, to ensure that your website is always accessible and runs smoothly. You'll need to upload your files to the webserver and configure your website settings so users can find your site on the web.¹⁸⁰

¹⁸⁰ The primary configuration is to acquire a domain name (such as www.example.com) and point it to your website. Hosting service providers take care of this detail, so we do not discuss this in more detail in this book. For details, please see https://www.hivelocity.net/kb/dns-configuration-everything-you-need-to-know/ (accessed June 2023). To see what the setup at a hosting service provider looks like, please see GoDaddy’s details at https://www.godaddy.com/garage/configuring-and-working-with-domains-dns/ (accessed June 2023).
Once your website is hosted, users can access it by typing your domain name (e.g., www.cyberflorida.org) into their web browsers. This sends a request to the webserver hosting your website. The webserver then responds by sending the requested webpages back to the user’s web browser, which displays them on their screen.

### Hosting from Your Home Computer

You can actually host your site from your own home computer. You’ll have to get a permanent IP address from your Internet service provider, purchase a domain name (website name) from a DNS registry, and connect the website name to your computer’s IP address. You’ll also need to ensure your computer can handle the load, is protected, is always turned on, and is being backed up regularly.

As you can see, this adds a lot of complexity and responsibility. Therefore, the alternative of taking advantage of web-hosting providers is very popular.

There are many popular web hosting sites available, each with its own features, pricing, and capabilities. Some of the most popular web hosting sites are GoDaddy, Bluehost, SiteGround, HostGator, DreamHost, A2 Hosting, InMotion Hosting, and Wix. It’s important to research and compare different options to find the one that best meets your needs and budget.

### Website Editors

There are multiple ways to build HTML pages. One way is to code it from scratch using the HTML elements like `<table>`, `<text>`, `<img>`, and others. Another way is to use a wireframe tool like Figma and then use the tool’s conversion feature to convert the wireframe into HTML code.

Another popular way is to use a WYSIWYG (Pronounced WIZ-EEE-WIG) editor. WYSIWYG is a popular industry acronym for “What You See Is What You Get.” In the context of HTML editors, a WYSIWYG editor is a software program that allows users to create and edit HTML documents without having to know the HTML code. With a WYSIWYG editor, users drag and drop website elements such as images and tables on a drawing canvas, and see the webpage they are creating as it will appear in a browser without having to manually code the HTML.

WYSIWYG HTML editors make web technologies accessible to everyone. They are popular among non-technical users, such as bloggers, content creators, and small business owners, who may not have the technical expertise to create webpages using HTML code. Some popular free WYSIWYG

---

182 You can check out plans at [https://www.hostgator.com/web-hosting](https://www.hostgator.com/web-hosting) (accessed June 2023). HostGator started in a dorm room at Florida Atlantic University, so we are biased towards them.
184 You can find several resources and comparisons online, e.g., [https://geekflare.com/best-static-site-hosting-platform/](https://geekflare.com/best-static-site-hosting-platform/) (accessed June 2023).
HTML editors\(^{186}\) include Brackets from Adobe\(^{187}\) Phoenix code editor from the Brackets developer community,\(^{188}\) and SeaMonkey from the Mozilla community.\(^{189}\) There are also commercial WYSIWYG editors, including Adobe's Dreamweaver.\(^{190}\)

**Website hosting** providers like Wix.com can also give you access to website builders and editors. The website builder is typically a WYSIWYG platform, where you can drag and drop elements on to the page. When you are ready, you can click the “Publish” button to deploy your website without writing or editing a line of code.

Google Sites (www.sites.google.com) is another website hosting provider that has its own WYSIWYG webpage builder (Figure 222).

The capabilities of HTML editors are evolving rapidly. While the editors share most of the same basic capabilities, they can also offer additional capabilities depending upon the needs of their target audience, pricing structure, etc. For example, commercial editors may offer advanced features like a library to maintain your common themes, icons, a collaboration framework (so you can get your work

---

186 There are other choices and reviews available online, including for example in the article at https://www.thoughtco.com/best-windows-wysiwyg-editors-3471324 (accessed June 2023).


188 The Phoenix editor is entirely online and can be accessed at https://phcode.dev/ (accessed June 2023).

189 You can download SeaMonkey from https://www.seamonkey-project.org/ (accessed June 2023). SeaMonkey continues the all-in-one Internet application suite concept popularized by the Netscape browser, which all authors of this book used in the early days of the Internet.

approved before it gets published), and a built-in version control system (so you can go back to any previous version). Check out the capabilities and pricing models of the different editors to find one that best fits your needs.

Learning to build webpages and websites is a great way to enter the fascinating world of modern information technologies. You can continue your journey by following tutorials available online freely. The Mozilla Developer Network (MDN) is a great resource to learn the technologies introduced in this chapter. W3Schools is also a great resource.

**HTML Extensions**

We close this chapter by introducing some popular related technologies that power the web and commerce today. The popularity of the web has led to several innovations in web design, collectively called DHTML. The concept of tags has also been extended to XML, which generalizes publishing and even allows machines to directly exchange information with each other.

**Dynamic HTML (DHTML)**

As you begin developing your own webpages, you will notice that popular websites have capabilities beyond those introduced in this chapter. A static HTML page can look attractive with text in different fonts, shapes and sizes, and colorful images sprinkled all over. However, in the world of mobile phones, we expect our webpages to also be interactive. We want to click and see things change. When we go to a site such as Amazon, we like to see items added immediately to the shopping cart when we click the buy button. Developers use multiple web technologies to enable this interactivity. The collection of web technologies including HTML, CSS, and JavaScript to create interactive websites is called Dynamic HTML, or DHTML.

One way to look at the DHTML technologies is to think of HTML code as answering the question “What to display,” CSS code as answering the question “How the display should look,” and JavaScript code answering the question “How the webpage should interact with the user.”

Using DHTML, you can create webpages that respond to user actions such as mouse clicks and hover events, without requiring the page to reload or requiring server-side processing.

Another recent development, popularized by social media websites such as Facebook and Twitter, is called the Single Page Architecture (SPA). Single Page Architecture (SPA) is a development approach in which the content and functionality of a web application is contained within a single HTML page. Instead of navigating to separate pages when clicking on links, the user remains on the same page while sections of the page get dynamically updated based on the user’s interaction with the website. The initial HTML page is loaded from the server, and then subsequent interactions with the application are handled through JavaScript. The JavaScript code updates sections of the page based on user

---


192 We consider JavaScript to be beyond the scope of this introductory book. But interested students can follow the JavaScript tutorial at the Mozilla Developer Network (MDN) link in the previous footnote.
interactions without requiring a full page reload, offering a seamless experience to visitors.\textsuperscript{193}

SPAs offer several advantages to page visitors, including a smoother user experience, faster page load times, and easier development and maintenance due to the separation of client-side and server-side code.\textsuperscript{194} Popular examples of SPAs include Gmail, Facebook, and Twitter. Their convenience is also a source of criticism of single page applications—users keep scrolling SPAs, anticipating interesting news, a phenomenon popularly labeled doom-scrolling.\textsuperscript{195}

**Extensible Markup Language (XML)**

HTML is designed specifically for publishing webpages. The popularity of HTML to publish information for the World Wide Web led to interest in generalizing this idea to standardize all forms of electronic publishing, including electronic documents, and electronic records. The core idea was that if information elements could be standardized, computers could easily modify the information for different purposes at the click of a button. For example, the same information could be formatted with stylesheets for web and print publication with the click of a button, saved as structured information into a relational database for archival storage at the click of another button, and relevant information extracted and reported to the government for tax purposes with another click.

Extensible Markup Language (XML) is the generalization of tags for electronic publishing.\textsuperscript{196} Several electronic publishing technologies use XML behind the scenes for their operation. Almost all electronic documents are saved behind the scenes as XML. For example, Figure 223 shows the XML created by Microsoft Word for the title and opening sentence of this section.\textsuperscript{197, 198} You can see the styles and content for the title and paragraph in the figure.

As you can see in Figure 223, XML encodes documents in a format that is both human-readable as well as machine-readable. XML can specify how every element of an electronic document should be interpreted. With this detail, computers can precisely and easily transform information for different uses such as recordkeeping and publishing.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{xml.png}
\caption{XML code for a Microsoft Word document.}
\end{figure}

\textsuperscript{193} The technology to load sections of the page is called AJAX (Asynchronous JavaScript and XML). AJAX requests pieces of data from the server and updates small sections on the page with the data without disturbing the user's flow. An example of the use of AJAX is how the Gmail just moves the entire view up one row when you delete an email, without reloading the entire page. You can learn more at the W3Schools AJAX tutorial at https://www.w3schools.com/xml/ajax_intro.asp (accessed June 2023).

\textsuperscript{194} However, SPA also present some challenges, such as search engine optimization (SEO) difficulties and the need for careful management of memory and performance.

\textsuperscript{195} Doomscrolling can not only be a time sink, but also a health hazard, please read about it, a starting point is https://www.webmd.com/balance/what-is-doomscrolling (accessed June 2023).

\textsuperscript{196} The homepage of the technology working group is at https://www.w3.org/XML/ (accessed June 2023).

\textsuperscript{197} To view the XML for a Microsoft Word document, first change the extension of the file from “.docx” to “.zip”. Then unzip the file and view the “document.xml” file within the Word folder.

\textsuperscript{198} The XML specification for Microsoft Office documents is called Office Open XML. You can read about it at https://en.wikipedia.org/wiki/Office_Open_XML (accessed June 2023).
XML is very useful if the organization you work for does data exchange, for example with buyers, sellers, credit card companies, banks, and other organizations. Data exchange between companies and customers is a critical aspect of online commerce, which makes XML a key language you should know about.

A unique feature of XML is that it allows users to create their own custom tags to represent specific data structures and identify elements within a document. This flexibility makes it suitable for a wide range of applications, including web services, data storage and retrieval, and configuration files for computer applications.

XML standards, which specify tags for specific uses, have been developed for most of the important information exchange contexts. For example, the XML Business Reporting Language (XBRL) helps organizations report financial performance in a universal format. Since 2018, the U.S. Securities and Exchange Commission, the principal regulator of publicly traded companies in the United States, has required that companies submit their annual reports using XBRL. An example is shown in Figure 224. A critical feature you can see in the example is that the numbers are clearly associated with their corresponding meanings specified by the International Financial Reporting Standards (IFRS). For example, the underlying XML conveys to every financial publishing technology anywhere in the world that the highlighted number of $3,404,031 represents salary expenses for the organization. This makes it easy and precise for investors to compare the performance of different organizations,

199 You can read about XBRL at https://www.xbrl.org/ (accessed June 2023).
201 The example is from https://www.xbrl.org/xbrl-samples/gleif-annual-report-2017/gleif-annual-report-2017-viewer.html#f-f6 (accessed June 2023) (you can find this example by searching for “inline xbrl example”)
and for regulators to ensure that the numbers are correctly reported. Finally, XML viewers can format documents for easy human readability, as you see in Figure 224.

FIGURE 224 — XBRL aids business users in organizing financial data.

One of the key features of XML for recordkeeping, electronic commerce, healthcare, and other industrial applications is that computers can validate data against a defined schema or document type definition (DTD). Computers can review every element of a document and ensure that the data is properly formatted and satisfies all necessary properties. For example, a DTD can specify that profits or losses must be numbers, or, as in the example of Figure 225, a person must have a name. This helps avoid errors and inconsistencies in electronic documents.

Figure 225 introduces the structure of XML documents. In this example, the XML document starts with a declaration that specifies the version of XML being used (“1.0”) and the character encoding used to encode the document (“UTF-8”).

The main content of the document is contained within the <person> element, which has several child elements representing different aspects of the person’s information. The <name> element contains the person’s name, the <age> element contains the age, and the <address> element contains the street, city, state, and ZIP code. The DTD can specify the properties of each of these tags, including whether they are required or optional.

Note that each element has a corresponding closing tag (e.g., </name>) marking the end of the element’s content, just like HTML. Also note that elements can be nested inside other elements to create a hierarchical structure, as with the <address> element in this example.

203 You can read about Unicode Transformation Format (UTF) and UTF-8 at https://blog.hubspot.com/website/what-is-utf-8 (accessed June 2023).
There are several reasons why the essential concept of XML is widely used:

- XML can be easily read by humans and parsed and processed by computers.
- XML is not tied to any specific platform or programming language, which makes it very flexible for data exchanges on the Internet.
- XML is designed to be extensible. Users can define their own tags to describe data in a way that is specific to their needs. This allows for the creation of custom data structures that can be tailored to particular use cases.
- XML documents can be validated against a defined schema or document type definition (DTD). You can easily ensure that data is properly formatted and structured to avoid errors and inconsistencies.
- XML is used for data exchanges between systems and applications. This is due in part to its flexibility, extensibility, and its ability to represent a wide range of data types and structures.
- XML is used for representing data, not formatting or styling. This allows for a clear separation between content and presentation, making it easier to reuse data in different contexts or formats.

If you receive data in an XML format, you may need an XML reader to read and parse the XML documents. An XML reader reads an XML document and extracts the information contained within it, making it available for use in applications. XML readers can also be used to validate XML documents against a schema or document type definition (DTD) and to transform XML documents into other formats.

**XML and JSON**

XML’s core idea of generalizable tags is extremely useful. However, the fast-moving technology industry has improved upon the idea to address one key limitation of XML technology—its verbosity. Of the 28 words in Figure 225, only 9 words (~33%) convey unique information. The remaining 19 words (67%) are just protocol tags, not conveying unique information about John Smith. This is inefficient, particularly when an important use is to exchange information.

Therefore, developers have increasingly replaced XML with JavaScript Object Notation (JSON). JSON replaces tags with key-value pairs, thereby eliminating closing tags and greatly reducing the number of words needed to convey the same information. Figure 226 shows an example of a JSON file that specifies how a web application written using Python’s Flask framework should be run. You can see that JSON eliminates closing tags such as </configurations>, </name> etc.
For the foreseeable future, in all likelihood, XML will continue to be popular in highly regulated environments such as financial reporting where the focus is on precisely specifying the document types and their properties. For most other contexts, where the focus is on simplified information exchange, XML is likely to be seen as a legacy technology, increasingly replaced by JSON.
**Anchor Tag:** The `<a>` tag allows you to add links to an HTML page, also known as hyperlinks

**Bold Tag:** The `<b>` tag is used to bold text

**Cascading Style Sheets (CSS):** A web markup standard that allows web designers to define the appearance and position of a webpage using special dynamic effects

**DOCTYPE Declaration:** The first line of an HTML page that tells the browser the kind of document to expect, specifically the version of HTML

**Form Tag:** Created using the `<form>` tag; every form also has multiple `<input>` tags to specify the kind of input you want from the user

**Head Tag:** The head tag contains information about the HTML document; it is mostly not visible to the user

**Heading Tag:** The `<h1>` tag represents the main heading on a page and is typically used for the page title, while the other headings `<h2>` to `<h6>` are used for subheadings

**HTML Tag:** A tag is a special keyword that is enclosed by `<>`; the HTML tag defines the start and end of an HTML document

**Hypertext Markup Language (HTML):** The standard markup language used for creating webpages and web applications

**Image Tag:** The `<img>` tag is used to add images to your HTML page; it requires the source (src) attribute to give the location of the image so the browser can get the image and display it on the webpage

**JavaScript:** A programming language that lets you supercharge your HTML with animation, interactivity, and dynamic visual effects

**Paragraph Tag:** The `<p>` tag is used to define a block of text as a paragraph

**POST:** An HTML method that allows users to submit data to a server

**Storyboard:** A process used in the development of a website that involves the designer and colleagues simulating a particular set of visits to the site in order to gain ideas about how the site should be structured

**Strike Tag:** The `<strike>` tag is used to visually strikethrough text

**Subscript Tag:** The `<sub>` tag is used to display text as subscript

**Superscript Tag:** The `<sup>` tag is used to display text as superscript

**Tables:** Created using the `<table>` tag, and `<tr>` and `<td>` tags to define the rows and cells of the table

**Underline Tag:** The `<u>` tag is used to underline text

**Wireframe:** A rough visual outline of a proposed application and is a specific kind of prototype generated during the user-centered design process to collect input and feedback from target users before designs are finalized
**Webserver:** Computer servers specifically designed and programmed to host websites and facilitate the communication between other computers across Internet networks

**Website Host:** Typically, a company or provider that allows users to maintain, secure, and support website domains on web servers
Wrenley Saves the Pets!

Wrenley was recently given an opportunity to develop a website for a local non-profit company dedicated to helping animals housed in an animal shelter. The focus of the website was to help the pets find permanent homes through adoption. Wrenley really had a passion for pets and wanted to help find them new homes anyway she could.

It is your job to help Wrenley plan out the new website.

Question 1: Write out a detailed list of all the webpages Wrenley should include in her website (examples: “About Us”, “Contact Us”, “Donate”). As you list each webpage, include a small sample of content you think would be helpful on the webpage.

Question 2: What webpage development languages would you recommend on Wrenley’s website and explain why those languages are necessary?

Question 3: Research a good website host provider that could potentially host Wrenley’s website. List the URL and provide a description of the website host provider.

Bonus: Develop an HTML prototype of your proposed website for Wrenley. You can learn more about how to develop a website using HTML at W3Schools (https://www.w3schools.com/html/default.asp).