
Lecture 1

Contents: Introduction; Editors for writing HTML program; Installing `notepad++` editor; Writing the first program; Executing a HTML program; Basic concepts / elements; Origin and current version of HTML; Exercises

1.1 Introduction

HTML stands for *hypertext markup language*, used widely as a language to write web pages. Hypertext documents are interconnected by hyperlinks, which are typically activated by a mouse click, or keypress set, or screen touch. The term *hypertext* is also sometimes used to describe tables, images, and other presentational content formats with integrated hyperlinks. Hypertext is one of the key underlying concepts of the World Wide Web, where Web pages are often written in HTML.

1.2 Editors for writing HTML program

There are many editors by which an HTML program could be developed. A few names are given as follows:

- Atom
- Notepad ++
- Sublime Text
- Visual Studio Code
- Adobe Dreamweaver CC
- Froala
- CoffeeCup

We shall take `notepad++` as an editor for coding HTML programs. `notepad` also serves as an editor.

`notepad++` is a free editor and it supports several languages. Running in the MS Windows environment, its use is governed by GNU General Public License.

Based on the powerful editing component `Scintilla`, `notepad++` is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size.

NOTE: GNU stands for GNU's Not Unix; API has full form application programming interface; STL is a file format native to the stereolithography CAD software created by 3D Systems. STL has several acronyms such as *standard triangle language* and *standard tessellation language*.

1.3 Installing `notepad++` editor

It is a free software. Visit website <https://notepad-plus-plus.org/downloads/> to download `notepad++` software. Otherwise, you can give a search using “download

notepad++”using Googlesearch. Follow the instructions to install the software and choose option to create a shortcut at the desktop.

1.4 Writing the first program

Every HTML program has extension `.html`. One can execute an HTML program in different ways.

▷Firstly, one can just double click the icon of the file containing the HTML program.

▷One could write or copy the entire path of the file containing HTML program in the address bar of a browser and then press the “Enter” key of the keyboard. Then the program gets executed.

HTML programs are tag-based. Here is an example of program file `first.html`.

```
<html>
<head>
<title>First Page</title>
</head>
<body>
<h1>This is an H1 heading</h1>
<p>My day starts with a morning walk of 45 minutes. </p>
</body>
</html>
```

Figure 1.1: An HTML program that displays an H1 heading and a paragraph

The output is displayed on the screen. After the execution of the program given in Fig. 1.1, the title “First Page” is displayed in the tab, placed above the screen. Note that program title can be different from the file name. The file name is given here as “first.html”. The output of the program is displayed on the screen, and highlighted below:

This is an H1 heading

My day starts with a morning walk of 45 minutes.

It has two lines: first line is an H1 heading, the string put within `<h1>` and `</h1>` tags, and second line is a paragraph, put within `<p>` and `</p>` tags. There are six different heading tags: H1, H2, H3, H4, H5, H6. Students are encouraged to replace H1 by one of the remaining headings and check the output.

1.5 Text formatting tags

Let us work with different text processing tags such as bold, italic, underlined, strong, super script and sub script. The program in Fig. 1.2 illustrates these tags. The output of the program is given below.

This is a line of bold text.

But, this line is kept in italic.

This is an underlined text.

But, this line looks strong.

Write 100 to the power 2: 100²

and x subscript 3: x₃

Note that we have used tag `
` to display the next output at the next line, i.e., a line-break given in the text.

1.6 Basic concepts / elements

▷ An HTML program may start with the statement `<!DOCTYPE html>`. This declaration represents the document type as `html` program, and helps browsers to display web pages correctly.

▷ How do you view HTML source code? Right-click on an HTML page and select "View page source" in Chrome or in Edge, or similar in other browsers.

▷ An HTML element is everything from the start tag to the end tag. Refer to Fig. 1.1. Here, `<title>First Page</title>` is an element. The following is an example of nested element.

```
<head>
```

```
<title>First Page</title>
```

```
</head>
```

▷ An element containing another element is known as nested element. HTML elements with no content are called empty elements. In Fig 1.2, `
` is an empty element.

```
<html>
<head>
<title>Text formatting tags</title>
</head>
<body>
<b>This is a line of bold text.</b><br>
<i>But, this line is kept in italic.</i><br>
<u>This is an underlined text.</u><br>
<strong>But, this line looks strong.</strong><br>
Write 100 to the power 2: 100<sup>2</sup><br>
and x subscript 3: x<sub>3</sub>
</body>
</html>
```

Figure 1.2: Different text processing tags

▷ HTML is not case sensitive. This means that `<body>`, `<Body>` and `<BODY>` are the same tag.

▷ Some names of browsers that support HTML programs are Chrome, Edge, Firefox, Safari and Opera.

▷ To work with `notepad`, you need Windows 8 or later. Then open the Start screen (the window symbol at the bottom left on your screen). Type `notepad`.

Lecture 2

Contents: `style` attribute; Putting comments; `<i>` and `` elements; `<blockquote>`, `<address>`, `<q>` and `<cite>` elements; Exercises

2.1 Working with `style` attribute

Headings can be changed with `style` attribute. See an example in Fig. 2.1.

```
<!DOCTYPE html>
<html>
<body>
<h2>Heading 2</h2>
<p>Ordinary H2 heading.</p>
<h2 style="font-size:30px;">Heading 2</h2>
<p>H2 heading with style property.</p>
</body>
</html>
```

Figure 2.1: `<h2>` heading with `style` attribute

Students are required to execute the program to understand the difference between ordinary H2 heading and H2 heading with `style` attribute. Output is highlighted below:

Heading 2

Ordinary H2 heading.

Heading 2

H2 heading with style property.

The HTML `style` attribute is used to add styles to an element, such as color, font, and size. The program in Fig. 2.2 generates the following output with background colour “powderblue”.

This is an H1 heading.

This is a sunny day.

```
<html>
<body style="background-color:powderblue;">
<h1>This is an H1 heading. </h1>
<p>This is a sunny day.</p>
</body>
</html>
```

Figure 2.2: <body> with style attribute

2.2 Putting comments

Comments play important role in coding programs. It is a good programming practice to put comments, whenever coder wishes to convey a message. Comments are not interpreted by the system. Comments are simply ignored, and do not generate any output, when the program is executed. Here is an example. (See Fig. 2.3)

The following output is produced.

This is a paragraph.

```
<html>
<body>
<!--This is a comment. It does not produce any output-->
<p>This is a paragraph.</p>
</body>
</html>
```

Figure 2.3: An HTML program having a comment statement

2.3 Understanding the difference between <i> and elements

```
<html>
<body>
<p>All the days are <i>not special</i>. </p>
<p>But, some days are <em>special</em>. </p>
</body>
</html>
```

Figure 2.4: An HTML program to understand the difference between <i> and elements

`<i>` element displays a text in italic. It is often used to indicate a technical term, a phrase from another language, a special thing. But, `` element is used to emphasize a text.

Check the output of program given in Fig. 2.4. The output of the program is highlighted below.

All the days are *not special*.

But, some days are *special*.

Check the word “special” displayed in the two lines of output. We observe that there is no much difference between their looks.

2.4 Working with `<blockquote>`, `<address>`, `<q>` and `<cite>` elements

While applying `<blockquote>` tag, a section is normally quoted from another source. A `<blockquote>` tag can be placed with `cite` attribute having a web address. The ideas of `cite` attribute and `<cite>` tag are different. Example in Fig. 2.5 illustrates the usage of `<blockquote>` and `<cite>` tags.

```
<html>
<head>
<title>Blockquote tag</title>
</head>
<body>
<h2>Working with blockquote tag</h2>
<p>Goa, India</p>
<blockquote cite = "https://en.wikipedia.org/wiki/Goa">
Goa is a state on the southwestern coast of India within the Konkan region, geographically
separated from the Deccan highlands by the Western Ghats. It is surrounded by the Indian
states of Maharashtra to the north and Karnataka to the east and south, with the Arabian
Sea forming its western coast. It is India's smallest state by area and its fourth-smallest by
population. Goa has the highest GDP per capita among all Indian states, two and a half times
as high as the GDP per capita of the country as a whole.
</blockquote>
<cite>Wikipedia</cite>
</body>
</html>
```

Figure 2.5: HTML program illustrating the usage of `<blockquote>` and `<cite>` tags

Now, we make use of `<q>` and `<address>` tags in a program shown in Fig. 2.6. Note the difference between a text formatted using `<q>` tag and displaying text using normal “ ”. We do not observe much difference between their appearances.

The output of the program given in Fig. 2.6 is shaded below.

The goal of sustainable living is to "live in harmony with nature".

The goal of sustainable living is to "live in harmony with nature".

Lecture 3

Contents:Hyperlink, image, email link, button; Bookmark; Unordered list, ordered list; Exercises

3.1 Embedding hyperlink, image, email link and button within a page

Hyperlinks allow pages to remain connected. Once a link is clicked, it moves to another web page in the same website, or to a new website. Also, a document can be downloaded by clicking a link.

```
<html>
<head>
<title>Linking documents</title>
</head>
<body>
<p><a href="https://www.adhikariacademy.com/" target="_blank">Adhikari Academy</a></p>
<!-- It opens website of Adhikari Academy in a new tab -->
<a href="https://www.adhikariacademy.com/misc/List_of_holidays.pdf">

</a>
<!-- It opens a pdf document using the image as a link-->
<p><a href="mailto:animeshadhikari@yahoo.com">Email to author</a></p>
<button
onclick="document.location='https://www.adhikariacademy.com/current_news.php'">Current
Updates</button>
<!-- Once the button is clicked, it opens current updates of Adhikari Academy at the same tab-->
</body>
</html>
```

Figure 3.1: HTML program illustrating different links

See Fig. 3.1. The output of the program is shown in Fig. 3.2, and it is obtained by pressing print screen button of the keyboard and then pasting the image into this word document. (Reference Windows 11)

Due to the presence of attribute / value `target = "_blank"` in `<a>` tag, it opens the target document in a new window or tab. In the `` tag, the `alt` attribute specifies an alternate text for an image, if the image cannot be displayed.

3.2 Creating bookmark

Refer to program in Fig. 3.3. Learners are required to execute the program. It illustrates how one can jump to a specific part of a given document.

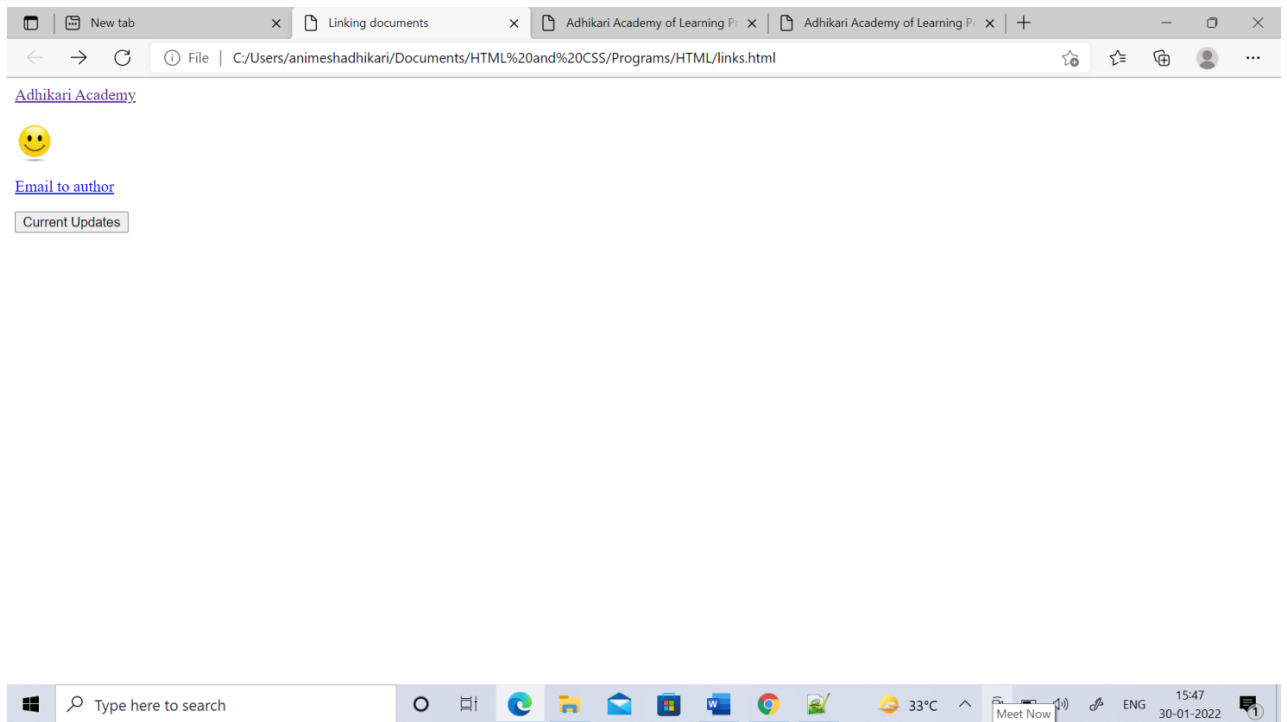


Figure 3.2:Output of program in Fig. 3.1

```
<html>
<body>
<p><a href = "#C7">Jump to Chapter 7</a></p>
<h2 id = "C1">Chapter 1</h2>
<p>This is Chapter 1. </p>
<h2>Chapter 2</h2>
<p> This is Chapter 2. </p>
<h2>Chapter 3</h2>
<p> This is Chapter 3. </p>
<h2>Chapter 4</h2>
<p> This is Chapter 4. </p>
<h2>Chapter 5</h2>
<p> This is Chapter 5. </p>
<h2>Chapter 6</h2>
<p> This is Chapter 6. </p>
<h2 id = "C7">Chapter 7</h2>
<p> This is Chapter 7. </p>
<h2>Chapter 8</h2>
<p> This is Chapter 8. </p>
<h2>Chapter 9</h2>
<p> This is Chapter 9. </p>
<h2>Chapter 10</h2>
<p> This is Chapter 10. </p>
<h2>Chapter 11</h2>
<p> This is Chapter 11. </p>
<h2>Chapter 12</h2>
<p> This is Chapter 12. </p>
<h2>Chapter 13</h2>
<p> This is Chapter 13. </p>
<h2>Chapter 14</h2>
<p> This is Chapter 14. </p>
<h2>Chapter 15</h2>
<p> This is Chapter 15. </p>
<p><a href = "#C1">Jump to Chapter 1</a></p>
</body>
</html>
```

Figure 3.3: HTML program illustrating bookmark

3.3 Working with lists

We create unordered lists using the program given in Fig. 3.4.