مجتمع فنّی تهران

HTML5

Tehran Institute of Technology

Zahra Mansoori
z.mansoori@gmail.com
http://tarsimm.com

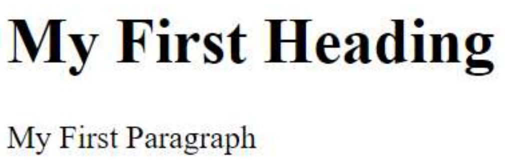
Contents

1. Introduction

HTML stands for Hyper Text Markup Language. It is the standard markup language for creating Web pages, which consists of a series of elements to tell the browser how to display the content. Tags represent the elements.

Example.1: First Page

```
<!DOCTYPE html>
<html>
  <head>
    <title> Page Title </title>
  </head>
  <body>

     <h1> My First Heading </h1>
     <p> My First Paragraph </p>

  </body>
</html>
```



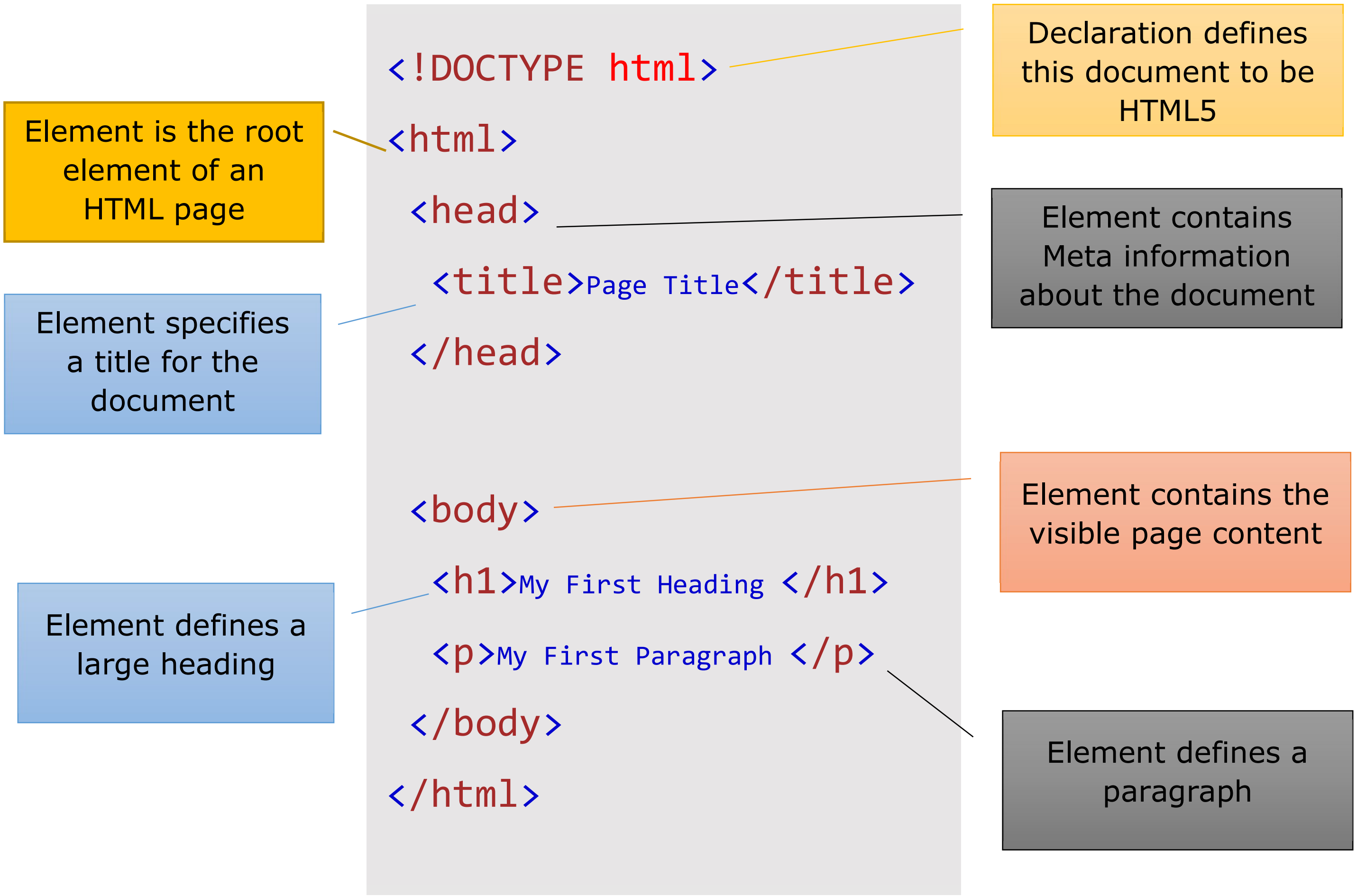
My First Heading

My First Paragraph

- The `<!DOCTYPE html>` declaration defines this document to be HTML5
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the document
- The `<title>` element specifies a title for the document
- The `<body>` element contains the visible page content
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

1.1. First Page Description

Previous example is described below in detail:



```
<!DOCTYPE html>
<html>
  <head>
   <title>Page Title</title>
  </head>

  <body>
   <h1>My First Heading </h1>
   <p>My First Paragraph </p>
  </body>
</html>
```

Declaration defines this document to be HTML5

Element is the root element of an HTML page

Element contains Meta information about the document

Element specifies a title for the document

Element contains the visible page content

Element defines a large heading

Element defines a paragraph

1.2. HTML Tag

Here you can introduce overall specific of your HTML page, one important attribute is dir and lang when you want to make a page right-to-left with specific language.

Example.2: Meta tag for keywords

```
<HTML dir=”rtl” lang=”fa”>
```

1.3. What are placed in <Head>?

Usually when you view a Web page, a friendly, descriptive title appears in that spot instead. A <title> tag that is placed in the <head> section specifies that text.

Another element you can place in the header is the <meta> tag. The <meta> tag has several purposes. One of these is to identify keywords related to your page. Placing appropriate keywords on your page can make it easier for people to find your page when they are searching the Web using a search engine such as MSN. When some search engines index your page, they rely not only on the full text of the page, but also on any keywords, they find in the `<meta>` tag area.

Example.3: Meta tag for keywords

```
      <meta name=”keyword” content=”pests, weeds, fungus, plants, flowers,
vegetables”>
```

If you want, you can add to the *charset* section to the *Head* of your document to explicitly spell out that your page is in English.

Example.4: Meta tag for character Set

```
      <meta charset=”utf-8”>
```

2. Formatting Texts

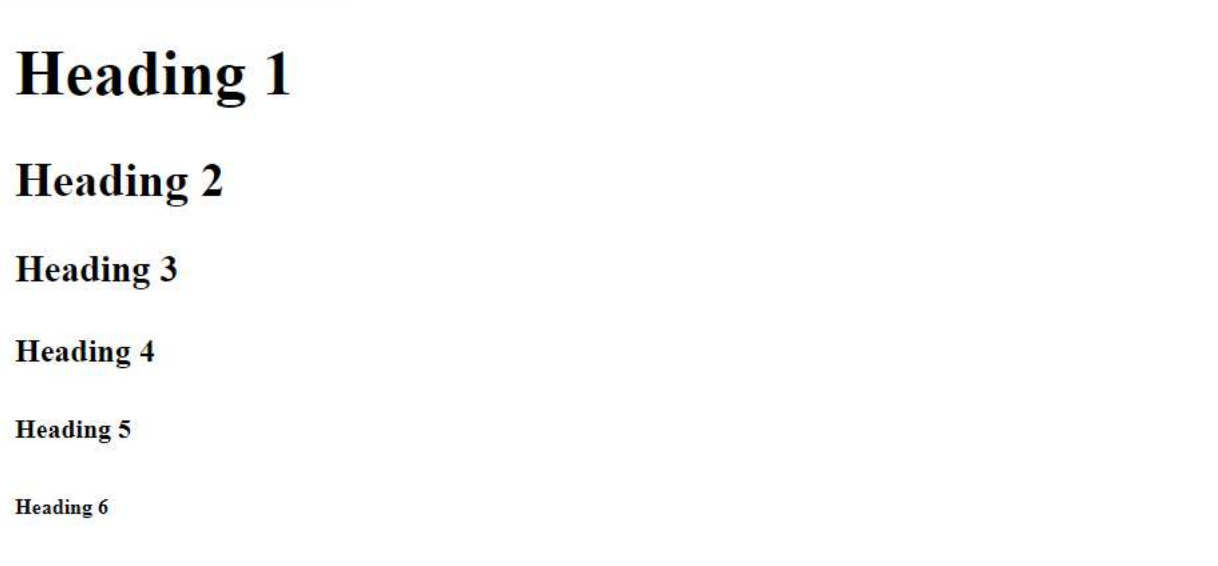
There are some semantic tags, which describes the function of the text.

2.1. Creating Headings

Headings in Web pages function separate text into sections. The HTML standard defines six levels of headings, <h1> through <h6>, each one progressively smaller.

Example.5: Headings

```
<body>

   <h1> Heading 1 </h1>
   <h2> Heading 2 </h2>
   <h3> Heading 3 </h3>
   <h4> Heading 4 </h4>
   <h5> Heading 5 </h5>
   <h6> Heading 6 </h6>

</body>
```



Heading 1

Heading 2

Heading 3

Heading 4

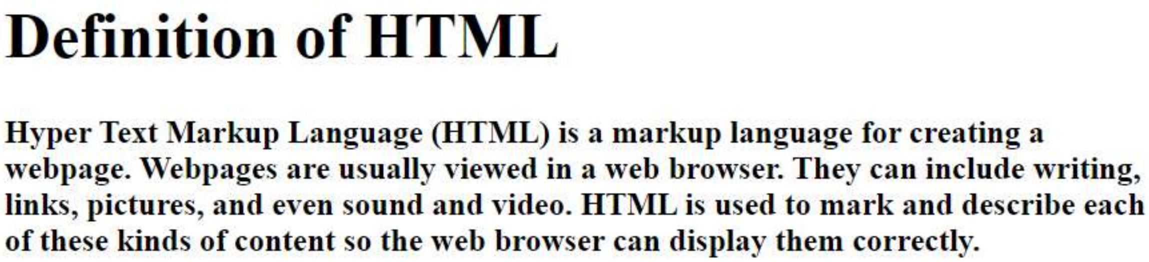
Heading 5

Heading 6

HTML5 introduces a new tag to deal with this situation, called <hgroup>. The HTML <hgroup> element represents a multi-level heading for a section of a document. It groups a set of <h1> to <h6> elements. When you enclose a stack of headings within <hgroup>, only the first heading in the stack will appear in an outline; screen readers and other outlining tools will ignore the others.

Example.6: Using Headings

```
<body>
      <hgroup>

           <h1>      Definition      of      HTML      </h1>
           <h4> Hyper Text Markup Language (HTML) is a markup
      language for creating a webpage. Webpages are usually viewed
      in a web browser. They can include writing, links, pictures,
      and even sound and video. HTML is used to mark and describe
      each of these kinds of content so the web browser can display
      them correctly.</h4>

      </hgroup>
</body>
```



Definition of HTML

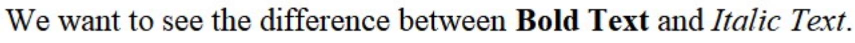
Hyper Text Markup Language (HTML) is a markup language for creating a webpage. Webpages are usually viewed in a web browser. They can include writing, links, pictures, and even sound and video. HTML is used to mark and describe each of these kinds of content so the web browser can display them correctly.

2.2. Bold and Italic Formatting

Applying bold and italic styles are two ways of making text stand out and attract attention. For simple **boldface** and *italics*, use the <b> and <i> tags, respectively. These are two-sided tags that enclose the text to be formatted.

Example.7: Boldface or italic Formatting

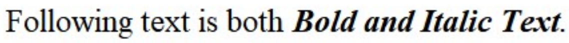
```
        <body>

           <p> We want to see the difference between <b> Bold Text </b>
      and <i> Italic Text </i>.</p>

        </body>
```



We want to see the difference between **Bold Text** and *Italic Text*.

Example.8: Boldface and italic Formatting

```
        <body>

           <p>Following  text  is  both  <b><i>  Bold  and  Italic  Text
      </i></b>.</p>

        </body>
```



Following text is both ***Bold and Italic Text***.

Hint: HTML also allows the <strong> tag as a substitute for <b>, and the <em> tag (emphasis) as a substitute for <i>.

2.3. Superscript or subscript Formatting

You can set a text character set slightly above (*superscript*) or below (*subscript*) the normal line of type using <sup> and <sub> tags.

Example.9: Superscript and subscript formatting

```
        <body>

            <p> Example of Superscrip: X<sup>2</sup> + 1.</p>
            <p> Example of Subscrip: H<sub>2</sub>O.</p>

        </body>
```

Example of Superscrip: X^2 + 1.

Example of Subscrip: H_2O.

2.4. Keyboard, Code and Sample Formatting

<kbd> represents a span of inline text denoting textual user input from a keyboard, voice input, or any other text entry device.

Example.10: Keyboard formatting

```
        <body>

            <p> Please press <kbd>Ctrl</kbd> + <kbd>Shift</kbd> +
      <kbd>R</kbd> for Render.</p>

        </body>
```

Please press Ctrl + Shift + R for Render.

The HTML <code> tag is used for indicating a piece of code.

Example.11: Code formatting

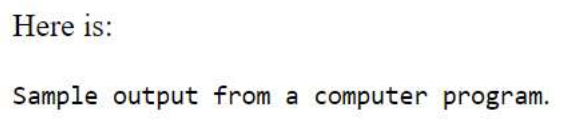
```
        <body>

            <p> Learning <code>HTML5</code> as Markup Language.</p>

        </body>
```

Learning HTML5 as Markup Language.

The HTML <samp> tag is used for indicating a piece of code.

Example.12: Sample formatting

```
<body>

    <p> Here is: </p>
    <samp> Sample output from a computer program</samp>.

</body>
```



Here is:

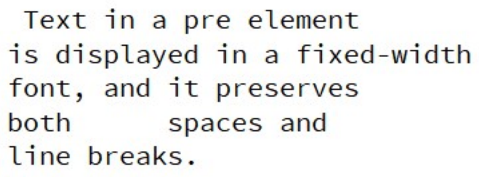
```
Sample output from a computer program.
```

2.5. Preformatted texts

The <pre> tag can also come in handy when text that you copied and pasted from another source contains many line and paragraph breaks. You could manually enter a <br> for every line break and a <p> for every paragraph break, but that is pretty labor intensive for a large file with many breaks. Using the <pre> tag is a shortcut. One common use for the <pre> tag is in poetry archives, for example, where line breaks and spacing add meaning to the poems.

Example.13: Pre formatting

```
<body>

    <pre> Text in a pre element
    is displayed in a fixed-width
    font, and it preserves
    both      spaces and
    line breaks.
    </pre>

</body>
```



```
 Text in a pre element
is displayed in a fixed-width
font, and it preserves
both      spaces and
line breaks.
```

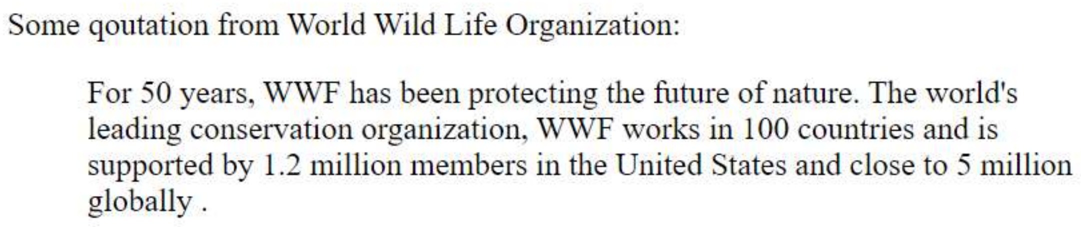
2.6. Block Quotation

When quoting blocks from other sources, it is customary—both on Web pages and in print—to indent those blocks from the main body of the text. The <blockquote> tag does exactly that. In addition, do not feel constrained about using it; you can use <blockquote> for any text that you want to indent, not just quotations.

The <blockquote> tag has a cite="URL" attribute, but most browsers do not do anything with it. If you happen to know the URL for the source you are citing, it is good practice to include it in the tag for browsers that do support the attribute, and as an aid to anyone who might be viewing or editing your raw HTML code later.

Example.14: Block quote formatting

```
              <body>

            <p>Some qoutation from World Wild Life Organization:</p>

            <blockquote cite="http://www.worldwildlife.org/who/index.html">
            For 50 years, WWF has been protecting the future of nature.
            The world's leading conservation organization, WWF works in
            100 countries and is supported by 1.2 million members in the
            United States and close to 5 million globally
            .</blockquote>

        </body>
```



Some qoutation from World Wild Life Organization:

> For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally .

2.7. Other text formatting tags

There are some text formatting tags available in HTML, they are listed below:

<strong>	**Important text**
<em>	*Emphasized text*
<mark>	Marked text
<small>	Small text
<del>	~~Deleted text~~

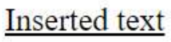
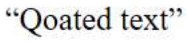
`<ins>`	 Inserted text
`<q>`	 “Qoated text”

Table.1 Text formatting tags

2.8. HTML Commenting

An HTML comment begins with `<! - -` and the comment closes with `- - >` . HTML comments are visible to anyone that views the page source code, but are not rendered when the HTML document is rendered by a browser.

3. Lists and Backgrounds

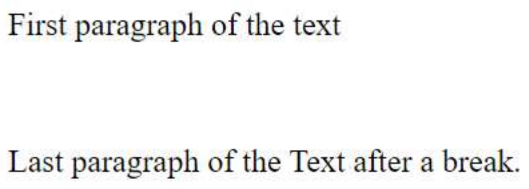
In this chapter, you will learn how to use bullets, numbering, and some special attributes for creating background colors.

3.1. Paragraph and Break tags

The HTML <p> element defines a paragraph and <br> tag defines a break between paragraphs or other tags.

Example.15: Paragraph and Break tags

```
      <body>

            <p>First paragraph of the text</p>
            <br>
            <p>Last paragraph of the Text after a break.</p>

       </body>
```



First paragraph of the text

Last paragraph of the Text after a break.

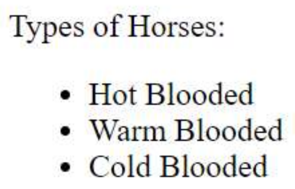
3.2. Bulleted and Numbered Lists

The tag for a numbered list is <ol>, which stands for ordered list. For a bulleted list, the tag is <ul>, which stands for unordered list. Each numbered or bulleted item within the list is tagged <li>, for list item.

You start the list with the opening <ol> or <ul> tag, enclose each list item with <li> and </li> tags, and then end the list with the closing </ol> or </ul> tag.

Example.16: Bulleted Lists

```
<body>
      <p>Types of Horses:</p>
      <ul>
            <li>Hot Blooded</li>
            <li>Warm Blooded</li>
            <li>Cold Blooded</li>
      </ul>
</body>
```

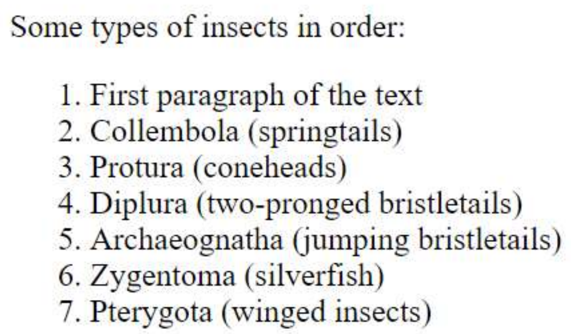


Types of Horses:

- Hot Blooded
- Warm Blooded
- Cold Blooded

Example.17: Numbered Lists

```
<body>
      <p>Some types of insects in order:</p>
      <ol>
            <li>First paragraph of the text</li>
            <li>Collembola (springtails)</li>
            <li>Protura (coneheads)</li>
            <li>Diplura (two-pronged bristletails) </li>
            <li>Archaeognatha (jumping bristletails) </li>
            <li>Zygentoma (silverfish) </li>
            <li>Pterygota (winged insects) </li>
      </ol>
 </body>
```



Some types of insects in order:

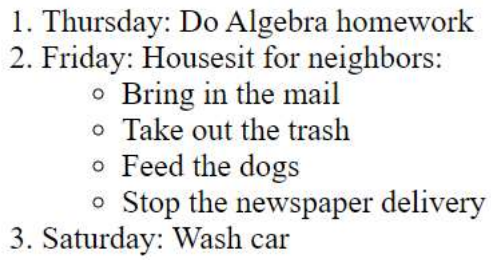
1. First paragraph of the text
2. Collembola (springtails)
3. Protura (coneheads)
4. Diplura (two-pronged bristletails)
5. Archaeognatha (jumping bristletails)
6. Zygentoma (silverfish)
7. Pterygota (winged insects)

3.3. Nesting Lists

You can nest lists within one another. Look at Example.18, bulleted sublist (the <ul> tag) is placed within one of the <li> tags within the numbered <ol> list.

Example.18: Nested Lists

```
      <body>

            <ol>
                  <li>Thursday: Do Algebra homework</li>
                  <li>Friday: Housesit for neighbors:
                        <ul><li>Bring in the mail</li>
                        <li>Take out the trash</li>
                        <li>Feed the dogs</li>
                        <li>Stop the newspaper delivery</li></ul></li>
                  <li>Saturday: Wash car</li>
            </ol>

      </body>
```



```
1. Thursday: Do Algebra homework
2. Friday: Housesit for neighbors:
     ◦ Bring in the mail
     ◦ Take out the trash
     ◦ Feed the dogs
     ◦ Stop the newspaper delivery
3. Saturday: Wash car
```

Changing the bullet or Number character

Bulleted and numbered lists can be styled by using a list-style-type: type attribute. This is a type of style-based attribute. You can use the values shown in Table.2 for the list-style-type attribute.

List Style	Value	Result
Bulleted	disc	Filled circle (the default)
	circle	Unfilled circle
	square	Filled square
Numbered	decimal	1, 2, 3, 4 (the default)
	decimal-leading-zero	01, 02, 03, 04
	lower-roman	i, ii, iii, iv
	upper-roman	I, II, III, IV
	lower-alpha	a, b, c, d
	upper-alpha	A, B, C, D
	none	(nothing)

Table.2. Bullet and Numbering Characters

To apply the attribute, place it in the opening <ul> or <ol> tag. For example, to create a bulleted list that uses the square bullet character, start the list off this way:

```
<ul style="list-style-type: square">
```

To create a numbered list that uses uppercase Roman numerals, start the list this way:

```
<ol style="list-style-type: upper-roman">
```

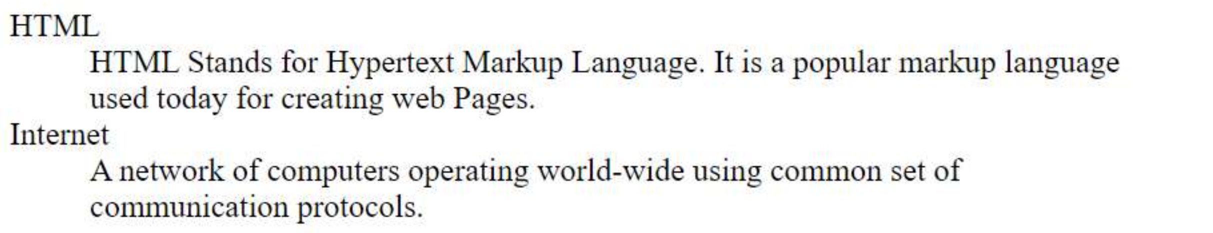
3.4. Creating Definition List

A definition list is just what it sounds like: a list that presents terms with their definitions, such as you would see in a glossary. The word being defined serves as a heading, and the definition paragraph is indented under it.

The complete list (headings and definition paragraphs) is contained within the <dl> and </dl> tags, which stands for definition list. Each word to be defined is contained in a <dt> (definition term) tag, and the definition paragraphs are in <dd> (definition description) tags. Here's the code for the example just shown:

Example.19: Definition List

```
      <body>
            <dl>
                  <dt>HTML</dt>
                  <dd>HTML Stands for Hypertext Markup Language. It is a
      popular markup language used today for creating web Pages.</dd>

                  <dt>Internet</dt>
                  <dd>A network of computers operating world-wide using
      common set of communication protocols.</dd>
            </dl>
      </body>
```



HTML
    HTML Stands for Hypertext Markup Language. It is a popular markup language used today for creating web Pages.
Internet
    A network of computers operating world-wide using common set of communication protocols.

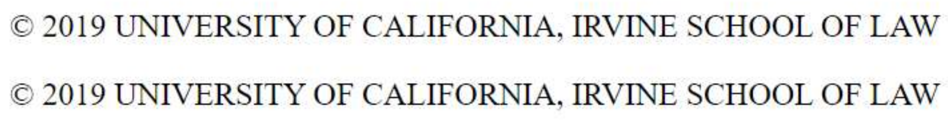
3.5. Special Characters

Some characters consider t be special because have special meaning to HTML browsers and interpreters. Therefore, if you want to use these character in you HTML code page, you have to use their equivalent special characters.

Symbol	Entity Name	Entity Number
& (ampersand)	&amp;	&#38;
< (less than)	&lt;	&#60;
> (greater than)	&gt;	&#62;
(nonbreaking space)	&nbsp;	&#160;
¢ (cent)	&cent;	&#162;
£ (pound)	&pound;	&#163;
¥ (yen)	&yen;	&#165;
© (copyright)	&copy;	&#169;
® (registered trademark)	&reg;	&#174;
° (degree)	&deg;	&#176;
± (plus or minus)	&plusmn;	&#177;
† (dagger)	&dagger;	&#8224;
™ (trademark)	&trade;	&#8482;

Example.20: Definition List

```
      <body>
            <p>&copy; 2019 UNIVERSITY OF CALIFORNIA, IRVINE SCHOOL OF LAW
      </p>
            <p>&#169; 2019 UNIVERSITY OF CALIFORNIA, IRVINE SCHOOL OF LAW
            </p>
      </body>
```



© 2019 UNIVERSITY OF CALIFORNIA, IRVINE SCHOOL OF LAW

© 2019 UNIVERSITY OF CALIFORNIA, IRVINE SCHOOL OF LAW

3.6. Horizontal Line

Horizontal lines can be useful as dividers between sections of text in a Web page. To add a horizontal line, simply add the following one-sided tag where you want the line to appear: `<hr>`

Following example shows a stylish horizontal line:

```
<hr style="color: green; background-color: green; height: 3px; width: 50%">
```

Some browsers use color to assign a **color** to the line, others use **background-color**; therefore, you should include both tags and assign the same color for both.

HTML recognizes these 16 basic color names:

Aqua	Gray	Navy	Silver
Black	Green	Olive	Teal
Blue	Lime	Purple	White
Fuchsia	Maroon	Red	Yellow

3.7. Background and Foreground Colors

Sometimes it is needed for you to choose a background or foreground color for one of your components. Here some description to do this.

Specifying Colors

The 16 basic colors presented earlier are the best colors to use on Web pages because they are universally accepted. Every browser interprets these colors the same way. However, you will probably find many situations in which none of those 16 colors is appropriate. For example, you might find that they are all too dark or too vivid to make an attractive page background. Therefore, you will sometimes need to rely on other ways of specifying colors.

One way to specify a color is by its **RGB (red-green-blue)** value. Using this method, you can describe a color using a series of three numbers, **from 0 to 255**. Each number represents the component of red, green, or blue that makes up the color. For example, **pure red is 255, 0 ,0;** that is, maximum **red (255), no green (0), and no blue (0).** You can create a large range of colors using these three values. For example, **255, 153, 0 represents a particular shade of orange**—**full red, a little more than half green, and no blue**.

Another way to express color values in HTML is by using a **hexadecimal value**. The hexadecimal values **represent the RGB values converted to the base-16 numbering system**. For example, the **value 255 converts to FF**, so the **RGB value 255, 255, 0 can also be expressed as the hexadecimal value #FFFF00.**

Applying a Background Color

To specify a background color for an entire page, insert the **style=”background-color: color”** attribute into the opening <body> tag.

```
<body style="background-color: yellow">
```

You can use the color name, the RGB value, or the hexadecimal value. Therefore, the following are equivalent to the code just shown:

```
<body style="background-color: #FFFF00">
```

```
<body style="background-color: rgb(255,255,0)">
```

Applying a Foreground Color

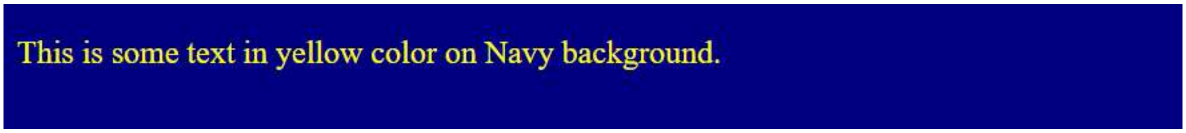
The foreground color is the default text color for the page. You can set the foreground color by using the style=”color: color” attribute. It can be combined with the attribute for the background color in a single style= statement.

```
<body style="background-color: navy; color: yellow">
```

When you combine two attributes in a single **style=** statement, you separate them with a semicolon, as shown in the preceding example.

Example.21: Specifying Foreground and Background colors

```
      <body style="background-color: navy; color: yellow">
            <p>This is some text in yellow color on Navy background.</p>
      </body>
```



This is some text in yellow color on Navy background.

3.8. Background Image File

A background image appears behind the text on a page. By default, the image is tiled to fill the page and scrolls with the page.

If you specify a background image, you should also specify a background color. The color will not be visible unless the image fails to display for some reason, or unless the image has transparent areas in it. The background color is especially important if you use a dark background image and a light foreground color; if the image does not appear, the text still must appear on a dark background to be readable.

To specify a background image, use below syntax:

```
<body style="background-image: url(granite.gif)">
```

To specify Background and Foreground colors and Background Image use below syntax:

```
<body style="background-image: url(granite.gif); color: green; background-color:
beige">
```

By default, the background image scrolls with the text when the user scrolls down the page. To force the image to stay fixed, add the **background-attachment=fixed** attribute to the <body> tag, as follows:

```
<body style="background-image: url(granite.gif); background-color: beige;
background-attachment=fixed">
```

4. Creating Hyperlinks and anchors

The Web is based on hyperlinks. Each Web page contains active links to other pages, which in turn link to even more pages, until presumably the entire Web (or at least a great chunk of it) is bound together. In fact, that is where the name “web” comes from.

4.1. Hyperlinking to a Web Page

No matter what type of hyperlink you want to create, the basic syntax is the same. It starts with the <a> tag, and then uses an href= attribute which provides the URL or the path to the destination. For example, an opening tag might look like this:

```
<a href="http://www.microsoft.com/en/us/default.aspx ">Microsoft page</a>
```

Or

```
<a href="http://www.microsoft.com/en/us ">Microsoft Entry point page</a>
```

4.2. Setting a Target Window

To direct the hyperlink to open a page in a new window, add the attribute **target=”blank”** to the <a> tag. For example, to open the foliage.htm file in a new window, the tag would be structured like this:

```
<a href="foliage.htm" target="_blank">Diagnosing Foliage Problems</a>
```

4.3. Hyperlinking to an E-Mail Address

Hyperlinks can point to anything, not just to Web pages. You can create e-mail hyperlinks, for example, that start the user’s default e-mail program, create a new message, and enter the recipient’s address. (You can also set it to fill in the message subject, if you like.)

E-mail hyperlinks are useful when you want to direct someone to send a message to a particular person, as below:

```
<a href="mailto:support@adatum.com">Contact Us</a>
```

To add a default subject line to the e-mail, add a **?subject=** attribute after the e-mail address, like this:

```
<a href="mailto:support@adatum.com?subject=Comment">Contact Us</a>
```

4.4. Creating to Anchors

An **anchor** is a marker within an HTML document, roughly analogous to a bookmark in a Word document. You define a specific location in the document with an anchor name, and then you can hyperlink directly to that anchor.

Anchors are most valuable in long documents with multiple sections. They provide a means for users to jump directly to whatever section they want rather than having to read or scroll through the entire document. You can do this internally by creating a list of hyperlinks at the top of the document, or you can do this externally by including an anchor name in a hyperlink to another document. There are two parts to the process:

- You mark the anchor location, and then
- You create a hyperlink that refers to it

To define an anchor, create an <a> tag around the destination text and include a **name=** attribute. For example, suppose you have a heading that reads Conclusion, and you want to create an anchor point with that same name:

```
<a name="conclusion">Conclusion</a>
```

To refer to the anchor point, include its name in the **href=** attribute. Precede the anchor name with a pound sign (#). If the anchor point is in the same document as the hyperlink, you can use a relative reference like this:

```
<a href="#conclusion">View the Conclusion</a>
```

Otherwise, you must include the name of the file in which the anchor is located. For example, if the anchor were in a file called **report.htm**, it would look like this:

```
<a href="report.htm#conclusion">View the Conclusion</a>
```

5. HTML5 Semantic Tags

HTML5 adds some semantic tags to define layouts in more intuitive ways than the generic <div> tag is capable.

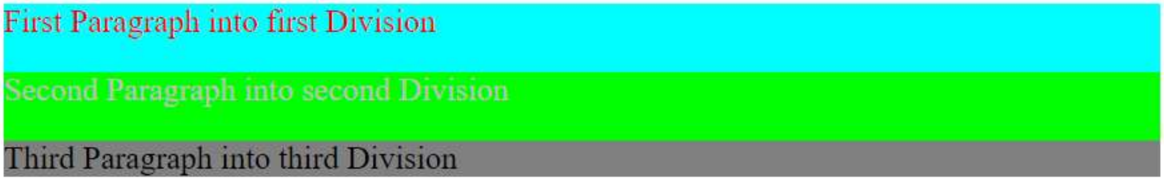
- <header>: Defines the masthead or other header information on the page. Typically, the header is repeated on every page of a site, although that is not required.
- <footer>: Defines the text at the bottom of a page, such as the copyright or contact information. Again, it is typically repeated on every page of the site.
- <article>: Defines a block of text that represents a single article, story, or message. An article can be distinguished from other text in that it can logically stand alone. For example, on a news site, each news story is an article.
- <aside>: Defines a block of text that is tangential to the main discussion, such as a note, tip, or caution. An aside can be distinguished from other text in that it could be pulled out and discarded without disrupting the main document in which it appears.
- <section>: Defines a generic content or application section. Examples of sections would be book chapters or the numbered sections of a thesis; a site's home page could be split into sections such as Introduction, News, and Contact Information. A section begins with a heading such as <h1> followed by other content. A general rule is to use <section> if the area being defined would be included in an outline of the document or page.

5.1. Thinking in Divisions

In an effective division-based layout, each part of the page you want to format separately should be a **division**. The <div> element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript.

Example.22: Using divisions

```
      <body>
            <div style="background-color: aqua; color: red">
                  <p>First Paragraph into first Division</p>
            </div>
            <div style="background-color: lime; color: silver">
                  <p> Second Paragraph into second Division </p>
            </div>
            <div style="background-color: gray; color:black">
                  <p> Second Paragraph into second Division </p>
            </div>
      </body>
```



First Paragraph into first Division

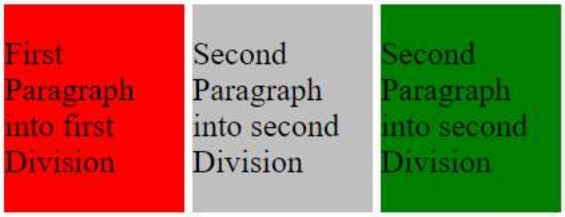
Second Paragraph into second Division

Third Paragraph into third Division

To make <div> tags horizontally align, display style can be used:

Example.23: Using divisions width display style

```
      <body>
            <div style="width: 300px; Height:300px">
                  <div    style="width:90px;    display:    inline-block;
            background-color: red">
                  <p>First Paragraph into first Division</p>
                  </div>
                  <div   style="width:90px;     display:   inline-block;
            background-color: silver">
                  <p> Second Paragraph into second Division </p>
                  </div>
                  <div    style="width:90px;    display:    inline-block;
            background-color:green">
                  <p> Second Paragraph into second Division </p>
                  </div>
            </div>
      </body>
```



First Paragraph into first Division

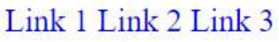
Second Paragraph into second Division

Second Paragraph into second Division

To understand more about display style, there is another example:

Example.24: List menu using display style

```
      <body>
           <div>
                <ul>
                     <li  style="list-style-type:none;display:inline-
      block"><a style="text-decoration: none" href="#">Link 1</a></li>
                     <li style=" list-style-type:none;display:inline-
      block"><a style="text-decoration: none" href="#">Link 2</a></li>
                     <li style=" list-style-type:none;display:inline-
      block"><a style="text-decoration: none" href="#">Link 3</a></li>
                </ul>

           </div>
      </body>
```



Link 1 Link 2 Link 3

5.2. HTML5 Semantic Layout

All semantic tags are a kind of division. You can put them into your webpage in a standards way and in order to define a proper layout for your webpage.

There are some rules to put these semantic layouts into your webpage that we show them in examples orderly.

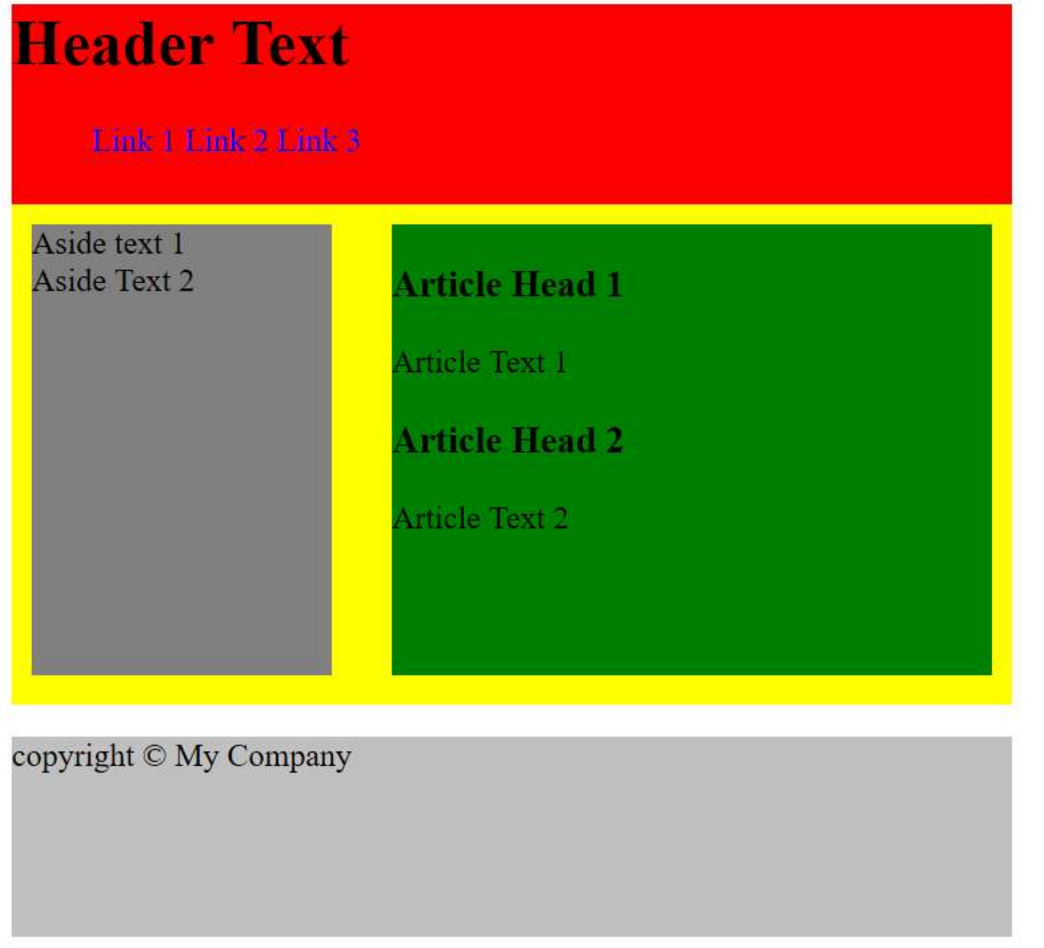
- You can put <header> at the top to include introduction of your webpage or your website, information such as **Title**, **Logo** and some **unique information of your website**.
- Put <nav> tag as navigation of your webpage or website at the bottom of even at the top of the header.
- <nav> tag usulally include **hyperlinks to other webpages** of the site or **other usefull links** to external websites.
- Put <main> after <header>, which contains main content of your webpage, the reason you created the webpage or website.
- <section> and <aside> both can be included into <main>.
- <section> can include more than one <article>.
- At the bottom of page you can put <footer> to put your **contact address**, **urgent links** and **copyright texts**.

Hint: <nav> tag also can be include at the top of your <main> tag.

Example.25: Using Semantic Layout

```
<body>

      <header style="width: 500px; height: 100px; background-color:
      Red">
            <h1> Header Text </h1>
            <nav>
                  <ul style="list-style-type:none">
                        <li style="display:inline-block"><a style="text-
                  decoration: none" href="#">Link 1</a></li>
                        <li style="display:inline-block"><a style="text-
                  decoration: none" href="#">Link 2</a></li>
                        <li style="display:inline-block"><a style="text-
                  decoration: none" href="#">Link 3</a></li>

                  </ul>
            </nav>
      </header>

      <main style="height:250px;width:500px; background-color:yellow">
            <aside style="float:left;margin:10px; height:90%;width:30%;
      background-color: Gray; display:inline-block ">
                  <div> Aside text 1 </div>
                  <div> Aside Text 2 </div>
            </aside>
            <section style="float:right; height:90%; width: 60%;
      background-color: Green; display:inline-block; margin : 10px">
                  <article>
                        <h1>Article Head 1</h1>
                        <p>Article Text 1</p>
                  </article>
                  <article >
                        <h1>Article Head 2</h1>
                        <p>Article Text 2
                        </p>
                  </article>
            </section>
      </main>

      <footer style=" width: 500px; height: 100px; background-color:
      Silver">
            <p>copyright &copy; My Company</p>
      </footer>

</body>
```

Example.25 Result: Using Semantic Layout



Header Text

Link 1 Link 2 Link 3

Aside text 1
Aside Text 2

Article Head 1

Article Text 1

Article Head 2

Article Text 2

copyright © My Company

6. Using Tables

A **table** is a grid of **rows** and **columns**, the intersections of which form **cells**. Each **cell** is a distinct area, into which you can place text, graphics, or even other tables.

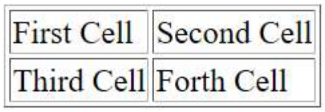
In this chapter, you will learn the basic HTML for creating tables, rows, and cells. You will also learn how to specify cell sizes and merge cells to create larger areas. After you master these skills, you will put them to work by creating a table-based page layout grid. Then, in the next chapter, you will learn how to format tables.

6.1. Creating a Simple Table

The <table> tag creates an HTML table. Within that tag, you include one or more <tr> tags, which define the table's rows, and within each <tr> tag, you define one or more <td> tags, which define the cells.

Example.26 Using table

```
<body>

      <table border="1">
            <tr>
                  <td>First Cell</td>
                  <td>Second Cell</td>
            </tr>
            <tr>
                  <td>Third Cell</td>
                  <td>Forth Cell</td>
            </tr>
      </table>

</body>
```

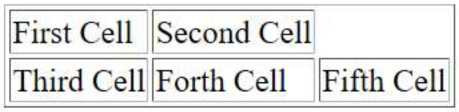


First Cell	Second Cell
Third Cell	Forth Cell

Another example about using table in an unbalanced cell counts:

Example.27 Using table with unbalanced cell count

```
    <body>

        <table border="1">
            <tr>
                <td>First Cell</td>
                <td>Second Cell</td>
            </tr>
            <tr>
                <td>Third Cell</td>
                <td>Forth Cell</td>
                <td>Fifth Cell</td>
            </tr>
        </table>

    </body>
```

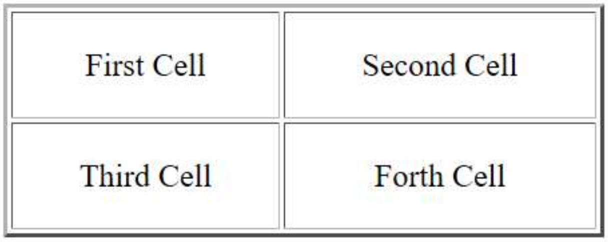


First Cell	Second Cell	
Third Cell	Forth Cell	Fifth Cell

You can specify style for the table and its cells, as example.28:

Example.28 Using table with unbalanced cell count

```
      <body>

            <table  border="2" style="width:300px">
                  <tr>
                        <td style="height:50px;  text-align:center">
                        First Cell</td>
                        <td style="height:50px;  text-align:center">
                        Second Cell</td>
                  </tr>
                  <tr>
                        <td style="height:50px;  text-align:center">
                        Third Cell</td>
                        <td style="height:50px;  text-align:center">
                        Forth Cell</td>
                  </tr>
            </table>

      </body>
```



First Cell	Second Cell
Third Cell	Forth Cell

6.2. Merging Table cells

As you have seen in the preceding sections, every cell in a given row must be the same height, and every cell in a given column must be the same width. You cannot make individual cells different heights or widths, but you can span (merge) two or more adjacent cells so that one cell spans multiple rows and/or columns. This technique is useful for centering text across several columns.

To merge a cell into adjacent cells to its right, use the **colspan** attribute and specify the number of columns to be spanned, like this:

```
<td colspan="3">
```

Merged cell in horizontal align		

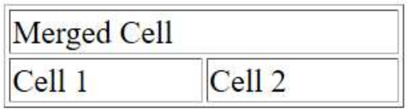
To merge a cell into adjacent cells below it, use the **rowspan** attribute and specify the number of rows to be spanned, as shown in the following:

```
<td rowspan="2">
```

Merged cell in vertical align		

Example.29 Using table with unbalanced cell count

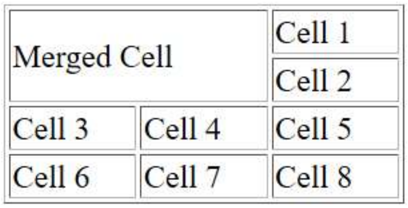
```
<body>

    <table  border="1" style="width: 200px">
        <tr>
            <td colspan="2">
            Merged Cell</td>
        </tr>
        <tr>
            <td>
            Cell 1</td>
            <td>
            Cell 2</td>
        </tr>
    </table>

</body>
```



Merged Cell	
Cell 1	Cell 2

Example.30 Using table with unbalanced cell count

```
      <body>

            <table  border="1" style="width: 200px">
                  <tr>
                        <td colspan="2" rowspan="2">
                        Merged Cell</td>
                        <td>Cell 1</td>
                  </tr>
                  <tr>
                        <td>Cell 2</td>
                  </tr>
                  <tr>
                        <td>Cell 3</td>
                        <td>Cell 4</td>
                        <td>Cell 5</td>
                  </tr>
                  <tr>
                        <td>Cell 6</td>
                        <td>Cell 7</td>
                        <td>Cell 8</td>
                  </tr>
            </table>
      </body>
```



Merged Cell		Cell 1
		Cell 2
Cell 3	Cell 4	Cell 5
Cell 6	Cell 7	Cell 8

6.3. Instruction of writing merged cells

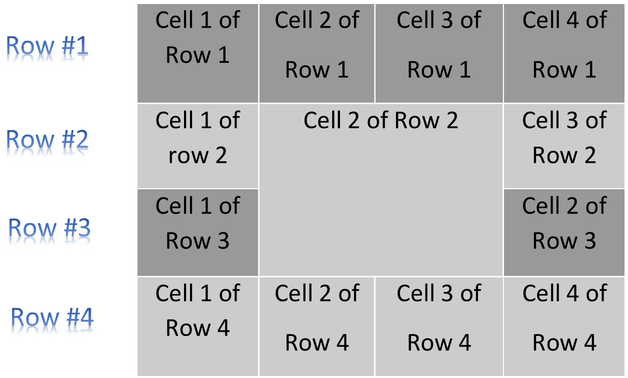
To explicitly define how to merge some cells in complicated situations, please takea look at below instruction:

1. Draw the table with merged cells on a paper
2. Define each row particularly. Every cells which locate in a horizontal row will be considered in a single particular row
3. For defining the row of merged cells, the first row which are passed through the merged cells will be assigned to merged cell
4. enumerate each row specifically from 1 to …

5. After defining the rows you have to write the code. Open a <tr>...</tr> tag for every rows
6. Previousely, you defined the rows and enumerated the cells. Now simply put them into every row without repeatation and over lap

Finally you have made it!

Exercise: Please write the HTML code of the table below.



<table>
<tr><td>Row #1</td><td>Cell 1 of Row 1</td><td>Cell 2 of Row 1</td><td>Cell 3 of Row 1</td><td>Cell 4 of Row 1</td></tr>
<tr><td>Row #2</td><td>Cell 1 of row 2</td><td colspan="2" rowspan="2">Cell 2 of Row 2</td><td>Cell 3 of Row 2</td></tr>
<tr><td>Row #3</td><td>Cell 1 of Row 3</td><td>Cell 2 of Row 3</td></tr>
<tr><td>Row #4</td><td>Cell 1 of Row 4</td><td>Cell 2 of Row 4</td><td>Cell 3 of Row 4</td><td>Cell 4 of Row 4</td></tr>
</table>

7. Creating Forms

Gathering feedback from your Web site's visitors can be a valuable way of assessing your site's success, and it can help you build a customer or subscriber database. In this chapter, you will learn how to create several types of user input forms. You will learn how to create forms with text boxes, option buttons, check boxes, and drop-down lists. You will also learn about some of the new HTML5 form controls, such as date boxes. At the end of this chapter, you will find some information and Web resources that can help you create forms that are even more advanced by using Common Gateway Interface (CGI) scripting and third party utilities and services.

7.1. Creating a Basic Form

You can place a form anywhere in the body of an HTML document. Some people like to use a table to organize form elements; others create form fields within ordinary paragraphs.

A form is enclosed in a two-sided <form> tag:

```
<form method="post">
…
</form>
```

The method attribute specifies how to send form-data (the form-data is sent to the page specified in the action attribute). The form-data can be sent as URL variables (with method="**get**") or as HTTP post transaction (with method="**post**").

Notes on GET:

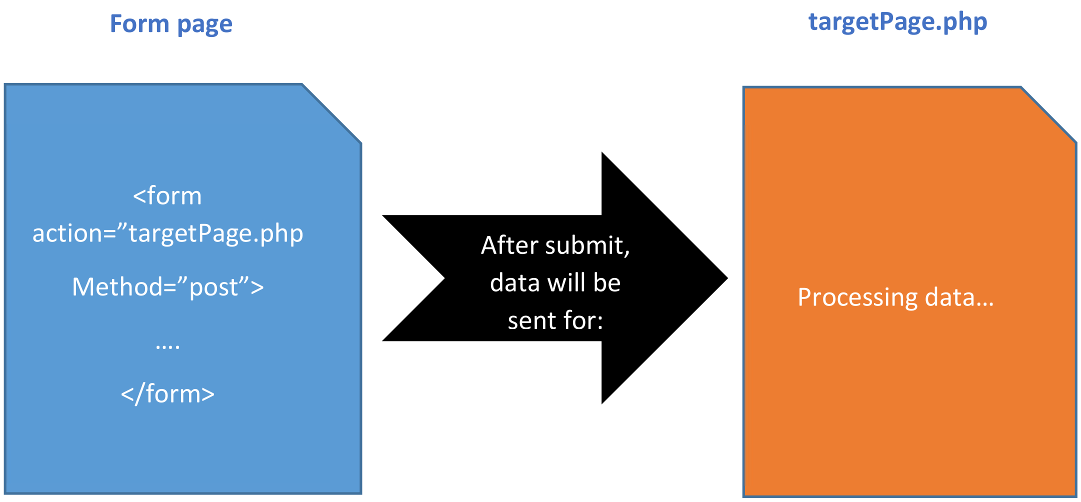
- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (about 3000 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user wants to bookmark the result
- GET is better for non-secure data, like query strings in Google

Notes on POST:

- Appends form-data inside the body of the HTTP request (data is not shown is in URL)
- Has no size limitations
- Form submissions with POST cannot be bookmarked

Every forms mus have a submit button. After submitting the form, the data in form fields will be sent to the page mentioned in **action** attribute:

```
<form action=”targetPage.php”>
…
</form>
```



7.2. Creating a Text Box

The most basic type of control is a text box. Users can enter data such as names, addresses, phone numbers, and comments into text boxes. There are two types of text boxes:

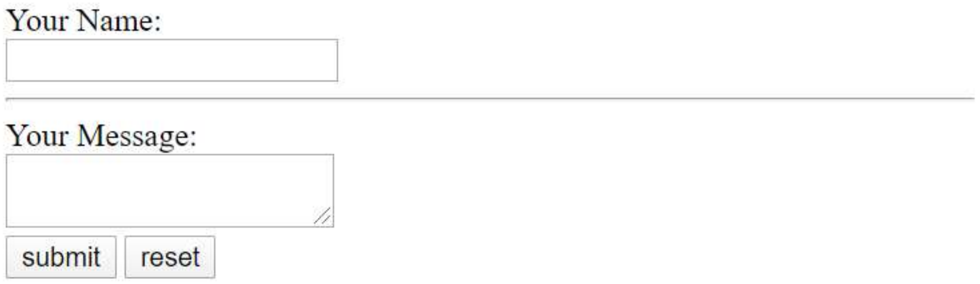
- regular text boxes (single line)
- text areas (multi-line)

Each Element of form can have a **Label** Before. At the next example, you can see a **text box** and another **Text Area** each have a specific **label** before.

We will talk about **Text Area** later.

Example.31 Text Box and Text Area Form Fields with Label

```
    <body>

        <form  method="Get" action="targetPage.php">

            <label>Your Name:</label>
            <br>
            <input type="text" name="textBox1">
            <hr>
            <label>Your Message:</label>
            <br>
            <textarea name="textArea1">
            </textarea>
            <br>
            <input type="submit" name="submit">
            <input type="reset" name="reset">
        </form>

    </body>
```



You can specify default and maximum length for both **Text Box** and **Text Area**. For default length:

```
<input type="text" name="phone" size="30">
```

For max length:

```
<input type="text" name="phone" size="30" maxlength="100">
```

In HTML5, you can require users to fill out a field before they will be able to submit the form (applies to HTML5-compliant browsers only). To mark a field as required, add the **required** attribute to its tag, like this:

```
<input type="text" name="phone" size="30" maxlength="100" required>
```

Two new input field types in HTML5 support **e-mail addresses** and Web addresses. Use the attribute type=”email” instead of type=”text” to define a field designed to collect e-mail addresses. If a browser does not support HTML5, the field defaults to a text type, so you do not risk anything by using it.

```
<input type="email" name="email-address">
```

The same goes for Web addresses (also known as uniform resource locators, or URLs). There is a special type attribute in HTML5 for them, as shown here:

```
<input type="URL" name="website">
```

By default, text boxes and text areas are blank when the form loads. You can optionally place either **default** or **placeholder text** in them.

- Default text is regular text that is submitted with the form results as if the user had actually typed it in.
- Placeholder text is “phantom” text that appears as a prompt within the text box but disappears when the user types something else there. If the user chooses to leave that text box blank, nothing is submitted.

Most browsers support the use of default text, even if they do not support HTML5. For a text box, add a value attribute to the tag that specifies the **default text**, as shown here:

```
<input type="text" name="country" value="United States of America">
```

In addition, for placeholder:

```
<input type="text" name="country" placeholder="Enter your country here">
```

7.3. Creating a Text Area

You create a multi-line text area by using a two-sided <textarea> tag containing a rows attribute that specifies the number of lines of text that the box should accommodate, such as shown in the following example:

```
<textarea name="comments" rows="5"> </textarea>
```

The columns attribute affects only the size of the box, not the maximum number of characters that can be entered. You can use the **maxlength** attribute to limit the number of characters a user can enter.

```
<textarea name="comments" rows="5" maxlength="300"></textarea>
```

The cols attribute specifies the visible width of a text area.

```
<textarea rows="4" cols="50"></textarea>
```

For **default value**, you should place default text between the opening and closing <textarea> tags, like this:

```
<textarea name="comments" rows="5">Great job! Keep up the good work.</textarea>
```

The placeholder attribute specifies a short hint that describes the expected value of a text area.

```
<textarea placeholder="Describe yourself here..."></textarea>
```

7.4. Creating a Submit or Clear Button

You will need to include a Submit button on the form so visitors can send the information to you. Submit refers to the button's function, not the wording that appears on the button face. The default button text is **Submit**, but you can use a value attribute to display different text on the button. For example, to make the word Send appear on the button face, set up the **value** attribute, as shown here:

```
<input type="submit" value="Send">
```

You can also include a Reset button on the form, which allows the user to clear all the fields. Again, use the **value** attribute to change the text on the button.

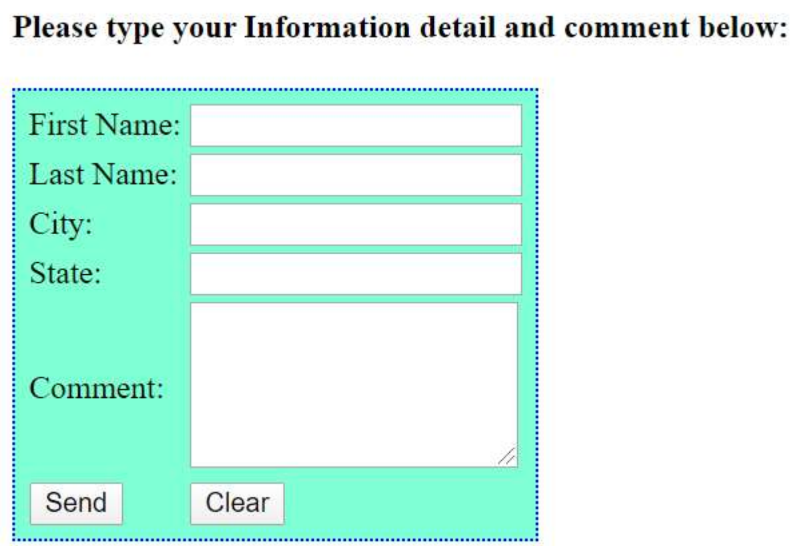
```
<input type="reset" value="Clear">
```

7.5. Use table for arranging form fields

Many Web designers find it useful to place form fields in tables to make it easier to align the fields. For example, as shown in the following image, you could place field labels in one column and the actual fields themselves in another. You will see this type of design in the example.32.

For arranging form fields in table you can put them into table cells, for example you can create a 2-column table with specific rows; then put labels at the left cells and form fields at the right cells.

Example.32 Result: Using table to decorate form fields



Please type your Information detail and comment below:

First Name:

Last Name:

City:

State:

Comment:

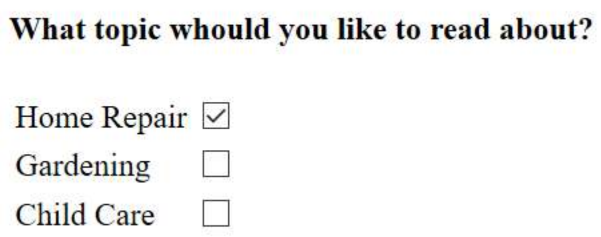
Example.32 using table to decorate form fields

```
<body>
      <form method="post" action="targetPage.php">
            <h4>Please type your Information detail and comment
      below:</h4>
            <table style="padding:5px; border: 1px dotted blue;
            background-color: aquamarine">
                    <tbody>
                        <tr>
                              <td><label>First Name:</label></td>
                              <td><input type="text" maxlength="40"
                              name="FirstName"></td>

                        </tr>
                        <tr>
                              <td><label>Last Name:</label></td>
                              <td><input type="text" maxlength="50"
                              name="LastName"></td>
                        </tr>
                        <tr>
                              <td><label>City:</label></td>
                              <td><input type="text" maxlength="30"
                              name="City"></td>
                        </tr>
                        <tr>
                              <td><label>State:</label></td>
                              <td><input type="text" maxlength="30"
                              name="State"></td>
                        </tr>
                        <tr>
                              <td><label>Comment:</label></td>
                              <td><textarea name="Comment"
                              maxlength="500" rows="5"></textarea></td>
                        </tr>
                        <tr>
                              <td><input type="Submit"
                        value="Send"></td>
                              <td><input type="Reset"
                        value="Clear"></td>
                        </tr>

                  </tbody>
            </table>
      </form>
</body>
```

7.6. Check Boxes and Option Buttons

Check Boxes are some items you can check multiple of them while browsing the form.

They are just like below:



What topic whould you like to read about?

Home Repair ☑

Gardening □

Child Care □

To create a check box, use the **type="checkbox"** attribute with the <input> tag, such as in the following:

```
<input type="checkbox" name="repair">
```

By default, value of checkboxes will be received at the target page by this format:

```
repair=on
```

You can change that default by specifying a **value** attribute. For example, you could report the word Yes for the check box, as shown here:

```
<input type="checkbox" name="repair" value="Yes">
```

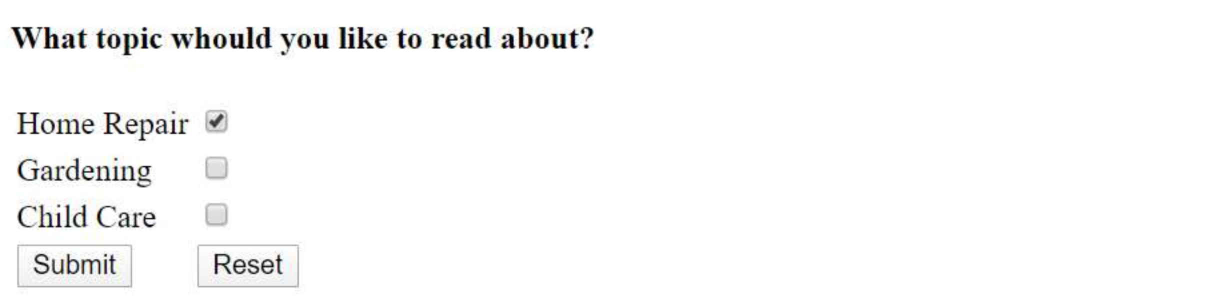
By default, check boxes appear unselected; users must click each check box to select it. But you can change the default value of a checkbox in a way they will be appeared checked; by below syntax:

```
<input type="checkbox" name="repair" checked>
```

The next example will show you how to arrange checkboxes and submit button into table:

Example.33 Checkboxes and tables

```
<body>
      <h4>What topic whould you like to read about?</h4>
      <form method="get" action="targetpage.php">
            <table>
                  <tbody>
                        <tr>
                              <td><label>Home Repair</label></td>
                              <td><input  type="checkbox"  name="repair"
checked></td>
                        </tr>
                        <tr>
                              <td><label>Gardening</label></td>
                              <td><input                 type="checkbox"
name="gardening"></td>
                        </tr>
                        <tr>
                              <td><label>Child Care</label></td>
                              <td><input                 type="checkbox"
name="childcare"></td>
                        </tr>
                        <tr>
                              <td><input type="submit"></td>
                              <td><input type="reset"></td>
                        </tr>
                  </tbody>
            </table>

      </form>
</body>
```



What topic whould you like to read about?

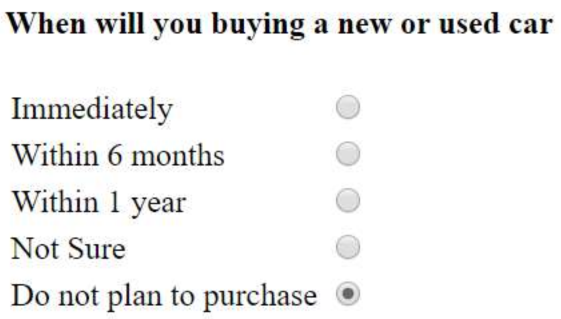
Home Repair ☑

Gardening □

Child Care □

Submit Reset

Use **option buttons** (also called **radio buttons**) to present a group of mutually exclusive options. When you select an option button, all the other option buttons in the group are cleared.

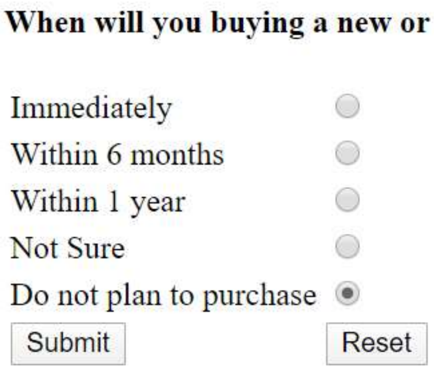


When will you buying a new or used car

Immediately ○
Within 6 months ○
Within 1 year ○
Not Sure ○
Do not plan to purchase ●

To create a group of option buttons, choose a **name** for the **group**. You will specify the same name in the **name** attribute for each individual button. Use the **value** attribute (which will be different for each button in the set) to specify the value that will be reported for the group in the form results.

Example.34 Result: Radio buttons and tables



When will you buying a new or used car

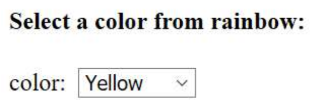
Immediately ○
Within 6 months ○
Within 1 year ○
Not Sure ○
Do not plan to purchase ●

Example.34 Radio buttons and tables

```
<body>

      <form action="targetPage.php" method="post">
      <h4>When will you buying a new or used car</h4>
      <table>
            <tbody>
                  <tr>
                        <td>Immediately</td>
                        <td><input type="radio" name="time" 
value="immediately"></td>
                  </tr>
                  <tr>
                        <td>Within 6 months</td>
                        <td><input type="radio" name="time" 
value="6months"></td>
                  </tr>
                  <tr>
                        <td>Within 1 year</td>
                        <td><input type="radio" name="time" 
value="1year"></td>
                  </tr>
                  <tr>
                        <td>Not Sure</td>
                        <td><input type="radio" name="time" 
value="notsure"></td>
                  </tr>
                  <tr>
                        <td>Do not plan to purchase</td>
                        <td><input type="radio" name="time" 
value="noplan" checked></td>
                  </tr>
                  <tr>
                        <td><input type="submit"></td>
                        <td><input type="reset"></td>
                  </tr>
            </tbody>
      </table>


      </form>
</body>
```

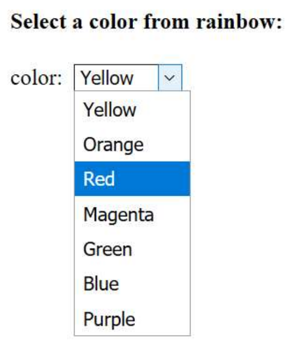
7.7. Creating Lists

For situations involving many options, consider a **list**, also called a **menu**. A list can contain as many options as needed, yet it takes up very little space on the form.



Select a color from rainbow:

color: Yellow



Select a color from rainbow:

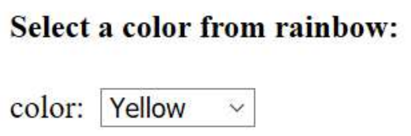
color: Yellow

To create a list, start with a two-sided <select> tag. Within it, place each option in its own <option> tag. Place the text that you want to appear on the list between the opening and closing <option> tags.

To make a specific option as the default value, put selected keyword within its tag.

Example. 35: Menus

```
      <body>
            <form action="targetPage.php" method="post">
                  <h4>Select a color from rainbow:</h4>
                  <p>color:&nbsp;
                  <Select name="color" size="1">
                        <option value="1">Yellow</option>
                        <option value="2">Orange</option>
                        <option value="3" selected>Red</option>
                        <option value="4">Magenta</option>
                        <option value="5">Green</option>
                        <option value="6">Blue</option>
                        <option value="7">Purple</option>
                  </Select>
                  </p>
            </form>
      </body>
```



Select a color from rainbow:

color: Yellow

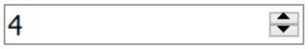
7.8. Additional Input Types in HTML5

HTML5 provides several other field types that can add that extra bit of polish to your forms. If the user's browser does not support them, it renders and treats them as text fields, so you can use them freely without worrying about backward compatibility.

Password input type which enables you to put a password field to hide the characters while the user inputs its value (usually shows ● or * instead of input characters):

```
<input type="password" name="pass">
```

Spin boxes are used to increment numeric values:

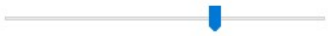


4

```
<input type="number" name="copies" min="0" max="100" step="1" value="1">
```

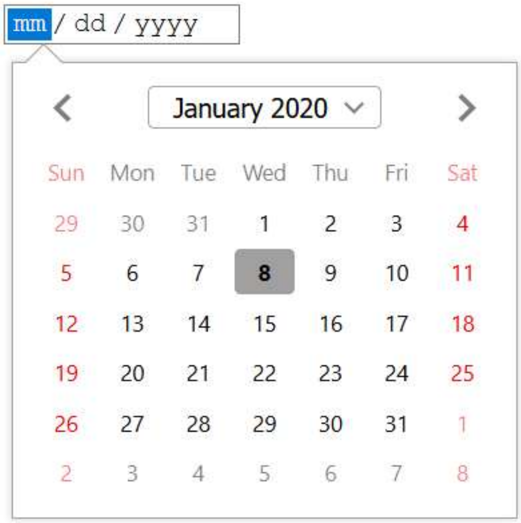
The min and max attributes control the minimum and maximum values, respectively. The step attribute specifies how much the value increments or decrements when you click the up or down arrow buttons. The value attribute specifies the default value.

A **slider** is a sliding bar that you can drag from side to side. Its type is range, and its attributes are nearly identical to those for a spin box.



```
<input type="range" name="count" min="1" max="4" step="1" value="1">
```

date picker pops up a calendar on which the user can click and select a date. Use the date type to get a date picker in HTML5-compliant browsers, as follows:



mm / dd / yyyy

January 2020

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

```
<input type="date">
```

For a standard date, use type=”date”. This enables the user to select a specific date from a calendar. You can also use any of the following types for other date and time-related selections:

- Type=”month” selects an entire month
- Type=”week” selects an entire week
- Type=”time” selects a time only (no date)
- Type=”datetime” select both a date and a time
- Type=”datetime-local” selects both a date and time using the user’s local time zone

8. Incorporating Multimedia

The addition of the <video> and <audio> tags in HTML5 makes the process of delivering and playing video and audio more straightforward. Playing multimedia will get easier over time as newer browsers support the tags and people upgrade their older browsers.

8.1. Embedding Video Clips

At a basic level, the <video> tag looks like this:

```
<video src="myvideo.ogv"></video>
```

There are several attributes and different ways to use the <video> tag that make it more configurable for your needs and the needs of your audience. Several attributes are helpful, including:

- autoplay
- controls
- height
- loop
- preload
- width

use the width and height attributes to set the width and the height of the video display area on the page:

```
<video src="myvideo.ogv" width="320" height="240"></video>
```

The controls attribute determines whether a default set of playback controls should be visible within the browser.

```
<video src="myvideo.ogv" controls></video>
```

The preload attribute tells the browser to begin downloading the video immediately when the element is encountered.

```
<video src="myvideo.ogv" preload></video>
```

The loop attribute tells the browser to restart the video immediately when it has finished playing, as shown here:

```
<video src="myvideo.ogv" loop></video>
```

Finally, the autoplay attribute makes the video automatically play when the page is loaded. For most purposes, this is generally a bad idea from a usability standpoint:

```
<video src="myvideo.ogv" autoplay></video>
```

Putting it together, a real-world video element looks like this:

```
<video src="myvideo.ogv" width="320" height="240" controls></video>
```

8.2. Different Encoding Remedy

The <video> tag enables more than one source (via the source element) which you can capitalize on by including links to multiple versions of a video. You can also add a type attribute to tell the browser a bit more about the video file to which you are linking.

```
<video width="320" height="240" controls>
        <source src="myvideo.mp4" type="video/mp4">
        <source src="myvideo.ogv" type="video/ogg">
        <source src="myvideo.webm" type="video/webm">
</video>
```

8.3. Incorporating Audio on a Web Page

Like the <video> tag, the <audio> tag supports multiple sources. With no common format, you will need to encode the audio multiple times to try to get the audio out to the widest possible audience. Also like the <video> tag, the <audio> tag supports attributes such as controls, autoplay, loop, and preload. Therefore, the syntax for the <audio> tag is essentially the same as the syntax for the <video> tag.

Here's an example that shows the <audio> tag with two files, which are called with the help of the <source> element that you saw earlier in the video section of this chapter:

```
<audio controls>
        <source src="myaudio.mp3">
        <source src="myaudio.ogg">
</audio>
```

8.4. Embedding Image

The HTML <img> element embeds an image into the document.

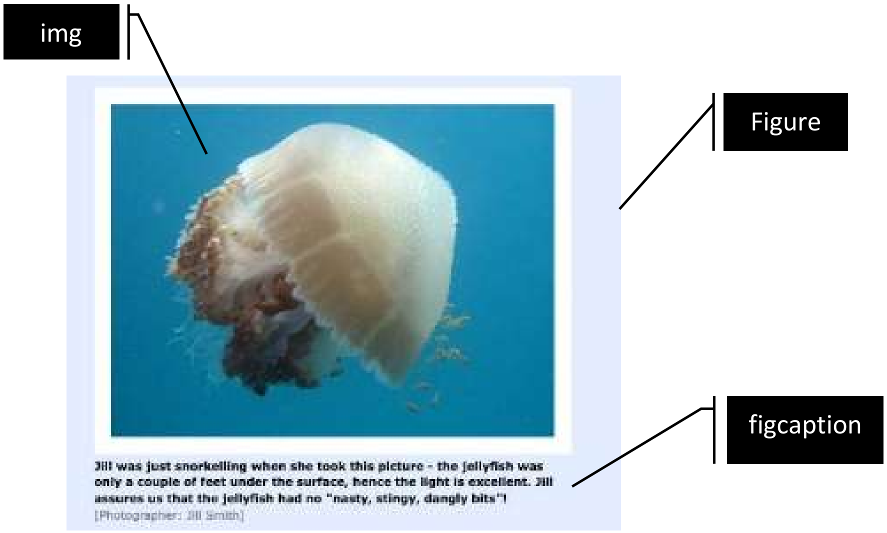
```
<img class="fit-picture"
    src="/media/examples/grapefruit-slice-332-332.jpg"
    alt="Grapefruit slice atop a pile of other slices">
```

- The **src** attribute is required, and contains the path to the image you want to embed.
- The **alt** attribute holds a text description of the image, which is not mandatory but is incredibly useful for accessibility — screen readers read this description out to their users so they know what the image means. Alt text is also displayed on the page if the image cannot be loaded for some reason: for example, network errors, content blocking, or linkrot.

8.5. Figure

Usually a <figure> is an image, illustration, diagram, code snippet, etc., that is referenced in the main flow of a document, but that can be moved to another part of the document or to an appendix without affecting the main flow.

A caption can be associated with the <figure> element by inserting a <figcaption> inside it (as the first or the last child). The first <figcaption> element found in the figure is presented as the figure's caption.



Jill was just snorkelling when she took this picture - the jellyfish was only a couple of feet under the surface, hence the light is excellent. Jill assures us that the jellyfish had no "nasty, stingy, dangly bits"!
[Photographer: Jill Smith]

```
<figure>
        <img src="pic_trulli.jpg" alt="Trulli" style="width:100%">
        <figcaption>Fig.1 - Trulli, Puglia, Italy.</figcaption>
</figure>
```

9. Frames

An HTML iframe is used to display a web page within a web page.

9.1. HTML iFrame Syntax

The HTML <iframe> tag specifies an inline frame.

```
<iframe src="url" title="description"></iframe>
```

Hint: It is a good practice to always include a title attribute for the <iframe>. This is used by screen readers to read out what the content of the iframe is.

Use the height and width attributes to specify the size of the iframe.

The height and width are specified in pixels by default:

Example. 36: Following is the example to create three horizontal frames

```
<!DOCTYPE html>
<html>
        <head>
                <title>iFrame Example</title>
                <style type="text/css">
                        iframe {
                          border-style: none;
                          width: 200px;
                          height: 200px;
                        }
                </style>
        </head>
        <body>
                <h2>Custom Iframe Border</h2>
                <p>With CSS, you can also change the size, style and color of the
                iframe's border:</p>

                <iframe src="http://www.google.com" title="Iframe
                Example"></iframe>
        </body>
</html>
```

9.2. Iframe as a Target for Links

An iframe can be used as the target frame for a link. The target attribute of the link must refer to the name attribute of the iframe:

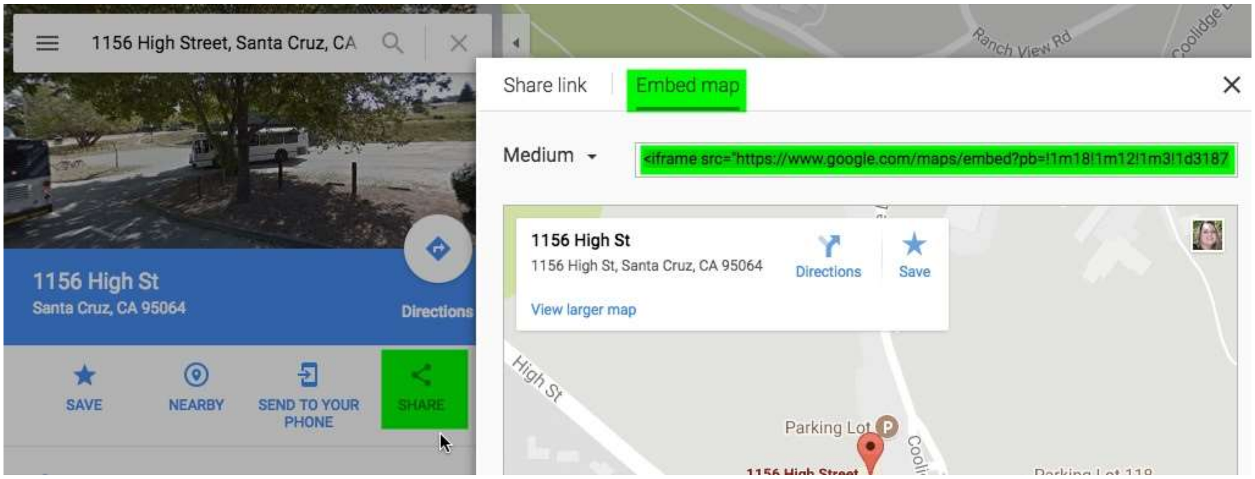
Example. 37: Using iFrame as target of a Link

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

                <h2>Iframe - Target for a Link</h2>

                <iframe src="demo_iframe.htm" name="iframe_a" height="300px"
                width="100%" title="Iframe Example"></iframe>

                <a href="sample_page.html" target="iframe_a">Sample Page </a>

                <p>When the target attribute of a link matches the name of an iframe,
                the link will open in the iframe.</p>

        </body>
</html>
```

9.3. How to use iFrame to insert Google Map

You can embed a simple map, a set of driving directions, a local search, or maps created by other users. Here's how:

1. Once you have your Google Map created, ensure that the map you'd like to embed appears in the current map display.
2. Click "Share" at the right of the page.
3. In the box that pops up, click "Embed"
4. Copy the entire HTML "<iframe> code string and paste it into the HTML code of your web page.
5. Click update and then while still in edit mode, in the Main Content area of the page, type a space. This tells the system that there is content on the page and will allow it to display "iframe" code.



1156 High Street, Santa Cruz, CA

1156 High St
Santa Cruz, CA 95064

Medium

```
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3187
```

1156 High St
1156 High St, Santa Cruz, CA 95064

10. Upload and Download

Here are some tags and attributes related to upload and download:

10.1. Upload

The <input type="file"> defines a file-select field and a "Browse" button for file uploads.

To define a file-select field that allows multiple files to be selected, add the multiple attribute.

```
<label for="myfile">Select a file:</label>
<input type="file" id="myfile" name="myfile">
```

10.2. Downlaod

The download attribute specifies that the target (the file specified in the href attribute) will be downloaded when a user clicks on the hyperlink.

The optional value of the download attribute will be the new name of the file after it is downloaded. There are no restrictions on allowed values, and the browser will automatically detect the correct file extension and add it to the file (.img, .pdf, .txt, .html, etc.).

If the value is omitted, the original filename is used.

Download file when clicking on the link (instead of navigating to the file):

```
<a href="/images/myimage.jpg" download>
```

Specify a value for the download attribute, which will be the new filename of the downloaded file ("w3logo.jpg" instead of "myimage.jpg"):

```
<a href="/images/myimage.jpg" download="w3logo">
```

11. SVG

11.1. What is SVG?

- SVG stands for Scalable Vector Graphics
- SVG is used to define vector-based graphics for the Web
- SVG defines the graphics in XML format
- Every element and every attribute in SVG files can be animated
- SVG is a W3C recommendation
- SVG integrates with other W3C standards such as the DOM and XSL

11.2. SVG Advantages

- SVG images can be created and edited with any text editor
- SVG images can be searched, indexed, scripted, and compressed
- SVG images are scalable
- SVG images can be printed with high quality at any resolution
- SVG images are zoomable
- SVG graphics do NOT lose any quality if they are zoomed or resized
- SVG is an open standard
- SVG files are pure XML

11.3. SVG in HTML

You can embed SVG elements directly into your HTML pages.

Example. 38: Simple SVG graphic

```
<!DOCTYPE html>
<html>
        <body>
                <h1>My first SVG</h1>
                <svg width="100" height="100">
                         <circle cx="50" cy="50" r="40" stroke="green" stroke-
                        width="4" fill="yellow" />
                </svg>
        </body>
</html>
```

11.4. SVG Shapes

SVG has some predefined shape elements that can be used by developers:

- Rectangle <rect>
- Circle <circle>
- Ellipse <ellipse>
- Line <line>

11.4.1. SVG Rectangle - <rect>

```
<svg width="400" height="110">
    <rect width="300" height="100" style="fill:rgb(0,0,255);stroke-
   width:3;stroke:rgb(0,0,0)" />
</svg>
```

11.4.2. SVG Circle - <circle>

```
<svg height="100" width="100">
    <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />
</svg>
```

11.4.3. SVG Ellipse - <ellipse>

```
<svg height="140" width="500">
    <ellipse cx="200" cy="80" rx="100" ry="50"
   style="fill:yellow;stroke:purple;stroke-width:2" />
</svg>
```

11.4.4. SVG Line - <line>

```
<svg height="210" width="500">
    <line x1="0" y1="0" x2="200" y2="200" style="stroke:rgb(255,0,0);stroke-
   width:2" />
</svg>
```