



Overview



HTML Basics



HTML5

TABLE DES MATIÈRES

Overview

• What is HTML ?	04
• Basic HTML Document Structure.....	05
• Creating Your First HTML Page	09
• Creating a Blog	12

HTML Basics

• Paragraphs	14
• Text Formatting	17
• Headings, Lines, Comments.....	19
• Blog Project About me	22
• Elements	23
• Attributes.....	24
• Images	28
• Links.....	32
• Lists.....	35
• Blog Project : My Skills	37
• Tables.....	38
• Blog Project : My Schedule.....	42
• Inline and Block Elements.....	43
• Forms.....	46
• Blog Project : Contact Form.....	51
• HTML Colors.....	52
• Frames.....	56
• Blog Project : Putting it All Together.....	58

HTML5

• Introducing HTML5	60
• Content Models.....	62
• HTML5 Page Structure	65
• header, nav & footer.....	66
• Article, section & aside.....	69
• The audio Element.....	72
• The video Element.....	75
• The progress Element.....	77
• Web Storage API	78
• Geolocation API	81
• Drag&Drop API	84
• SVG	86
• SVG Animations & paths.....	94
• Canvas.....	96
• SVG vs. Canvas.....	100
• Canvas Transformations	101
• HTML5 Forms, Part 1	106
• HTML5 Forms, Part 2	111



Overview

Welcome to HTML!

HTML stands for HyperText Markup Language.

Unlike a scripting or programming language that uses scripts to perform functions, a markup language uses tags to identify content.

Here is an example of an HTML tag:

```
<p> I'm a paragraph </p>
```

Try It Yourself

This course is a combination of interactive quizzes and exercises to facilitate speed and quality of learning.
Tap the **Continue** button to go to your first quiz!

2046 COMMENTS



The Web Structure

The ability to code using HTML is essential for any web professional. Acquiring this skill should be the starting point for anyone who is learning how to create content for the web.

Modern Web Design

HTML: Structure

CSS: Presentation

JavaScript: Behavior

PHP or similar: Backend

CMS: Content Management

HTML is easy to learn. So don't wait! Dive right in!

938 COMMENTS



The <html> Tag

Although various versions have been released over the years, HTML basics remain the same.

The structure of an HTML document has been compared with that of a sandwich. As a sandwich has two slices of bread, the HTML document has opening and closing HTML tags.

These tags, like the bread in a sandwich, surround everything else:

```
<html>  
...  
</html>
```

Everything in an HTML document is surrounded by the <html> tag.

576 COMMENTS



The <head> Tag

Immediately following the opening HTML tag, you'll find the **head** of the document, which is identified by opening and closing head tags.

The head of an HTML file contains all of the **non-visual elements** that help make the page work.

```
<html>  
  <head>...</head>  
</html>
```

The head section elements will be discussed later.

448 COMMENTS



The <body> Tag

The **body** tag follows the head tag.

All visual-structural elements are contained within the body tag.

Headings, paragraphs, lists, quotes, images, and links are just a few of the elements that can be contained within the body tag.

Basic HTML Structure:

```
<html>  
<head>  
</head>  
<body>  
</body>  
</html>
```

Try It Yourself

The <body> tag defines the main content of the HTML document.

483 COMMENTS

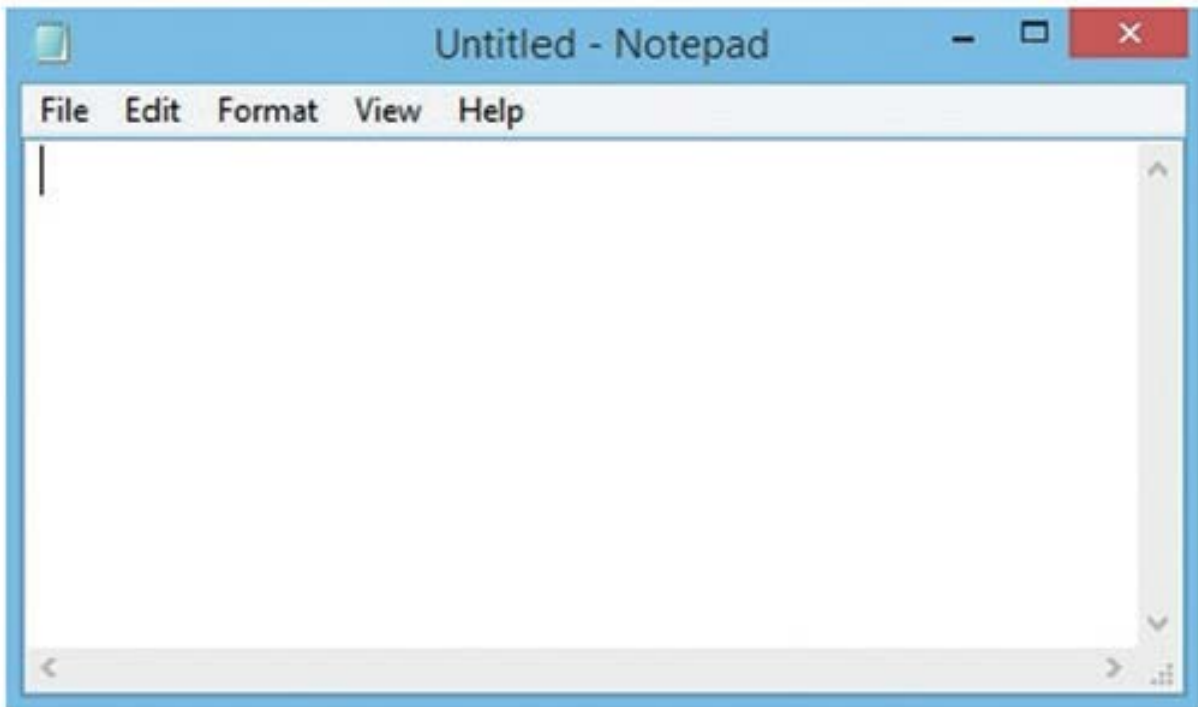


Q&A



The HTML File

HTML files are text files, so you can use any **text editor** to create your first webpage. There are some very nice HTML editors available; you can choose the one that works for you. For now let's write our examples in **Notepad**.



You can run, save, and share your HTML codes on our **Code Playground**, without installing any additional software.

880 COMMENTS



Q&A



The HTML File

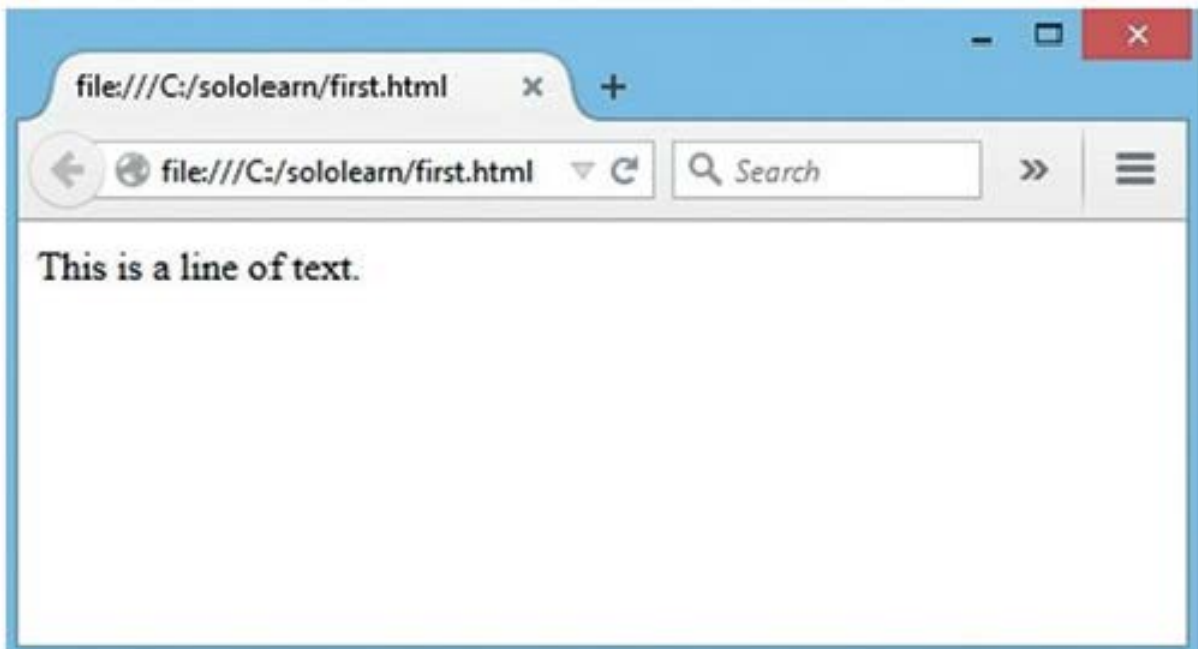
Add the basic HTML structure to the text editor with "This is a line of text" in the body section.

```
<html>
  <head>
  </head>
  <body>
    This is a line of text.
  </body>
</html>
```

Try It Yourself

In our example, the file is saved as **first.html**

When the file is opened, the following result is displayed in the web browser:



Don't forget to save the file. HTML file names should end in either **.html** or **.htm**

568 COMMENTS



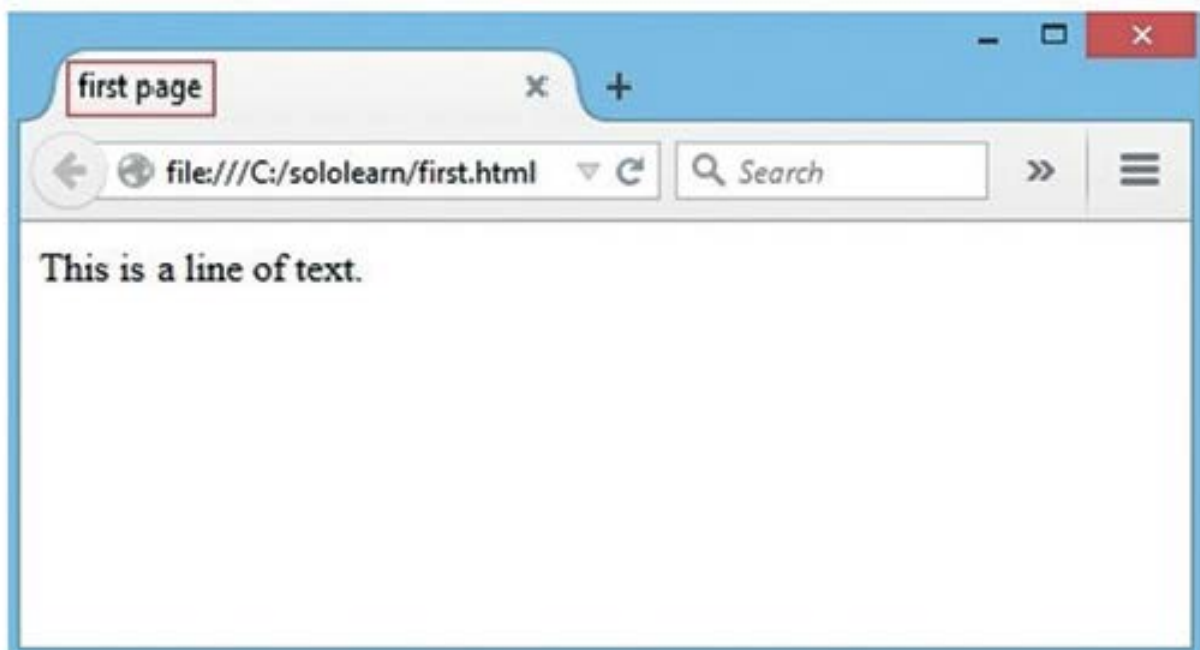
The <title> Tag

To place a title on the tab describing the web page, add a <title> element to your head section:

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    This is a line of text.
  </body>
</html>
```

Try It Yourself

This will produce the following result:



The title element is important because it describes the page and is used by search engines.

494 COMMENTS



Creating a Blog

Throughout this course, we'll help you practice and create your own unique blog project, so you'll retain what you've learned and be able to put it to use. Just keep going and follow the instructions in the **TASK** section. This is what your finished blog page will look like.

TASK: Tap **Try It Yourself** to see the code and its output.

```
<html>
  <head>
    <title>My Blog</title>
  </head>
  <body>
  ...
```

Try It Yourself

Don't be afraid of long codes. By the time you complete the course, everything will make complete sense and look highly doable. We guarantee it!

TASK:

1. Open the code.
2. On the top header, change the name to your own name.
3. Change the page title. Remember, the page title is located inside the <title> tag in the <head> of the page.

527 COMMENTS



Q&A





HTML Basics

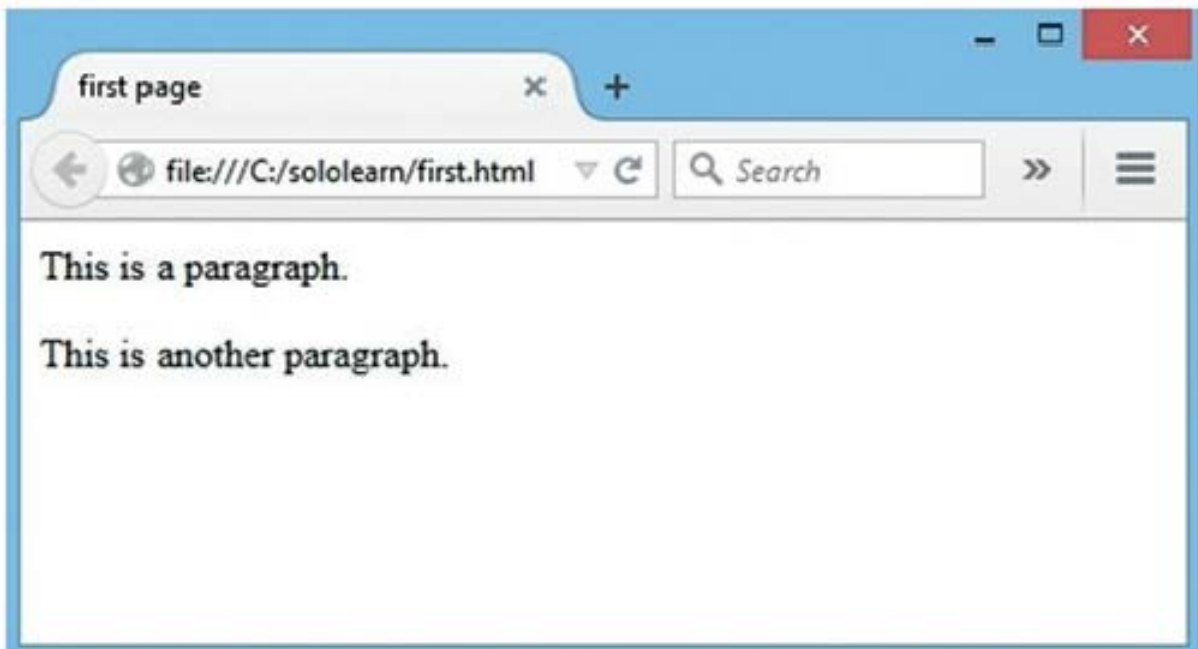
The <p> Element

To create a paragraph, simply type in the <p> element with its opening and closing tags:

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    <p>This is a paragraph. </p>
    <p>This is another paragraph. </p>
  </body>
</html>
```

Try It Yourself

The result:



Browsers automatically add an empty line before and after a paragraph.

684 COMMENTS



Single Line Break

Use the `
` tag to add a single line of text without starting a new paragraph:

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    <p>This is a paragraph.</p>
    <p>This is another paragraph. </p>
    <p>This is <br /> a line break </p>
  </body>
</html>
```

Try It Yourself

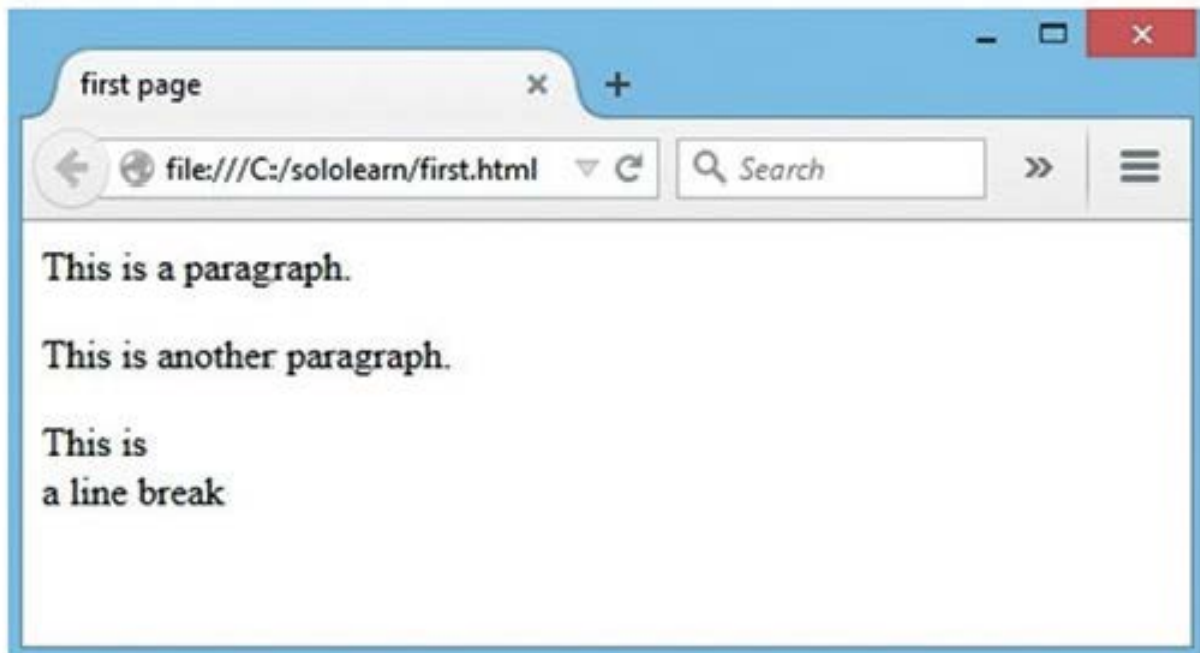
The `
` element is an empty HTML element. It has no end tag.

557 COMMENTS



Single Line Break

Opening the HTML file in the browser shows that a single line break has been added to the paragraph:



The `
` element has no end tag.

269 COMMENTS



Formatting Elements

In HTML, there is a list of elements that specify text style.

Formatting elements were designed to display special types of text:

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    <p>This is regular text </p>
    <p><b> bold text </b></p>
    <p><big> big text </big></p>
    <p><i> italic text </i></p>
    <p><small> small text </small></p>
    <p><strong> strong text </strong></p>
    <p><sub> subscripted text </sub></p>
    <p><sup> superscripted text </sup></p>
    <p><ins> inserted text </ins></p>
    <p><del> deleted text </del></p>
  </body>
</html>
```

Try It Yourself

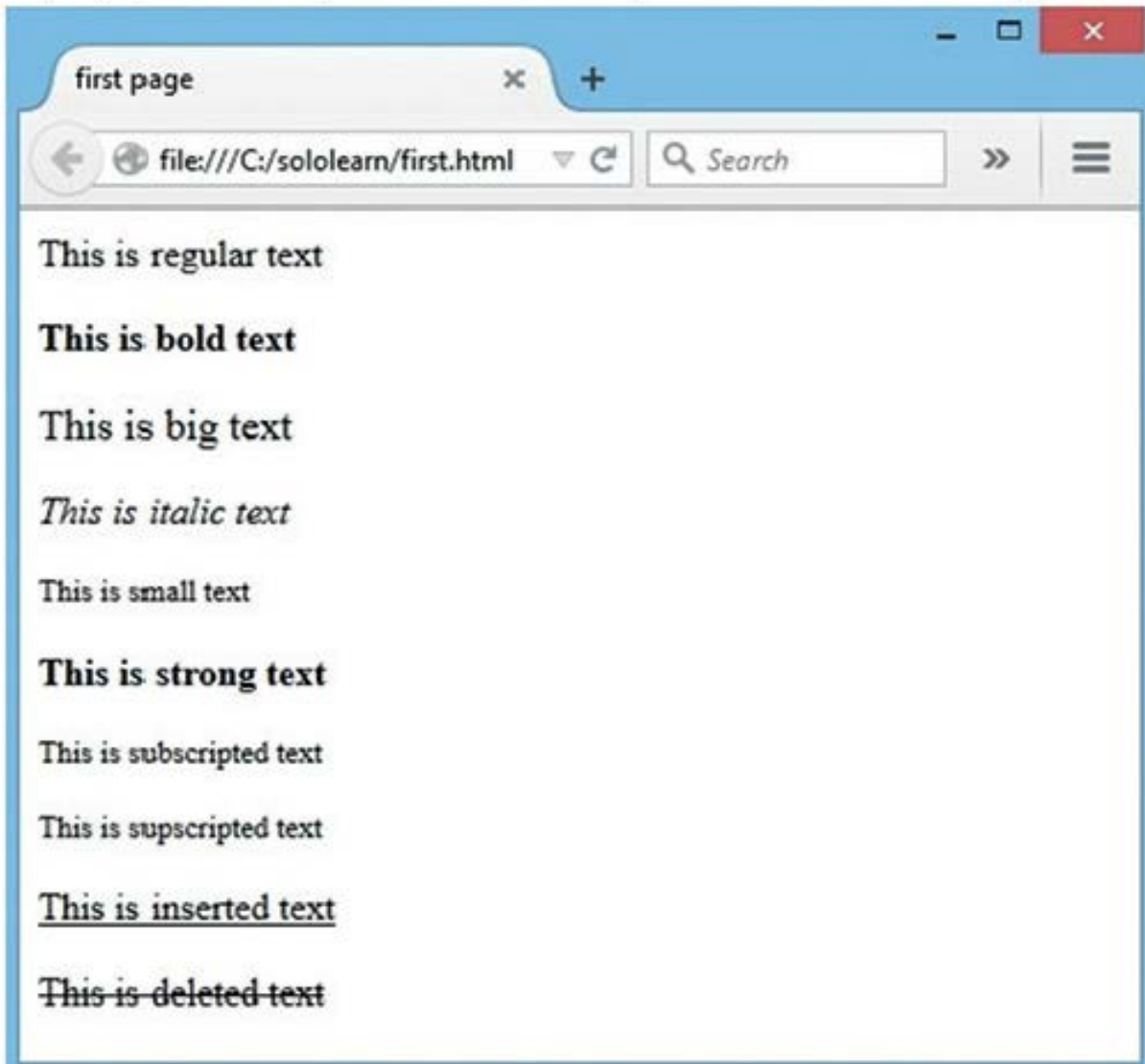
The `` tag is a phrase tag. It defines important text.

903 COMMENTS



Formatting Elements

Each paragraph in the example is formatted differently to demonstrate what each tag does:



Browsers display `` as ``, and `` as `<i>`. However, the meanings of these tags differ: `` and `<i>` define bold and italic text, respectively, while `` and `` indicate that the text is "important".

449 COMMENTS



HTML Headings

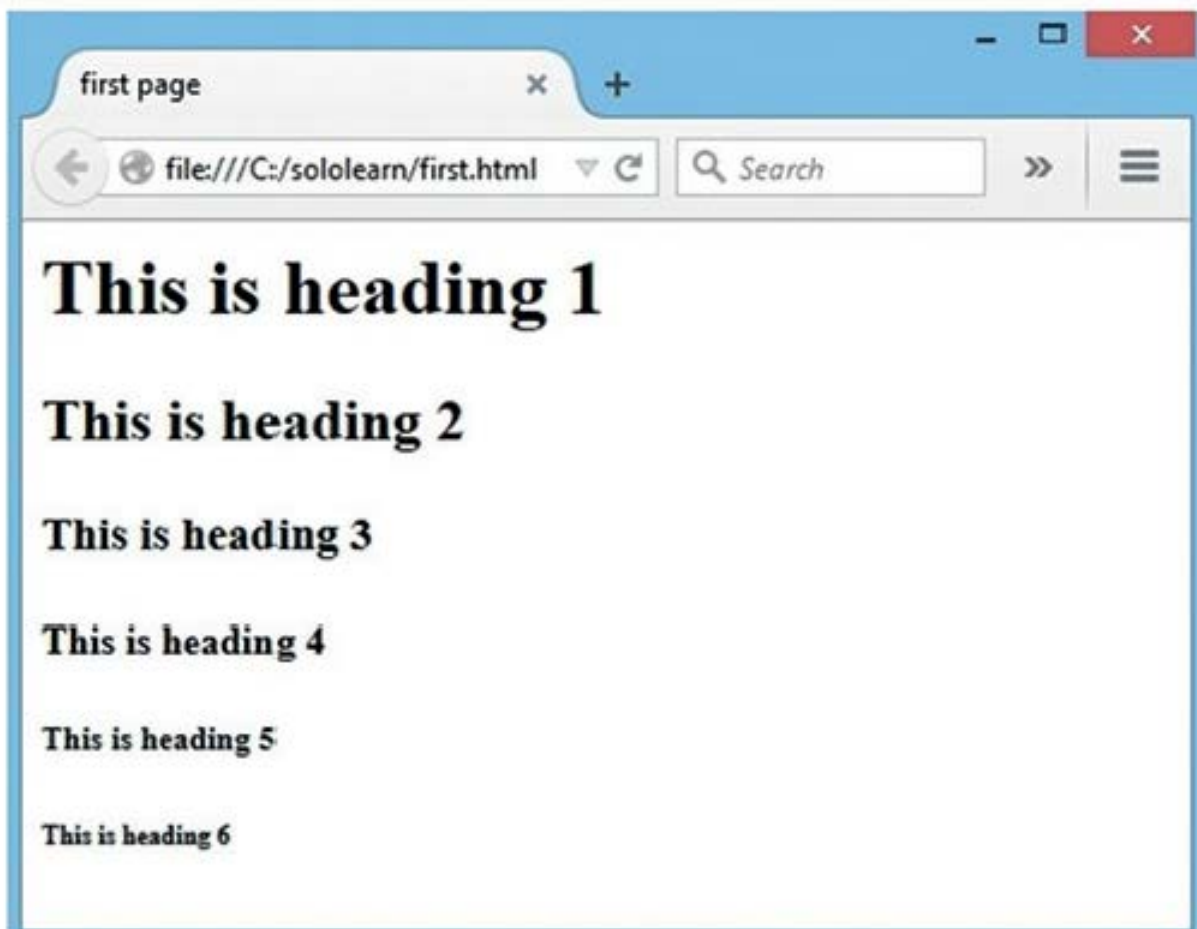
HTML includes six levels of headings, which are ranked according to importance. These are `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`.

The following code defines all of the headings:

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
  </body>
</html>
```

Try It Yourself

Result:



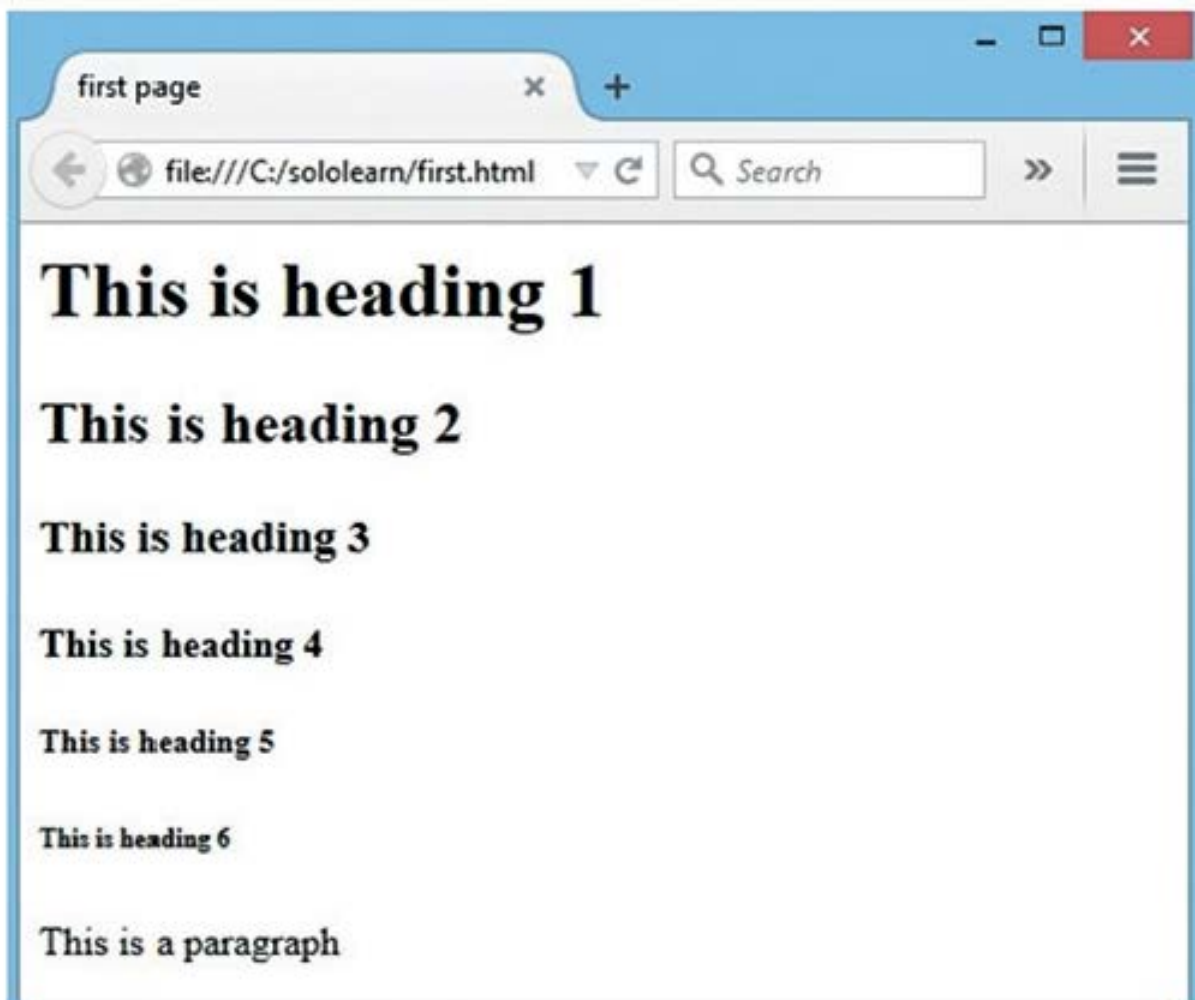
Horizontal Lines

To create a horizontal line, use the `<hr />` tag.

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
    <p>This is a paragraph </p>
    <hr />
    <p>This is a paragraph </p>
  </body>
</html>
```

Try It Yourself

Result:



Comments

The browser does not display comments, but they help document the HTML and add descriptions, reminders, and other notes.

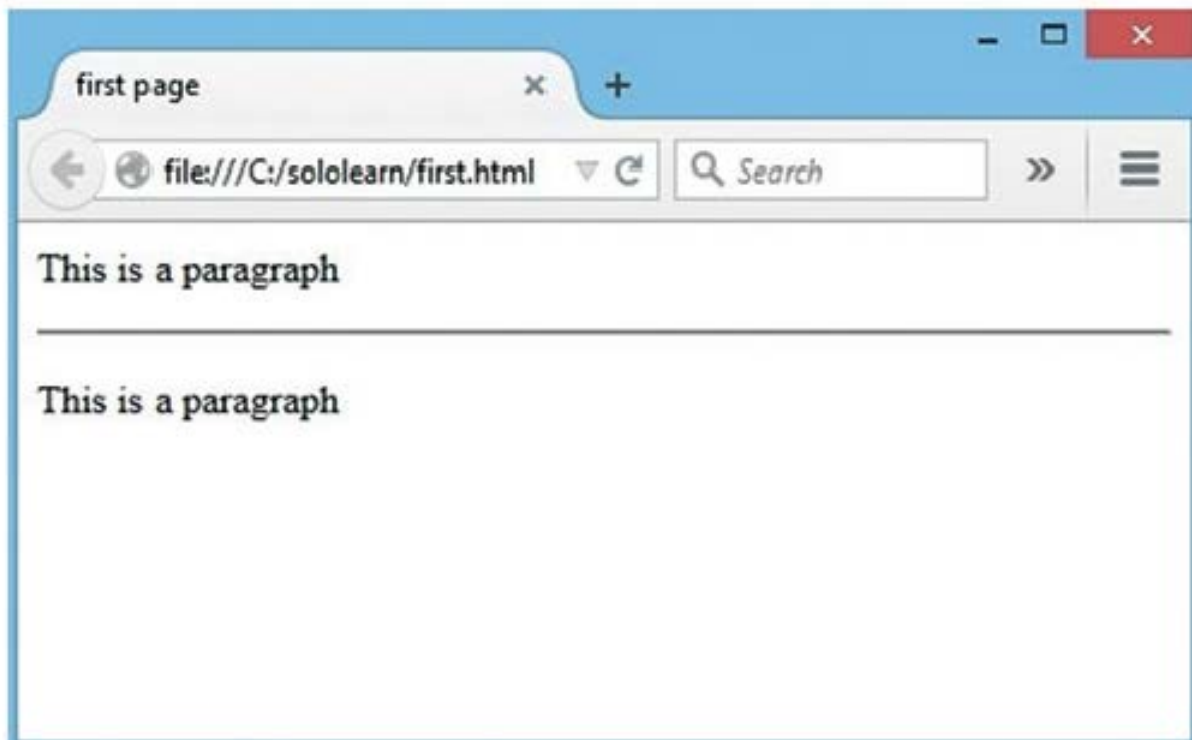
```
<!-- Your comment goes here -->
```

Example:

```
<html>  
  <head>  
    <title>first page</title>  
  </head>  
  <body>  
    <p>This is a paragraph </p>  
    <hr />  
    <p>This is a paragraph </p>  
    <!-- This is a comment -->  
  </body>  
</html>
```

Try It Yourself

Result:



As you can see, the comment is not displayed in the browser.

There is an exclamation point (!) in the opening tag, but not in the closing tag.

Formatting Text

Let's get back to our blog project.

The **About Me** section will contain a heading that's wrapped in a `<h1>` tag, along with two paragraphs that format text using the tags you've just learned.

Let's take a look at the code:

```
<h1><span>About Me</span></h1>
<p>
Hey! I'm <strong>Alex</strong>. Coding has changed my world ...
</p>
<p class="quote">"Declare variables, not war"</p>
```

Try It Yourself

You may have noticed that we've also used some CSS to add colors and styles to the page. When you complete HTML, we strongly recommend our free CSS Tutorial!

TASK:

1. Create your own About Me section by changing the text.
2. Play around with the code; experiment with formatting text.

812 COMMENTS



Q&A



HTML Elements

HTML documents are made up of HTML elements.

An HTML element is written using a **start tag** and an **end tag**, and with the **content** in between.

HTML documents consist of nested HTML elements. In the example below, the body element includes the `<p>` tags, the `
` tag and the content, "This is a paragraph".

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    <p>This is a paragraph <br /></p>
  </body>
</html>
```

Try It Yourself

Some HTML elements (like the `
` tag) do not have end tags.

381 COMMENTS



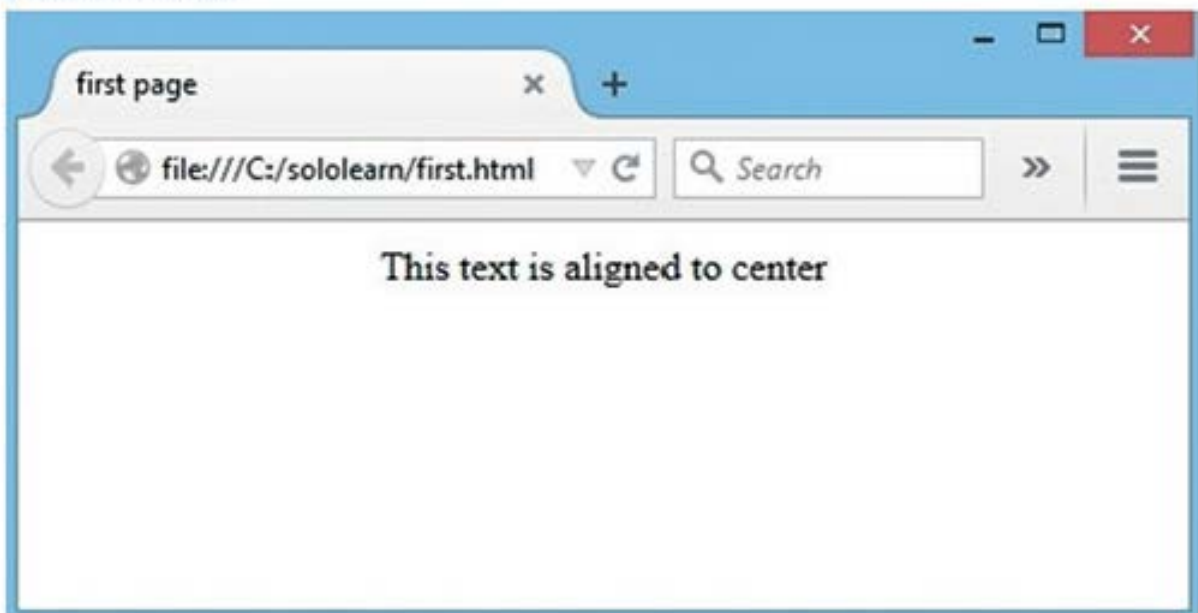
HTML Attributes

Attributes provide **additional information** about an element or a tag, while also **modifying** them. Most attributes have a value; the value modifies the **attribute**.

```
<p align="center">  
  This text is aligned to center  
</p>
```

Try It Yourself

In this example, the value of "center" indicates that the content within the p element should be aligned to the center.



Attributes are always specified in the start tag, and they appear in name="value" pairs.

621 COMMENTS



Attribute Measurements

As an example, we can modify the horizontal line so it has a width of 50 pixels.

This can be done by using the width attribute:

```
<hr width="50px" />
```

Try It Yourself

An element's width can also be defined using percentages:

```
<hr width="50%" />
```

Try It Yourself

An element's width can be defined using pixels or percentages.

632 COMMENTS



The Align Attribute

The **align** attribute is used to specify how the text is aligned.

In the example below, we have a paragraph that is aligned to the center, and a line that is aligned to the right.

```
<html>
  <head>
    <title>Attributes</title>
  </head>
  <body>
    <p align="center">This is a text <br />
      <hr width="10%" align="right" /> This is also a text.
    </p>
  </body>
</html>
```

Try It Yourself

The align attribute of `<p>` is not supported in HTML5.

582 COMMENTS



Attributes

You may be wondering what happens if you try to apply contradictory attributes within the same element.

```
<p_align="center">  
  This is a text.  
  <hr width="50%" align="left" />  
</p>
```

Try It Yourself



The align attribute of `<p>` is not supported in HTML5.

364 COMMENTS



The Tag

The tag is used to insert an image. It contains only attributes, and does not have a closing tag.

The image's URL (address) can be defined using the src attribute.

The HTML image syntax looks like this:

```

```

The alt attribute specifies an alternate text for an image.

846 COMMENTS



Image Location

You need to put in the **image location** for the src attribute that is between the quotation marks.

For example, if you have a photo named "tree.jpg" in the same folder as the HTML file, your code should look like this:

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    
  </body>
</html>
```

Try It Yourself

In case the image cannot be displayed, the alt attribute specifies an alternate text that describes the image in words. The alt attribute is **required**.

948 COMMENTS



Image Resizing

To define the image size, use the width and height attributes. The value can be specified in **pixels** or as a **percentage**:

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    
    <!-- or -->
    
  </body>
</html>
```

Try It Yourself

Loading images takes time. Using large images can slow down your page, so use them with care.

609 COMMENTS



Image Border

By default, an image has no borders. Use the border attribute within the image tag to create a border around the image.

```

```

Try It Yourself

By default, Internet Explorer 9, and its earlier versions, display a border around an image unless a border attribute is defined.

457 COMMENTS



The <a> Tag

Links are also an integral part of every web page. You can add links to text or images that will enable the user to click on them in order to be directed to another file or webpage. In HTML, links are defined using the `<a>` tag.

Use the `href` attribute to define the link's destination address:

```
<a href=""></a>
```

To link an image to another document, simply nest the `` tag inside `<a>` tags.

846 COMMENTS



Creating Your First Link

In the example below, a link to SoloLearn's website is defined:

```
<a href="http://www.sololearn.com"> Learn Playing </a>
```

Try It Yourself

Once the code has been saved, "Learn Playing" will display as a link:

[Learn Playing](http://www.sololearn.com)

Clicking on "**Learn Playing**" redirects you to www.sololearn.com

Links can be either absolute or relative.

656 COMMENTS



The target Attribute

The **target** attribute specifies where to open the linked document.

Giving a **_blank** value to your attribute will have the link open in a new window or new tab:

```
<a href="http://www.sololearn.com" target="_blank">  
  Learn Playing  
</a>
```

Try It Yourself

A visited link is underlined and purple.

523 COMMENTS



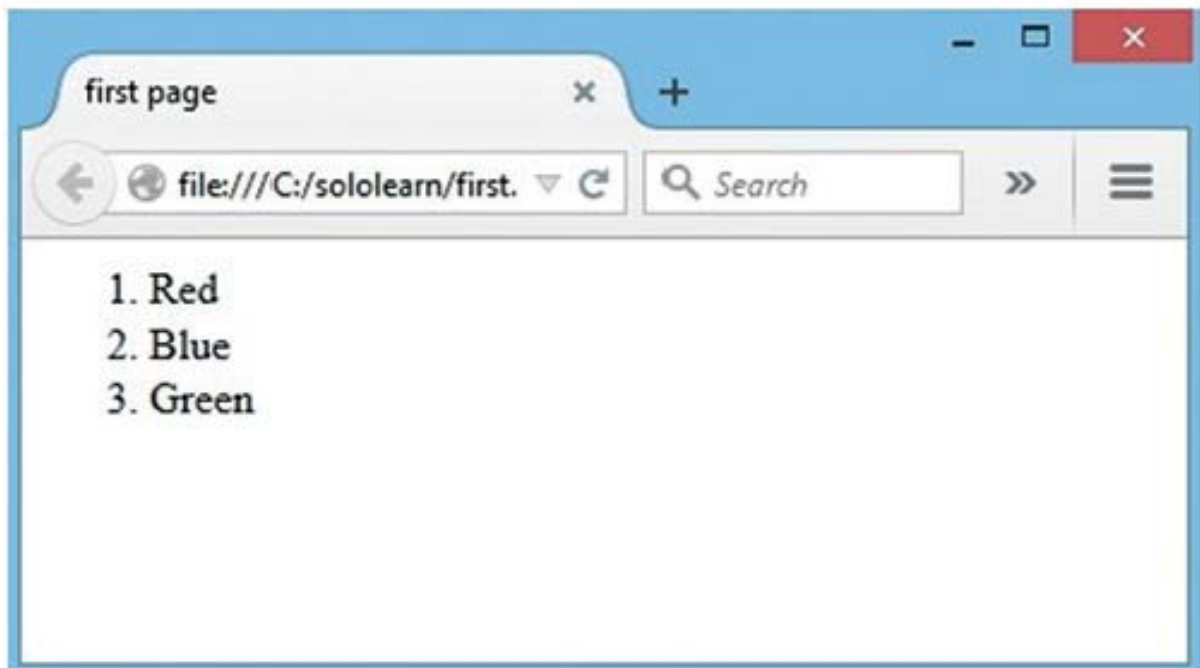
HTML Ordered Lists

An ordered list starts with the `` tag, and each list item is defined by the `` tag. Here is an example of an ordered list:

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    <ol>
      <li>Red</li>
      <li>Blue</li>
      <li>Green</li>
    </ol>
  </body>
</html>
```

Try It Yourself

Result:



The list items will be automatically marked with numbers.

700 COMMENTS



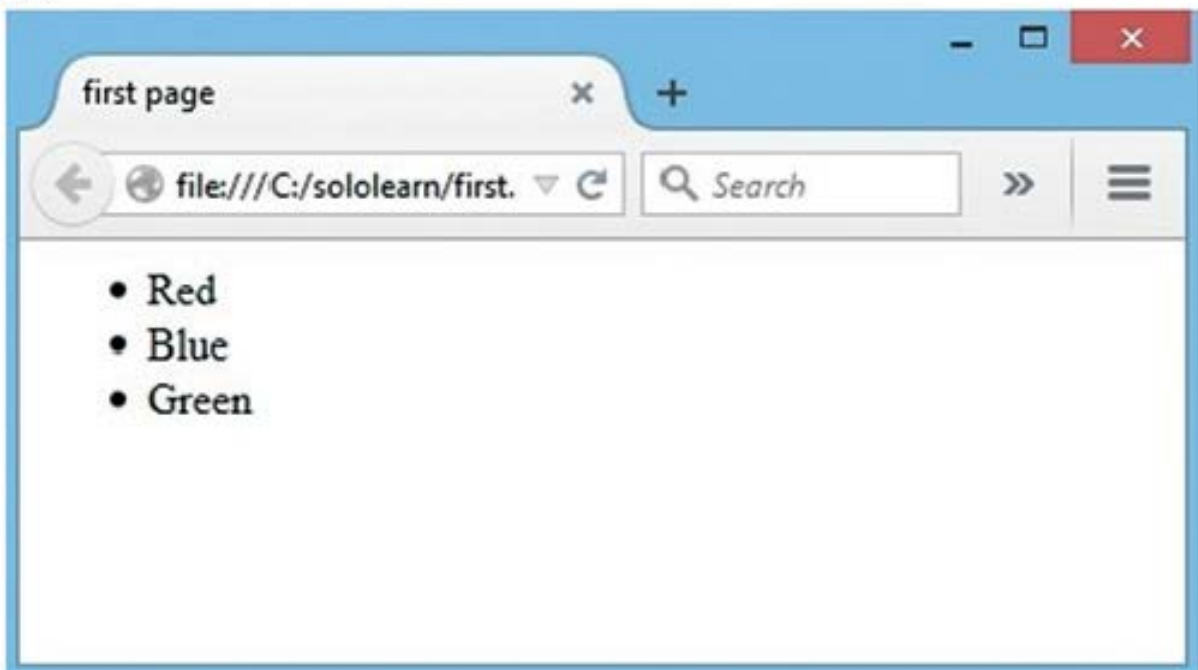
HTML Unordered List

An unordered list starts with the `` tag.

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body>
    <ul>
      <li>Red</li>
      <li>Blue</li>
      <li>Green</li>
    </ul>
  </body>
</html>
```

Try It Yourself

Result:



The list items will be marked with bullets.

321 COMMENTS



My Skills

Back to our blog! Let's create the **My Skills** section, which is an unordered list of languages you know.

Reminder: Use the `` tag, in which each item is represented by the `` tag, to create an unordered list. Take a look at the code:

```
<h1><span>My Skills</span></h1>
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ul>
```

Try It Yourself

We'll learn how to add progress bars to each of the skills in the following module.

TASK:

1. Create your own My Skills section.
2. Use the `<a>` tag to make each list item a link.

595 COMMENTS



Q&A



Creating a Table

Tables are defined by using the `<table>` tag.

Tables are divided into table rows with the `<tr>` tag.

Table rows are divided into table columns (table data) with the `<td>` tag.

Here is an example of a table with **one row and three columns**:

```
<table>  
<tr>  
  <td></td>  
  <td></td>  
  <td></td>  
</tr>  
</table>
```

Table data tags `<td>` act as data containers within the table.

They can contain all sorts of HTML elements, such as text, images, lists, other tables, and so on.

338 COMMENTS



The border and colspan Attributes

A border can be added using the **border** attribute:

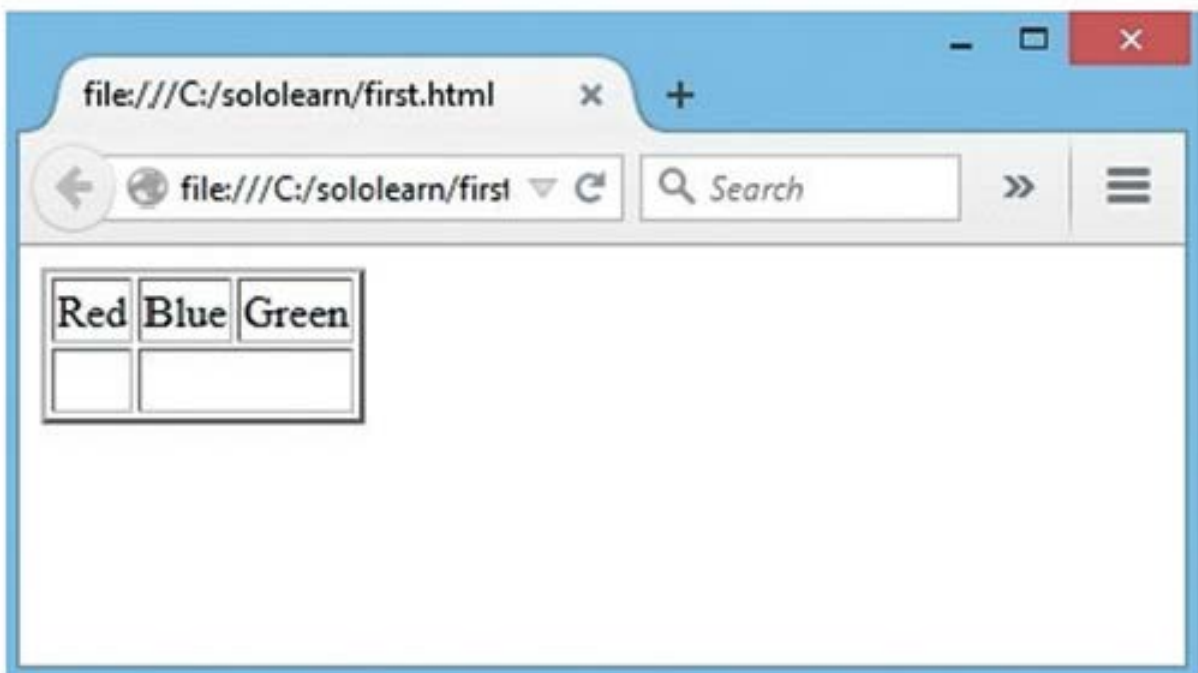
```
<table border="2">
```

A table **cell** can span two or more columns:

```
<table border="2">
  <tr>
    <td>Red</td>
    <td>Blue</td>
    <td>Green</td>
  </tr>
  <tr>
    <td><br /></td>
    <td colspan="2"><br /></td>
  </tr>
</table>
```

Try It Yourself

Result:



The border attribute is not supported in HTML5.

765 COMMENTS

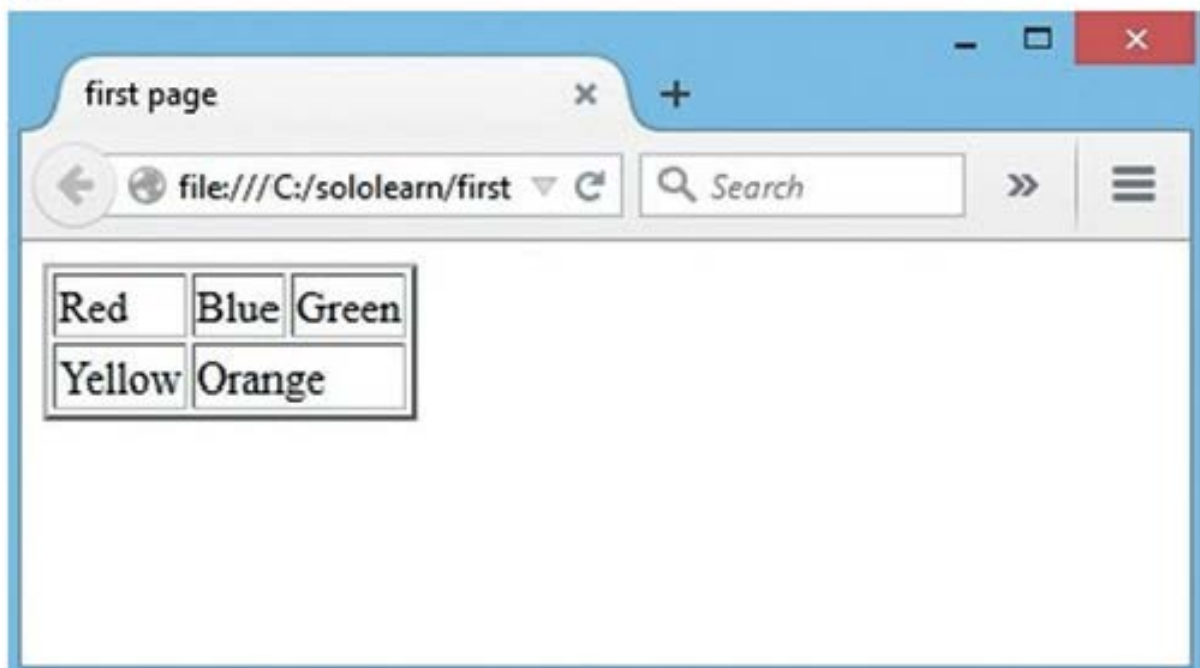
Colspan Color

The example below demonstrates the **colspan** attribute in action:

```
<table border="2">
  <tr>
    <td>Red</td>
    <td>Blue</td>
    <td>Green</td>
  </tr>
  <tr>
    <td>Yellow</td>
    <td colspan="2">Orange</td>
  </tr>
</table>
```

Try It Yourself

Result:



You can see that the cell containing "Orange" spans two cells.

To make a cell span more than one row, use the **rowspan** attribute.

277 COMMENTS



The align and bgcolor Attributes

To change your table's position, use the **align** attribute inside your table tag:

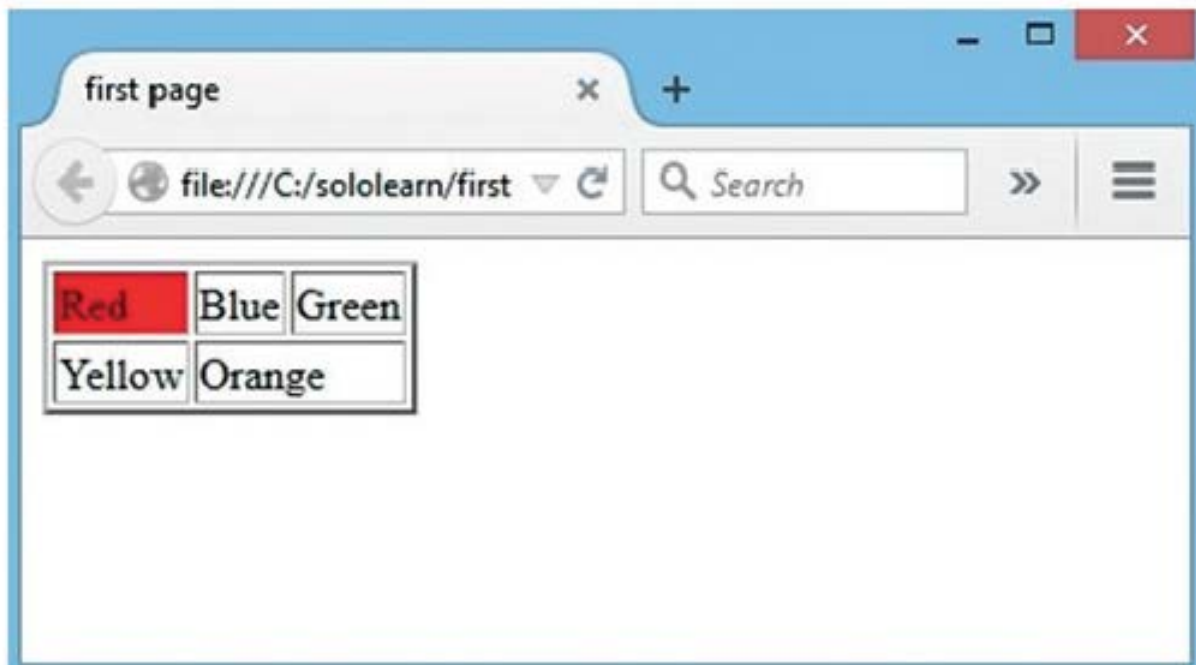
```
<table align="center">
```

Now let's specify a background color of red for a table cell. To do that, just use the **bgcolor** attribute.

```
<table border="2">
  <tr>
    <td bgcolor="red">Red</td>
    <td>Blue</td>
    <td>Green</td>
  </tr>
  <tr>
    <td>Yellow</td>
    <td colspan="2">Orange</td>
  </tr>
</table>
```

Try It Yourself

Result:



In the case of styling elements, CSS is more effective than HTML. Try our free "Learn CSS" course to learn more about CSS and styles.

437 COMMENTS

My Schedule

Use table tags to add a table to your blog that represents your daily learning schedule. The `<th>` tags represent the table headers. Take a look at the following code:

```
<h1><span>My Coding Schedule</span></h1>
<table>
<tr>
  <th>Day</th>
  <th>Mon</th>
  <th>Tue</th>
  <th>Wed</th>
  <th>Thu</th>
  <th>Fri</th>
</tr>
<tr>
  <td>8-8:30</td>
  <td class="selected">Learn</td>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
</tr>
<tr>
  <td>9-10</td>
  <td></td>
  <td class="selected">Practice</td>
  <td></td>
  <td></td>
  <td></td>
</tr>
<tr>
  <td>1-1:30</td>
  <td></td>
  <td></td>
  <td class="selected">Play</td>
  <td></td>
  <td></td>
</tr>
<tr>
  <td>3:45-5</td>
  <td></td>
  <td></td>
  <td></td>
  <td class="selected">Code</td>
  <td></td>
</tr>
<tr>
  <td>6-6:15</td>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
  <td class="selected">Discuss</td>
</tr>
```

Types of Elements

In HTML, most elements are defined as **block level** or **inline** elements.

Block level elements start from a new line.

For example: `<h1>`, `<form>`, ``, ``, ``, `<p>`, `<pre>`, `<table>`, `<div>`, etc.

Inline elements are normally displayed without line breaks.

For example: ``, `<a>`, ``, ``, `<input>`, ``, ``, etc.

The `<div>` element is a block-level element that is often used as a **container for other HTML elements**.

When used together with some CSS styling, the `<div>` element can be used to style blocks of content:

```
<html>
<body>
  <h1>Headline</h1>
  <div style="background-color:green; color:white; padding:20px;">
    <p>Some paragraph text goes here.</p>
    <p>Another paragraph goes here.</p>
  </div>
</body>
</html>
```

Try It Yourself

Output

Headline

Some paragraph text goes here.

Another paragraph goes here.

Similarly, the `` element is an inline element that is often used as a container for some text. When used together with CSS, the `` element can be used to style **parts of the text**:

```
<html>
<body>
  <h2>Some
    <span style="color:red">Important</span>
  Message</h2>
</body>
</html>
```

Try It Yourself

Output

Some **Important** Message

Summary

The `<div>` element defines a **block-level** section in a document.
The `` element defines an **inline** section in a document.

750 COMMENTS



Q&A



Types of Elements

Other elements can be used either as block level elements or inline elements. This includes the following elements:

APPLET - embedded Java applet

IFRAME - Inline frame

INS - inserted text

MAP - image map

OBJECT - embedded object

SCRIPT - script within an HTML document

You can insert inline elements inside block elements. For example, you can have multiple `` elements inside a `<div>` element.

Inline elements **cannot** contain any block level elements.

337 COMMENTS



Q&A



The <form> Element

HTML forms are used to collect information from the user.

Forms are defined using the `<form>` element, with its opening and closing tags:

```
<body>  
  <form>...</form>  
</body>
```

Use the **action** attribute to point to a webpage that will load after the user submits the form.

```
<form action="http://www.sololearn.com">  
</form>
```

Usually the form is submitted to a web page on a web server.

524 COMMENTS



The method and name Attributes

The **method attribute** specifies the HTTP method (**GET** or **POST**) to be used when forms are submitted (see below for description):

```
<form action="url" method="GET">
```

```
<form action="url" method="POST">
```

When you use **GET**, the form data will be visible in the page address.

Use **POST** if the form is updating data, or includes sensitive information (passwords). **POST** offers better security because the submitted data is not visible in the page address.

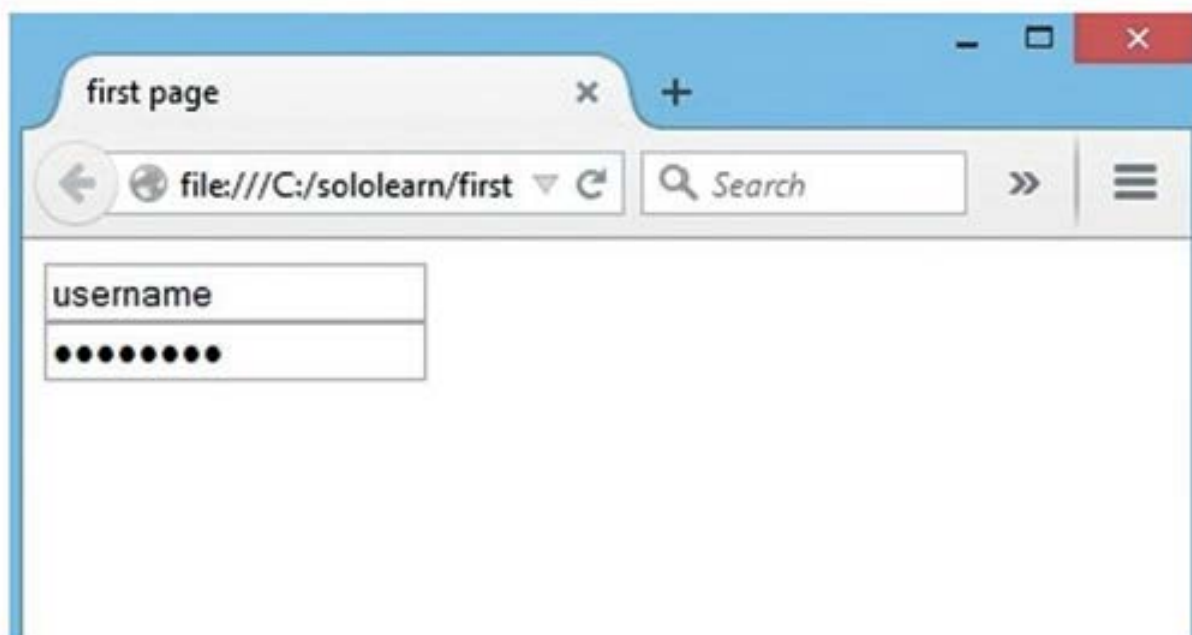
To take in user input, you need the corresponding form elements, such as text fields. The `<input>` element has many variations, depending on the type attribute. It can be a text, password, radio, URL, submit, etc.

The example below shows a form requesting a username and password:

```
<form>  
  <input type="text" name="username" /><br />  
  <input type="password" name="password" />  
</form>
```

Try It Yourself

Result:



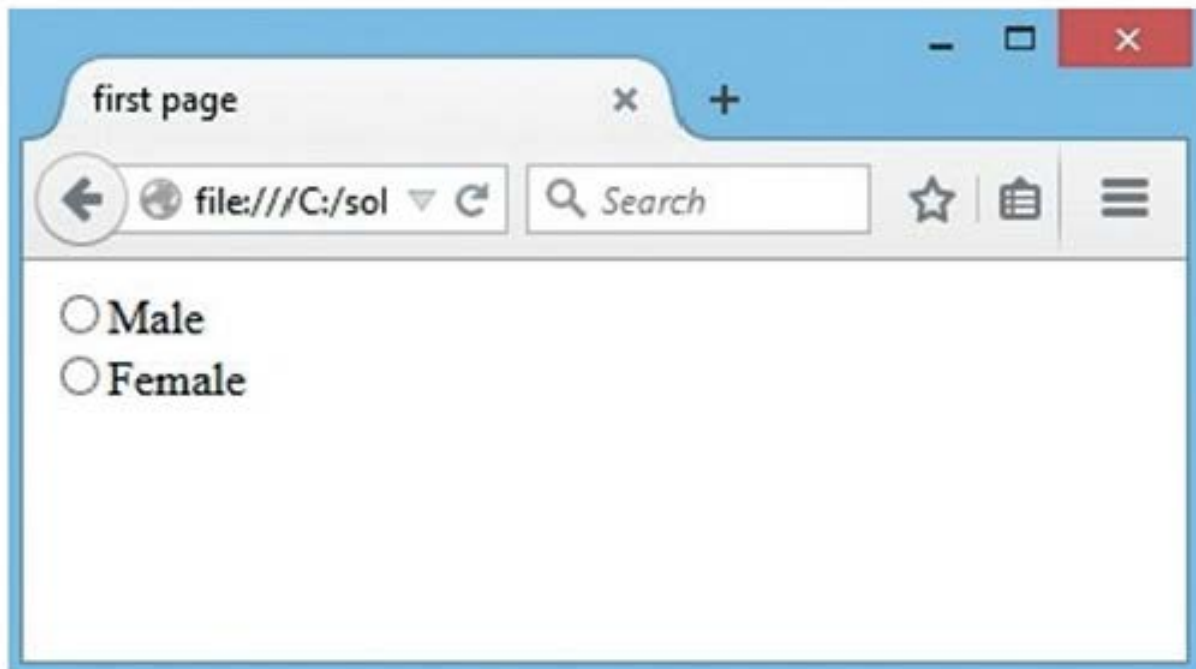
Form Elements

If we change the input type to **radio**, it allows the user select only one of a number of choices:

```
<input type="radio" name="gender" value="male" /> Male <br />  
<input type="radio" name="gender" value="female" /> Female <br />
```

Try It Yourself

Result:

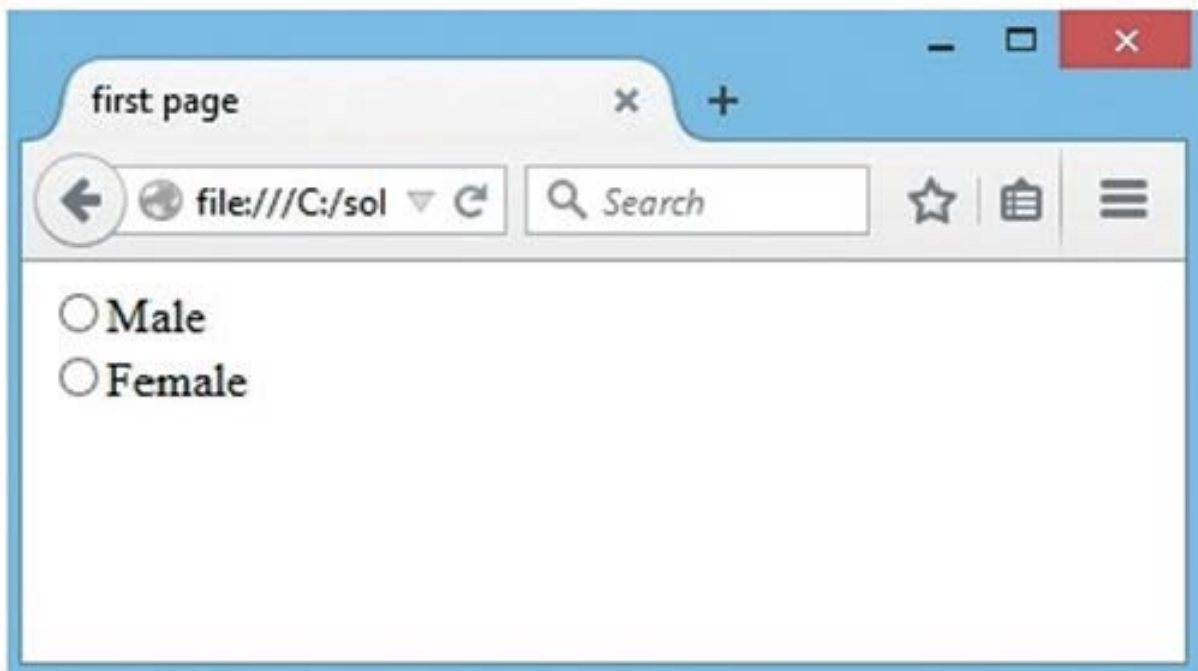


The type "checkbox" allows the user to select more than one option:

```
<input type="checkbox" name="gender" value="1" /> Male <br />  
<input type="checkbox" name="gender" value="2" /> Female <br />
```

Try It Yourself

Result:

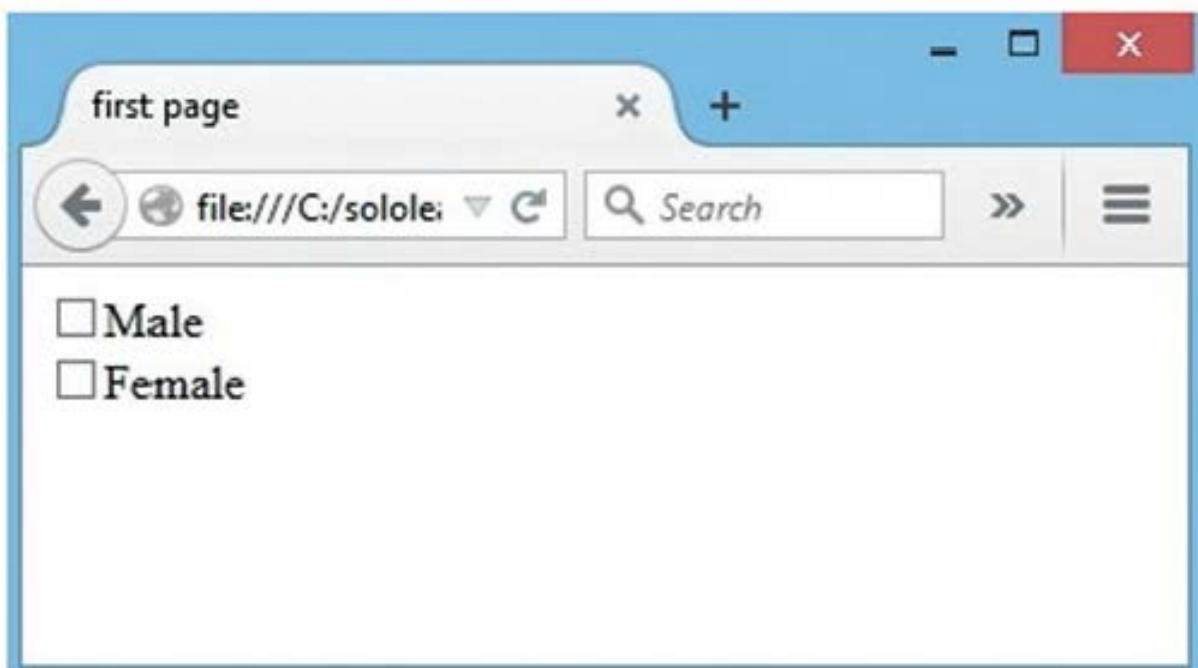


The type "checkbox" allows the user to select more than one option:

```
<input type="checkbox" name="gender" value="1" /> Male <br />  
<input type="checkbox" name="gender" value="2" /> Female <br />
```

Try It Yourself

Result:



The `<input>` tag has no end tag.

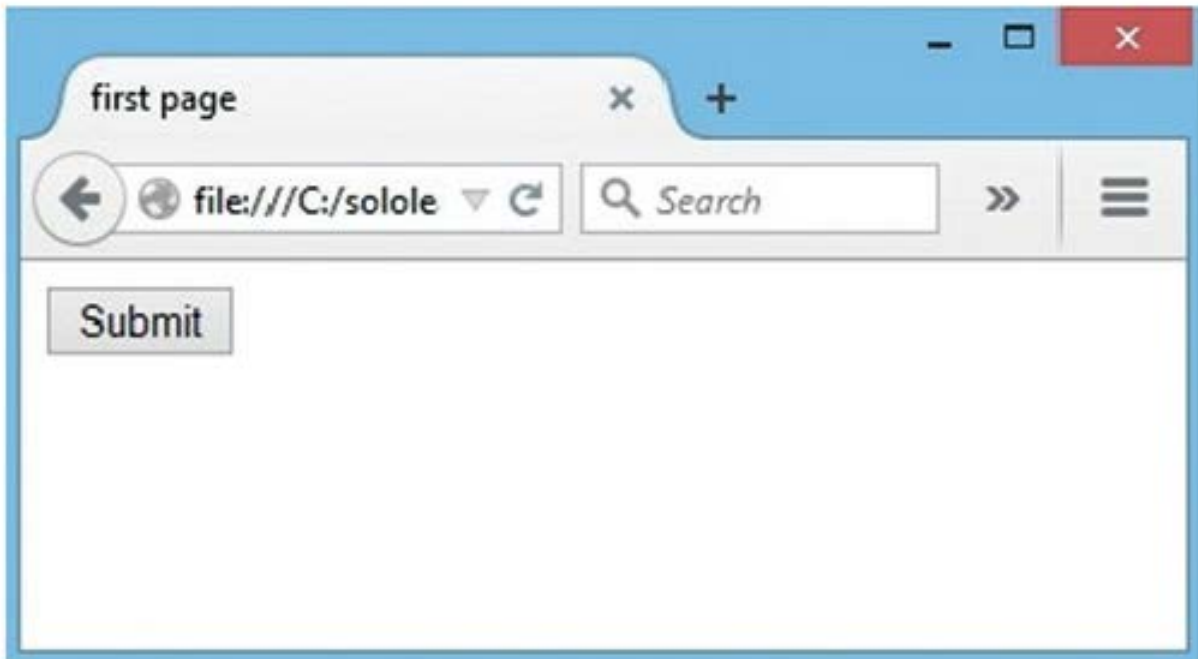
Form Elements

The submit button **submits a form** to its action attribute:

```
<input type="submit" value="Submit" />
```

Try It Yourself

Result:



After the form is submitted, the data should be processed on the server using a programming language, such as PHP.

365 COMMENTS



Contact Form

Next, we'll create a **Contact Form** for your blog. The form will include Name, Email, and Message fields. We'll also add a Submit button.

Check out the code:

```
<h1><span>Contact Me</span></h1>  
<form>  
  <input name="name" type="text" /><br/>  
  <input name="email" type="email" /><br/>  
  <textarea name="message"></textarea>  
  <input type="submit" value="SEND" class="submit" />  
</form>
```

Try It Yourself

This is just a static HTML page, so it won't work to actually submit the form. You'd need to create the server-side code in order to submit a real form and process the data. To learn how, complete SoloLearn's PHP course once you've completed the HTML and CSS courses.

TASK: Check out the code and customize the form to fit your needs!

422 COMMENTS



Q&A



HTML Colors!

HTML colors are expressed as hexadecimal values.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

As you can see, there are 16 values there, 0 through F. Zero represents the lowest value, and F represents the highest.

312 COMMENTS

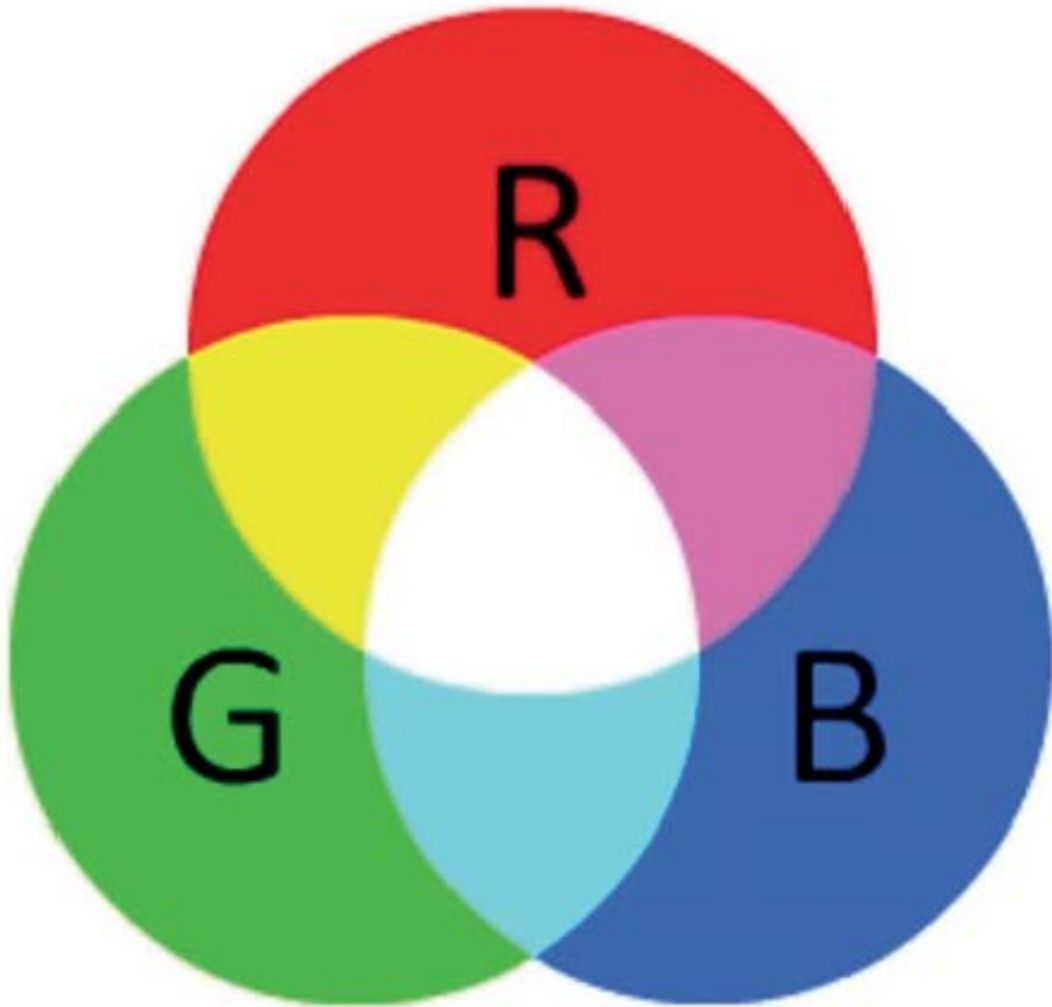


HTML Color Model

Colors are displayed in combinations of **red**, **green**, and **blue** light (RGB).

Hex values are written using the hashtag symbol (#), followed by either three or six hex characters.

As shown in the picture below, the circles overlap, forming new colors:



RGB color values are supported in all browsers.


264 COMMENTS



Color Values

All of the possible **red**, **green**, and **blue** combinations potentially number over 16 million.

Here are only a few of them:

#FF0000 

#00FF00 

#0000FF 

#000000 

#777777 

#FFFFFF 

We can mix the colors to form additional colors:

Orange and red mix:

#C1250F 

Hexadecimal color values are supported in all browsers.

350 COMMENTS



Background and Font Colors

The `bgcolor` attribute can be used to change the web page's background color.

This example would produce a dark blue background with a white headline:

```
<html>
  <head>
    <title>first page</title>
  </head>
  <body bgcolor="#000099">
    <h1>
      <font color="#FFFFFF"> White headline </font>
    </h1>
  </body>
</html>
```

Try It Yourself

Result:



The `color` attribute specifies the color of the text inside a `` element.

329 COMMENTS



The <frame> Tag

A page can be divided into frames using a special frame document.

The <frame> tag defines one specific window (frame) within a <frameset>. Each <frame> in a <frameset> can have different attributes, such as border, scrolling, the ability to resize, etc.

The <frameset> element specifies the number of columns or rows in the frameset, as well as what percentage or number of pixels of space each of them occupies.

```
<frameset cols="100, 25%, *"></frameset>  
<frameset rows="100, 25%, *"></frameset>
```

The <frameset> tag is not supported in HTML5.

320 COMMENTS



Working with Frames

Use the `<noresize>` attribute to specify that a user cannot resize a `<frame>` element:

```
<frame noresize="noresize">
```

Frame content should be defined using the `src` attribute.

Lastly, the `<noframes>` element provides a way for browsers that do not support frames to view the page. The element can contain an alternative page, complete with a body tag and any other elements.

```
<frameset cols="25%,50%,25%">
  <frame src="a.htm" />
  <frame src="b.htm" />
  <frame src="c.htm" />
  <noframes>Frames not supported!</noframes>
</frameset>
```

The `<frame>` tag is not supported in HTML5.

198 COMMENTS



Blog Project

To finalize our blog, we'll use a frame to embed a YouTube video. We'll also create a **Follow Me** section that includes links at the end of the page. Check out the final code:

```
...  
<div class="section">  
  <h1><span>My Media</span></h1>  
  <iframe height="150" width="300"  
  src="https://www.youtube.com/embed/Q6_5lnVJZ88" allowfullscreen frameborder="0">  
</iframe>  
</div>  
...
```

Try It Yourself

TASK:

1. Finalize your blog page.
2. Share your code creation with the community, get feedback, and receive upvotes!

734 COMMENTS



Q&A





HTML5

HTML5

When writing HTML5 documents, one of the first new features that you'll notice is the doc type declaration:

```
<!DOCTYPE HTML>
```

The character encoding (charset) declaration is also simplified:

```
<meta charset="UTF-8">
```

New Elements in HTML5

`<article>`, `<aside>`, `<audio>`, `<canvas>`, `<datalist>`, `<details>`, `<embed>`, `<footer>`, `<header>`, `<nav>`, `<output>`, `<progress>`, `<section>`, `<video>`, and even more!

The default character encoding in HTML5 is UTF-8.

1189 COMMENTS



New in HTML5

Forms

- The Web Forms 2.0 specification allows for creation of more powerful forms and more compelling user experiences.
- Date pickers, color pickers, and numeric stepper controls have been added.
- Input field types now include email, search, and URL.
- PUT and DELETE form methods are now supported.

Integrated API (Application Programming Interfaces)

- Drag and Drop
- Audio and Video
- Offline Web Applications
- History
- Local Storage
- Geolocation
- Web Messaging

You will learn more about these new features in the upcoming lessons.
Tap the **Continue** button to begin!

255 COMMENTS



The List of Content Models

In HTML, elements typically belonged in either the block level or inline content model. HTML5 introduces **seven** main content models.

- Metadata
- Embedded
- Interactive
- Heading
- Phrasing
- Flow
- Sectioning

The HTML5 content models are designed to make the markup structure more meaningful for both the browser and the web designer.

509 COMMENTS



Content Models

Metadata: Content that sets up the presentation or behavior of the rest of the content. These elements are found in the **head** of the document.

Elements: `<base>`, `<link>`, `<meta>`, `<noscript>`, `<script>`, `<style>`, `<title>`

Embedded: Content that imports other resources into the document.

Elements: `<audio>`, `<video>`, `<canvas>`, `<iframe>`, ``, `<math>`, `<object>`, `<svg>`

Interactive: Content specifically intended for user interaction.

Elements: `<a>`, `<audio>`, `<video>`, `<button>`, `<details>`, `<embed>`, `<iframe>`, ``, `<input>`, `<label>`, `<object>`, `<select>`, `<textarea>`

Heading: Defines a section header.

Elements: `<h1 >`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`, `<hgroup>`

Phrasing: This model has a number of inline level elements in common with HTML4.

Elements: ``, ``, ``, `<label>`, `
`, `<small>`, `<sub>`, and more.

The same element can belong to more than one content model.

379 COMMENTS

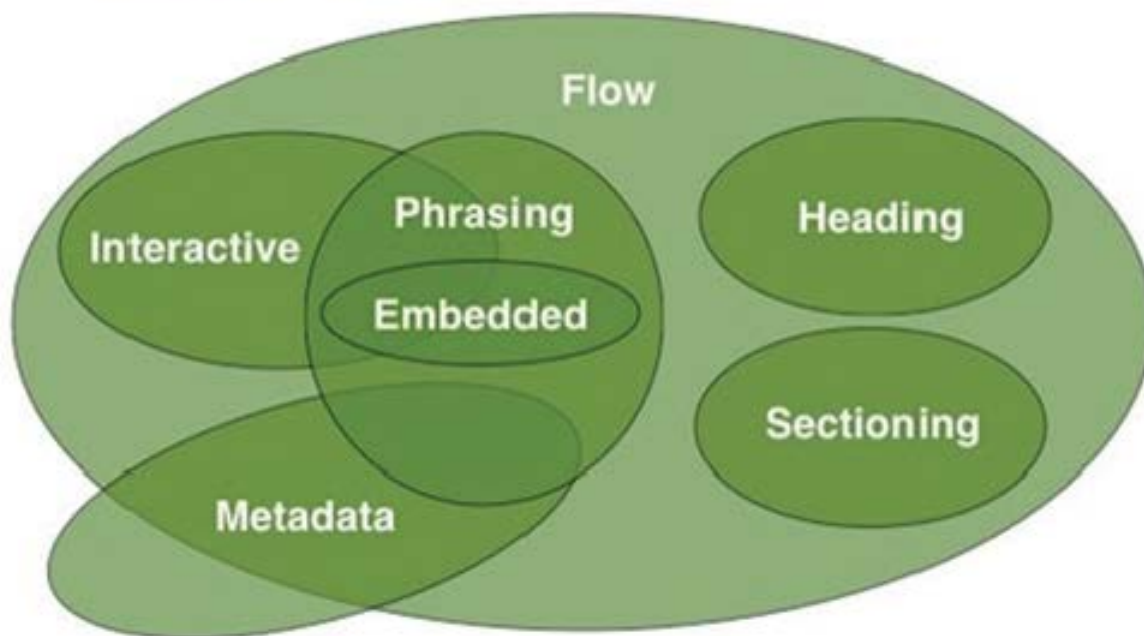


Content Models

Flow content: Contains the majority of HTML5 elements that would be included in the normal flow of the document.

Sectioning content: Defines the scope of headings, content, navigation, and footers.

Elements: `<article>`, `<aside>`, `<nav>`, `<section>`



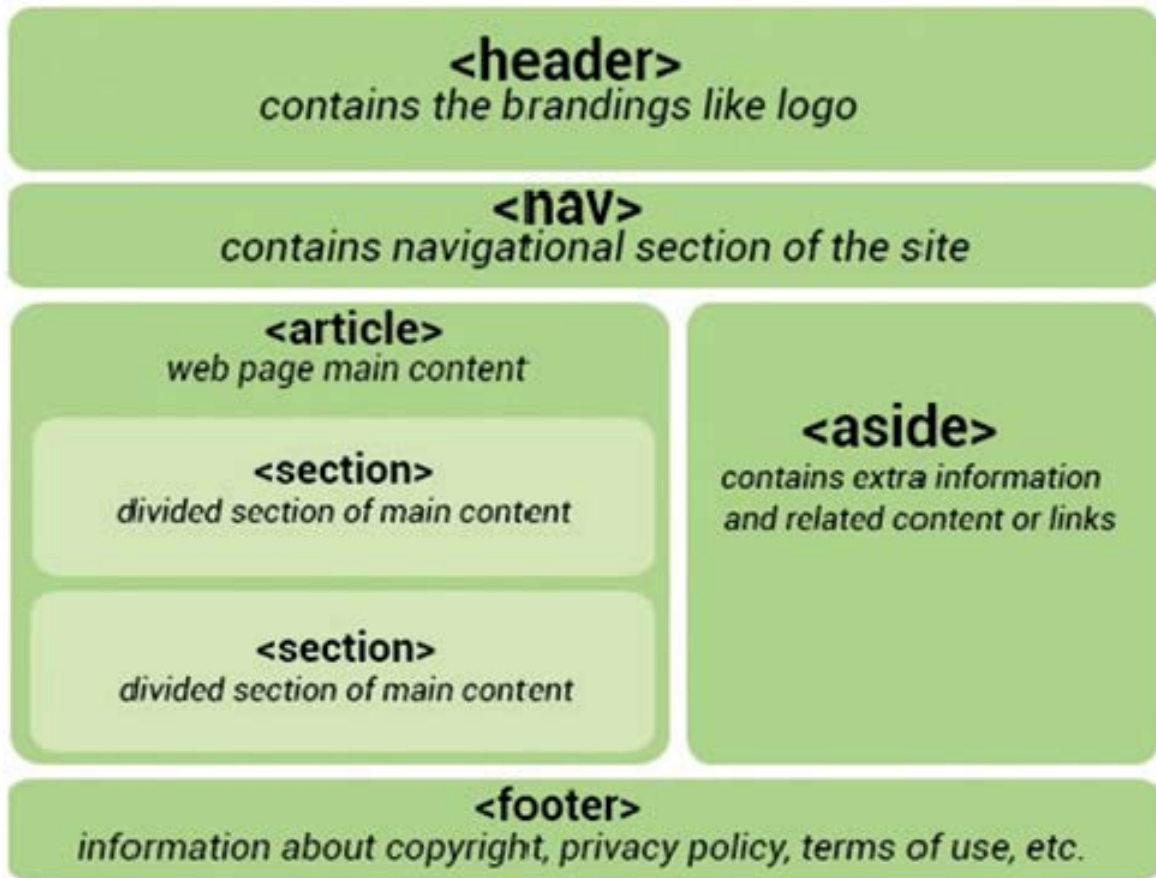
The various content models overlap in certain areas, depending on how they are being used.

252 COMMENTS



Page Structure in HTML5

A generic HTML5 page structure looks like this:



You may not need some of these elements, depending on your page structure.

426 COMMENTS



The <header> Element

In HTML4, we would define a header like this:

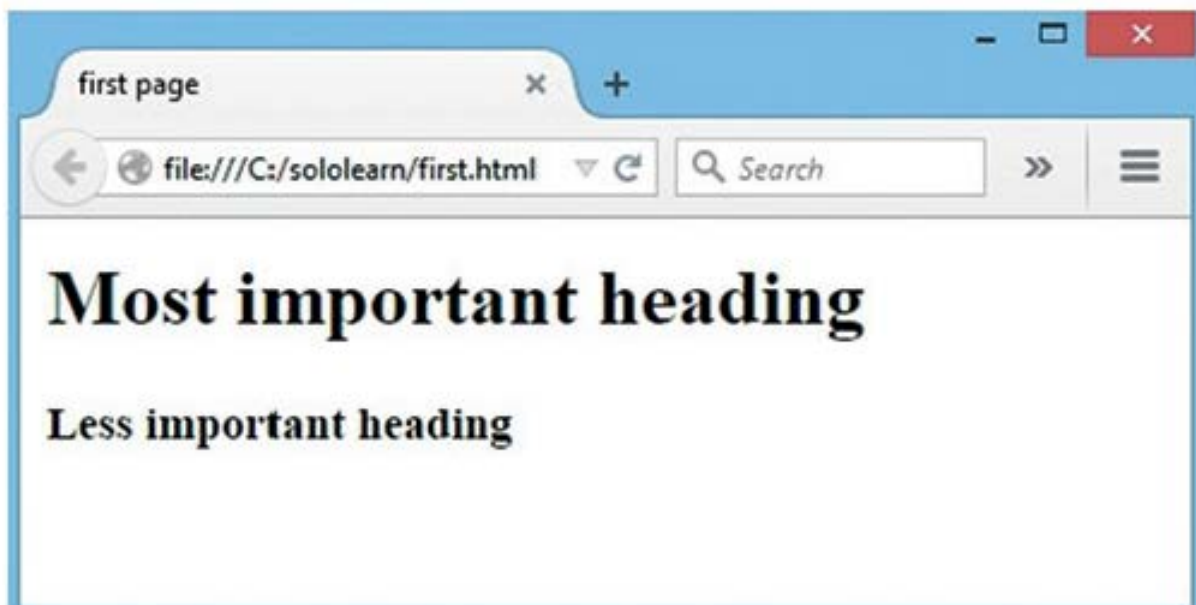
```
<div id="header">
```

In HTML5, a simple <header> tag is used, instead.

The <header> element is appropriate for use inside the body tag.

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <header>
      <h1> Most important heading </h1>
      <h3> Less important heading </h3>
    </header>
  </body>
</html>
```

Try It Yourself



Note that the <header> is completely different from the <head> tag.

594 COMMENTS



The <footer> Element

The footer element is also widely used. Generally we refer to a section located at the very bottom of the web page as the footer.

```
<footer>...</footer>
```

The following information is usually provided between these tags:

- Contact Information
- Privacy Policy
- Social Media Icons
- Terms of Service
- Copyright Information
- Sitemap and Related Documents

215 COMMENTS



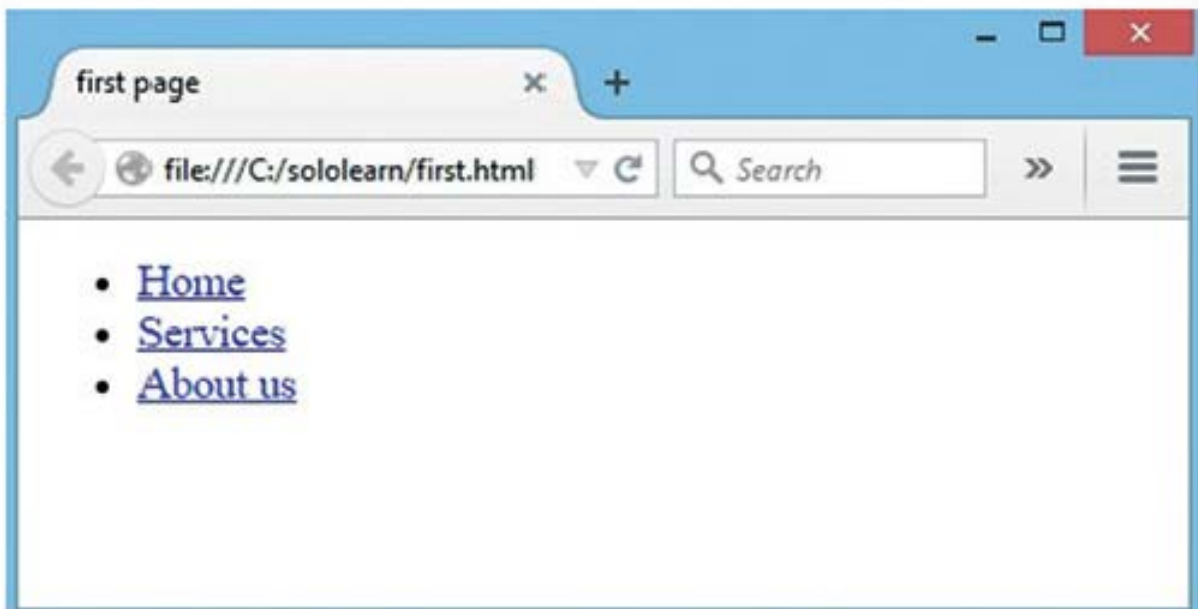
The <nav> Element

This tag represents a section of a page that links to other pages or to certain sections within the page. This would be a section with navigation links.

Here is an example of a major block of navigation links:

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Services</a></li>
    <li><a href="#">About us</a></li>
  </ul>
</nav>
```

Try It Yourself



Not all of the links in a document should be inside a <nav> element. The <nav> element is intended only for major blocks of navigation links. Typically, the <footer> element often has a list of links that don't need to be in a <nav> element.

476 COMMENTS



The <article> Element

Article is a self-contained, independent piece of content that can be used and distributed separately from the rest of the page or site. This could be a forum post, a magazine or newspaper article, a blog entry, a comment, an interactive widget or gadget, or any other independent piece of content.

The <article> element replaces the <div> element that was widely used in HTML4, along with an id or class.

```
<article>
  <h1>The article title</h1>
  <p>Contents of the article element </p>
</article>
```

Try It Yourself

When an <article> element is nested, the inner element represents an article related to the outer element. For example, blog post comments can be <article> elements nested in the <article> representing the blog post.

420 COMMENTS



The <section> Element

<section> is a logical container of the page or article.

Sections can be used to divide up content within an article.

For example, a homepage could have a section for introducing the company, another for news items, and still another for contact information.

Each <section> should be identified, typically by including a heading (h1-h6 element) as a child of the <section> element.

```
<article>
  <h1>Welcome</h1>
  <section>
    <h1>Heading</h1>
    <p>content or image</p>
  </section>
</article>
```

Try It Yourself

If it makes sense to separately syndicate the content of a <section> element, use an <article> element instead.

481 COMMENTS



The <aside> Element

<aside> is secondary or tangential content which could be considered separate from but indirectly related to the main content.

This type of content is often represented in sidebars.

When an <aside> tag is used within an <article> tag, the content of the <aside> should be specifically related to that article.

```
<article>
  <h1> Gifts for everyone </h1>
  <p> This website will be the best place for choosing gifts </p>
  <aside>
    <p> Gifts will be delivered to you within 24 hours </p>
  </aside>
</article>
```

Try It Yourself

When an <aside> tag is used outside of an <article> tag, its content should be related to the surrounding content.

295 COMMENTS



Audio on the Web

Before HTML5, there was no standard for playing audio files on a web page. The HTML5 `<audio>` element specifies a standard for embedding audio in a web page.

There are two different ways to specify the audio source file's URL. The first uses the source attribute:

```
<audio src="audio.mp3" controls>  
  Audio element not supported by your browser  
</audio>
```

Try It Yourself

The second way uses the `<source>` element inside the `<audio>` element:

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

Try It Yourself

Multiple `<source>` elements can be linked to different audio files. The browser will use the first recognized format.

708 COMMENTS



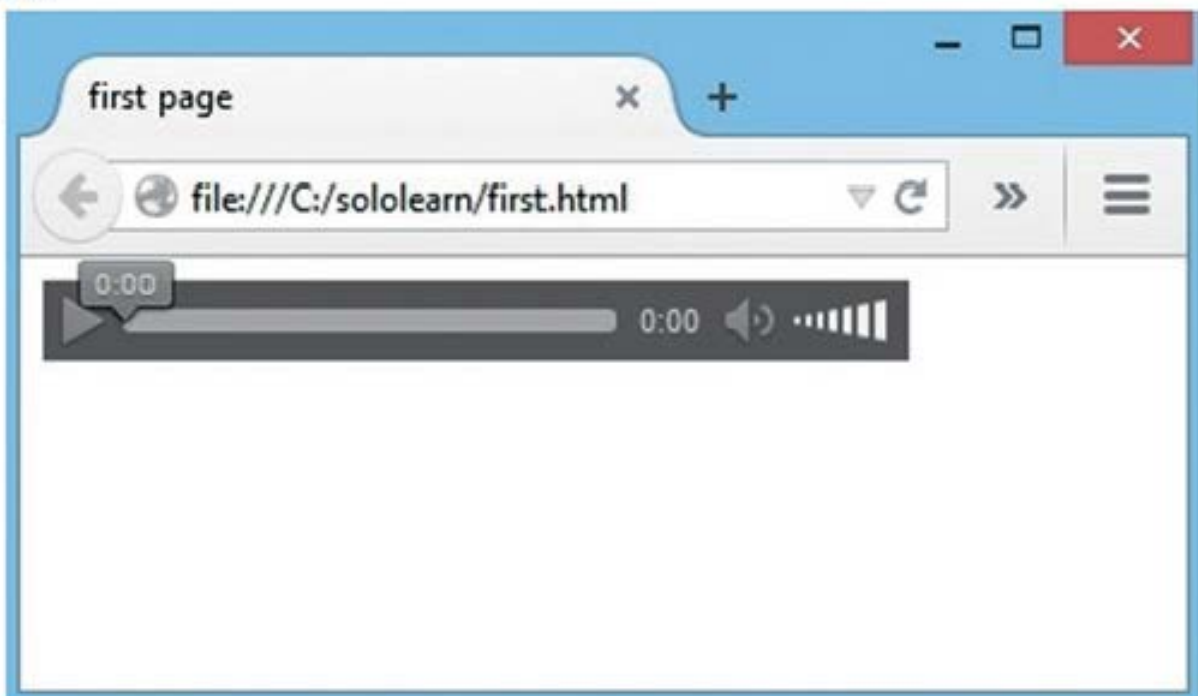
Audio on the Web

The `<audio>` element creates an audio player inside the browser.

```
<audio controls>  
  <source src="audio.mp3" type="audio/mpeg">  
  <source src="audio.ogg" type="audio/ogg">  
  Audio element not supported by your browser.  
</audio>
```

Try It Yourself

Result:



The text between the `<audio>` and `</audio>` tags will display in browsers that do not support the `<audio>` element.

259 COMMENTS



Attributes of <audio>

controls

Specifies that audio controls should be displayed (such as a play/pause button, etc.)

autoplay

When this attribute is defined, audio starts playing as soon as it is ready, without asking for the visitor's permission.

```
<audio controls autoplay>
```

Try It Yourself

loop

This attribute is used to have the audio replay every time it is finished.

```
<audio controls autoplay loop>
```

Try It Yourself

Currently, there are three supported file formats for the <audio> element: MP3, WAV, and OGG.

509 COMMENTS



Videos in HTML

The video element is similar to the audio element.

You can specify the video source URL using an attribute in a video element, or using source elements inside the video element:

```
<video controls>  
  <source src="video.mp4" type="video/mp4">  
  <source src="video.ogg" type="video/ogg">  
  Video is not supported by your browser  
</video>
```

Try It Yourself

Another aspect that the audio and video elements have in common is that the major browsers do not all support the same file types. If the browser does not support the first video type, it will try the next one.

455 COMMENTS



Q&A



Attributes of <video>

Another aspect shared by both the audio and the video elements is that each has **controls**, **autoplay** and **loop** attributes.

In this example, the video will replay after it finishes playing:

```
<video controls autoplay loop>  
  <source src="video.mp4" type="video/mp4">  
  <source src="video.ogg" type="video/ogg">  
  Video is not supported by your browser  
</video>
```

Try It Yourself

Currently, there are three supported video formats for the <video> element: MP4, WebM, and OGG.

307 COMMENTS



Progress Bar

The `<progress>` element provides the ability to create progress bars on the web. The progress element can be used within headings, paragraphs, or anywhere else in the body.

Progress Element Attributes

Value: Specifies how much of the task has been completed.

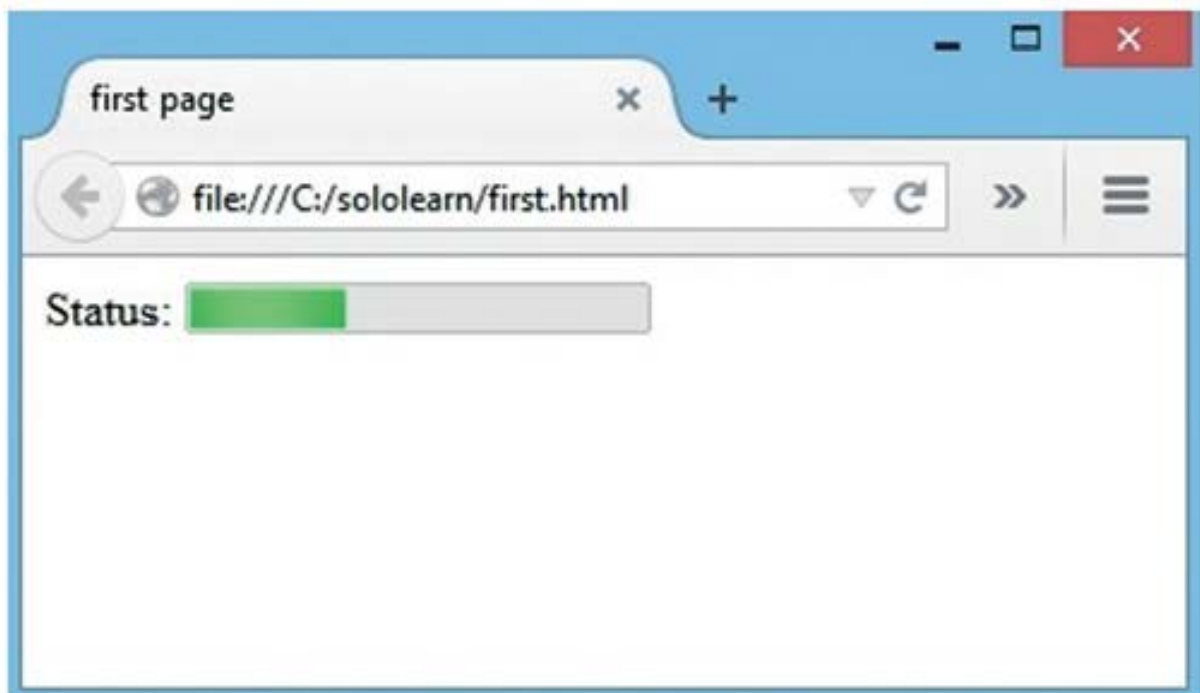
Max: Specifies how much work the task requires in total.

Example:

```
Status: <progress min="0" max="100" value="35">
</progress>
```

Try It Yourself

Result:



Use the `<progress>` tag in conjunction with JavaScript to dynamically display a task's progress.

498 COMMENTS



HTML5 Web Storage

With HTML5 web storage, websites can store data on a user's local computer. Before HTML5, we had to use **JavaScript cookies** to achieve this functionality.

The Advantages of Web Storage

- More secure
- Faster
- Stores a larger amount of data
- Stored data is not sent with every server request

Local storage is per domain. All pages from one domain can store and access the same data.

478 COMMENTS



Types of Web Storage Objects

There are two types of web storage objects:

- `sessionStorage()`
- `localStorage()`

Local vs. Session

- Session Storage is destroyed once the user closes the browser
- Local Storage stores data with no expiration date

You need to be familiar with basic JavaScript in order to understand and use the API.

315 COMMENTS



Working with Values

The syntax for web storage for both local and session storage is very simple and similar. The data is stored as key/value pairs.

Storing a Value:

```
localStorage.setItem("key1", "value1");
```

Getting a Value:

```
//this will print the value  
alert(localStorage.getItem("key1"));
```

Removing a Value:

```
localStorage.removeItem("key1");
```

Removing All Values:

```
localStorage.clear();
```

The same syntax applies to the session storage, with one difference: Instead of `localStorage`, `sessionStorage` is used.

380 COMMENTS



What is the Geolocation API?

In HTML5, the Geolocation API is used to obtain the user's geographical location.

Since this can compromise user privacy, the option is not available unless the user approves it.

Geolocation is much more accurate for devices with GPS, like smartphones and the like.

215 COMMENTS



Using HTML Geolocation

The Geolocation API's main method is `getCurrentPosition`, which retrieves the current geographic location of the user's device.

```
navigator.geolocation.getCurrentPosition();
```

Parameters:

showLocation (mandatory): Defines the callback method that retrieves location information.

ErrorHandler(optional): Defines the callback method that is invoked when an error occurs in processing the asynchronous call.

Options (optional): Defines a set of options for retrieving the location information.

You need to be familiar with basic JavaScript in order to understand and use the API.

253 COMMENTS



Presenting Data

User location can be presented in two ways: **Geodetic** and **Civic**.

1. The geodetic way to describe position refers directly to latitude and longitude.
2. The civic representation of location data is presented in a format that is more easily read and understood by the average person.

Each parameter has both a geodetic and a civic representation:

Attribute	Geodetic	Civic
Position	59.3, 18.6	Stockholm
Elevation	10 meters	4 th floor
Heading	234 degrees	City center
Speed	5km / h	Walking
Orientation	45 degrees	North-East

The `getCurrentPosition()` method returns an object if it is successful. The latitude, longitude, and accuracy properties are always returned.

209 COMMENTS



Making Elements Draggable

The drag and drop feature lets you "grab" an object and drag it to a different location. To make an element draggable, just set the **draggable** attribute to true:

```
<img draggable="true" />
```

Any HTML element can be draggable.

The API for HTML5 drag and drop is event-based.

Example:

```
<!DOCTYPE HTML>
<html>
  <head>
    <script>
      function allowDrop(ev) {
        ev.preventDefault();
      }

      function drag(ev) {
        ev.dataTransfer.setData("text", ev.target.id);
      }

      function drop(ev) {
        ev.preventDefault();
        var data = ev.dataTransfer.getData("text");
        ev.target.appendChild(document.getElementById(data));
      }
    </script>
  </head>
  <body>

    <div id="box" ondrop="drop(event)"
      ondragover="allowDrop(event)"
      style="border:1px solid black;
      width:200px; height:200px"></div>

  </body>
</html>
```

What to Drag

When the element is dragged, the **ondragstart** attribute calls a function, `drag(event)`, which specifies what data is to be dragged.

The `dataTransfer.setData()` method sets the data type and the value of the dragged data:

```
function drag(ev) {
    ev.dataTransfer.setData("text", ev.target.id);
}
```

In our example, the data type is "text" and the value is the ID of the draggable element ("image").

Where to Drop

The **ondragover** event specifies where the dragged data can be dropped. By default, data and elements cannot be dropped in other elements. To allow a drop, we must prevent the default handling of the element.

This is done by calling the `event.preventDefault()` method for the **ondragover** event.

Do the Drop

When the dragged data is dropped, a drop event occurs.

In the example above, the **ondrop** attribute calls a function, `drop(event)`:

```
function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
}
```

The `preventDefault()` method prevents the browser's default handling of the data (default is open as link on drop).

The dragged data can be accessed with the `dataTransfer.getData()` method. This method will return any data that was set to the same type in the `setData()` method.

The dragged data is the ID of the dragged element ("image").

At the end, the dragged element is appended into the drop element, using the `appendChild()` function.

Basic knowledge of JavaScript is required to understand and use the API.

559 COMMENTS



Drawing Shapes

SVG stands for Scalable Vector Graphics, and is used to draw shapes with HTML-style markup.

It offers several methods for drawing paths, boxes, circles, text, and graphic images.

SVG is not pixel-based, so it can be magnified infinitely with no loss of quality.

197 COMMENTS



Inserting SVG Images

An SVG image can be added to HTML code with just a basic image tag that includes a source attribute pointing to the image:

```

```

SVG defines vector-based graphics in XML format.

192 COMMENTS



Drawing a Circle

To draw shapes with SVG, you first need to create an **SVG** element tag with two attributes: width and height.

```
<svg width="1000" height="1000"></svg>
```

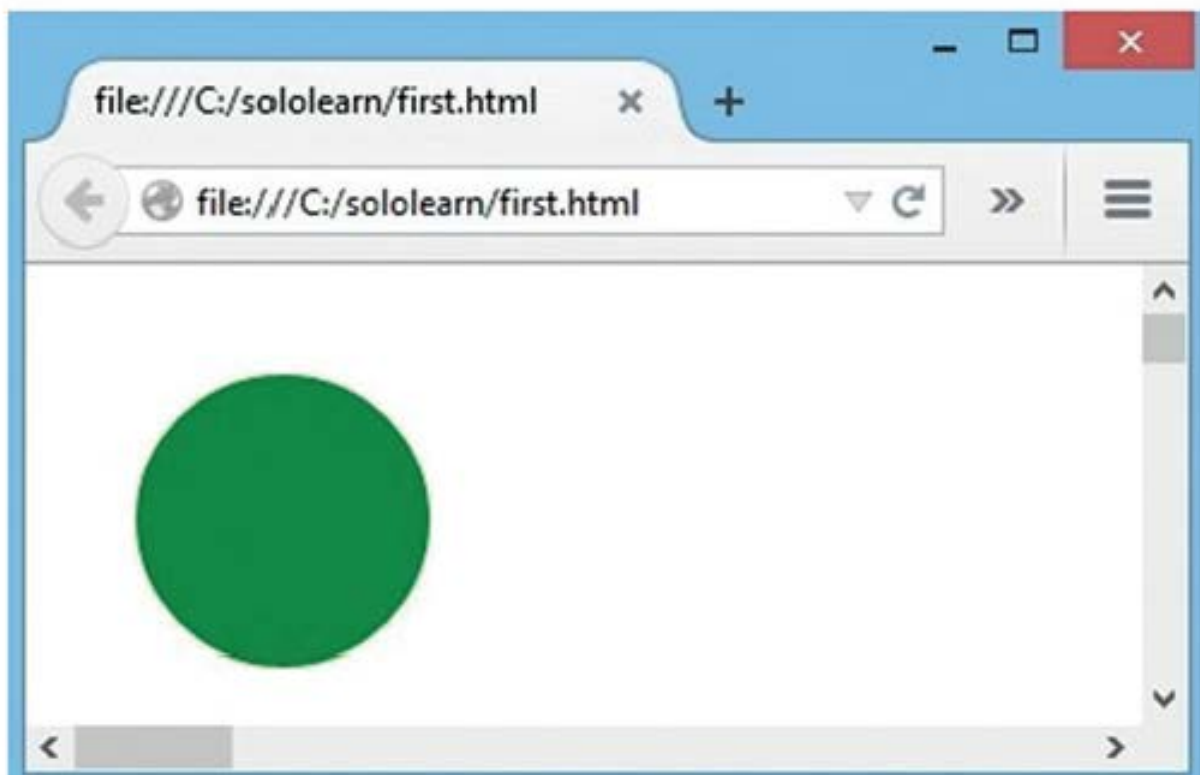
To create a circle, add a `<circle>` tag:

```
<svg width="2000" height="2000">  
  <circle cx="80" cy="80" r="50" fill="green" />  
</svg>
```

Try It Yourself

- **cx** pushes the center of the circle further to the right of the screen
- **cy** pushes the center of the circle further down from the top of the screen
- **r** defines the radius
- **fill** determines the color of our circle
- **stroke** adds an outline to the circle

Result:



Every element and every attribute in SVG files can be animated.

577 COMMENTS

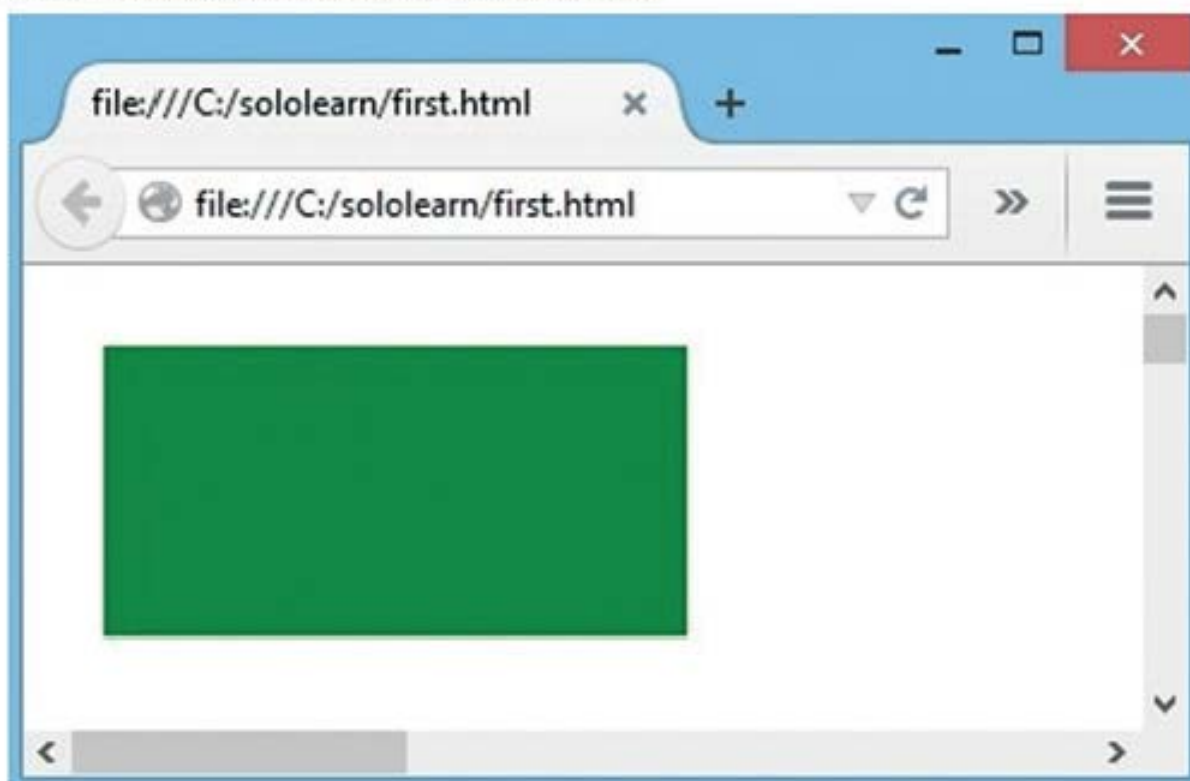
Other Shape Elements

`<rect>` defines a rectangle:

```
<svg width="2000" height="2000">  
  <rect width="300" height="100"  
    x="20" y="20" fill="green" />  
</svg>
```

Try It Yourself

The following code will draw a green-filled rectangle.



`<line>` defines a line segment:

```
<svg width="400" height="410">  
  <line x1="10" y1="10" x2="200" y2="100"  
    style="stroke:#000000; stroke-linecap:round;  
    stroke-width:20" />  
</svg>
```

Try It Yourself

(x1, y1) define the start coordinates(x2, y2) define the end coordinates.

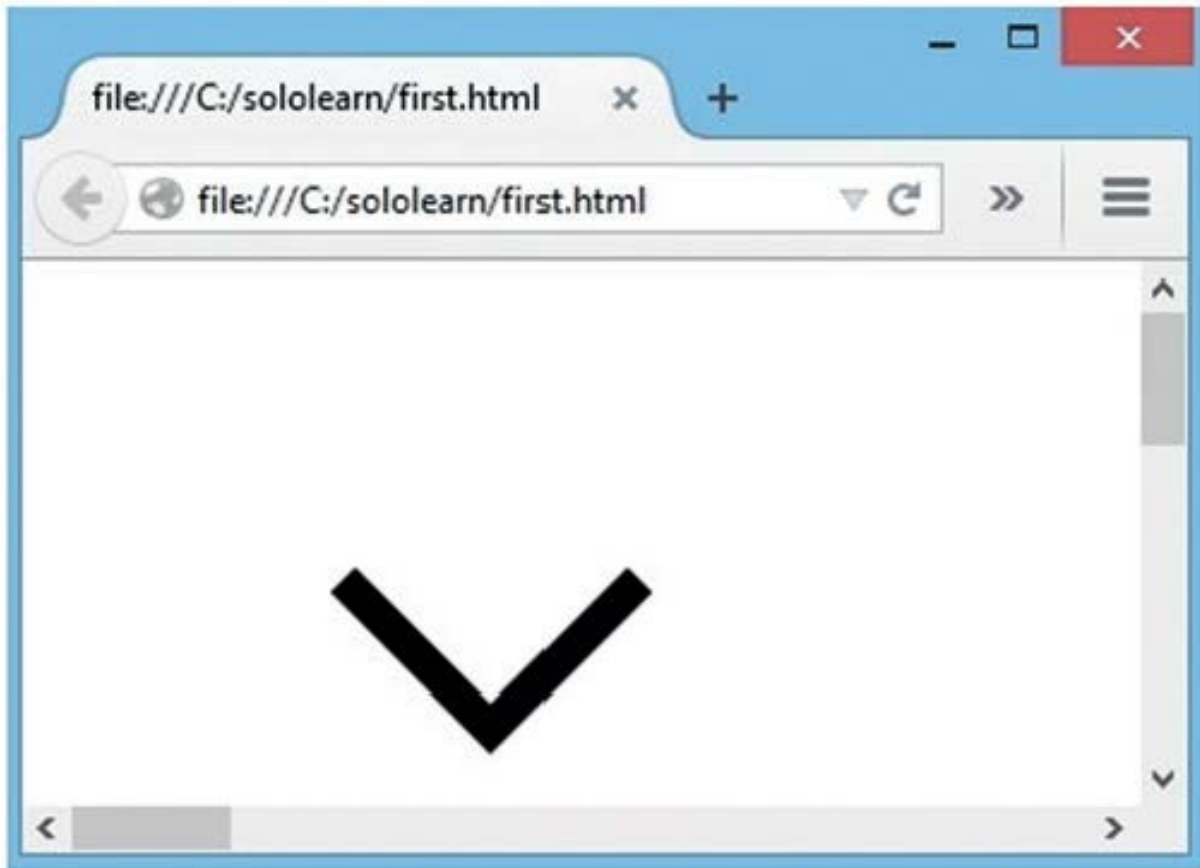


<polyline> defines shapes built from multiple line definitions:

```
<svg width="2000" height="500">  
  <polyline style="stroke-linejoin:miter; stroke:black;  
    stroke-width:12; fill:none;"  
    points="100 100, 150 150, 200 100" />  
</svg>
```

Try It Yourself

Points are the polyline's coordinates.
The code below will draw a black check sign:



The width and height attributes of the `<rect>` element define the height and the width of the rectangle.

533 COMMENTS



<ellipse> and <polygon>

Ellipse

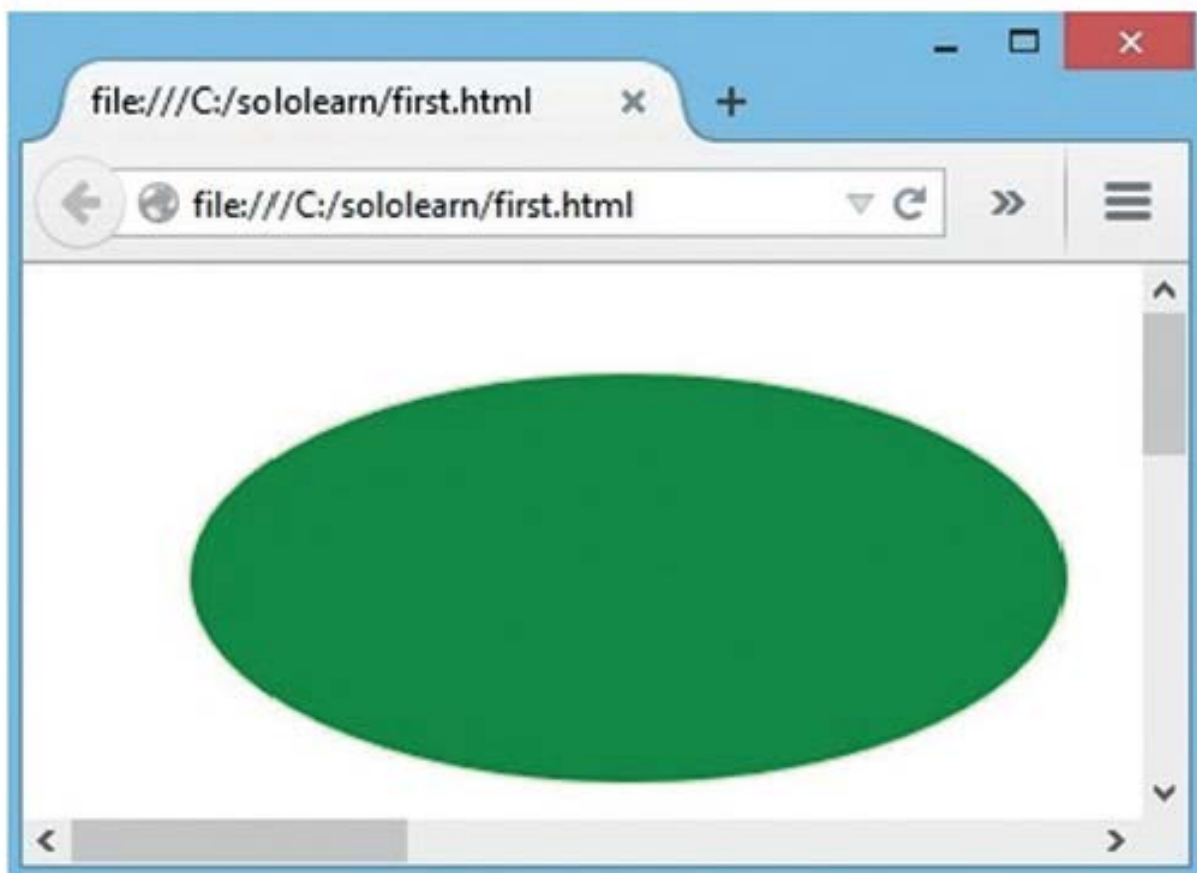
The <ellipse> is similar to the <circle>, with one exception:

You can independently change the horizontal and vertical axes of its radius, using the rx and ry attributes.

```
<svg height="500" width="1000">
  <ellipse cx="200" cy="100" rx="150" ry="70" style="fill:green" />
</svg>
```

Try It Yourself

Result:



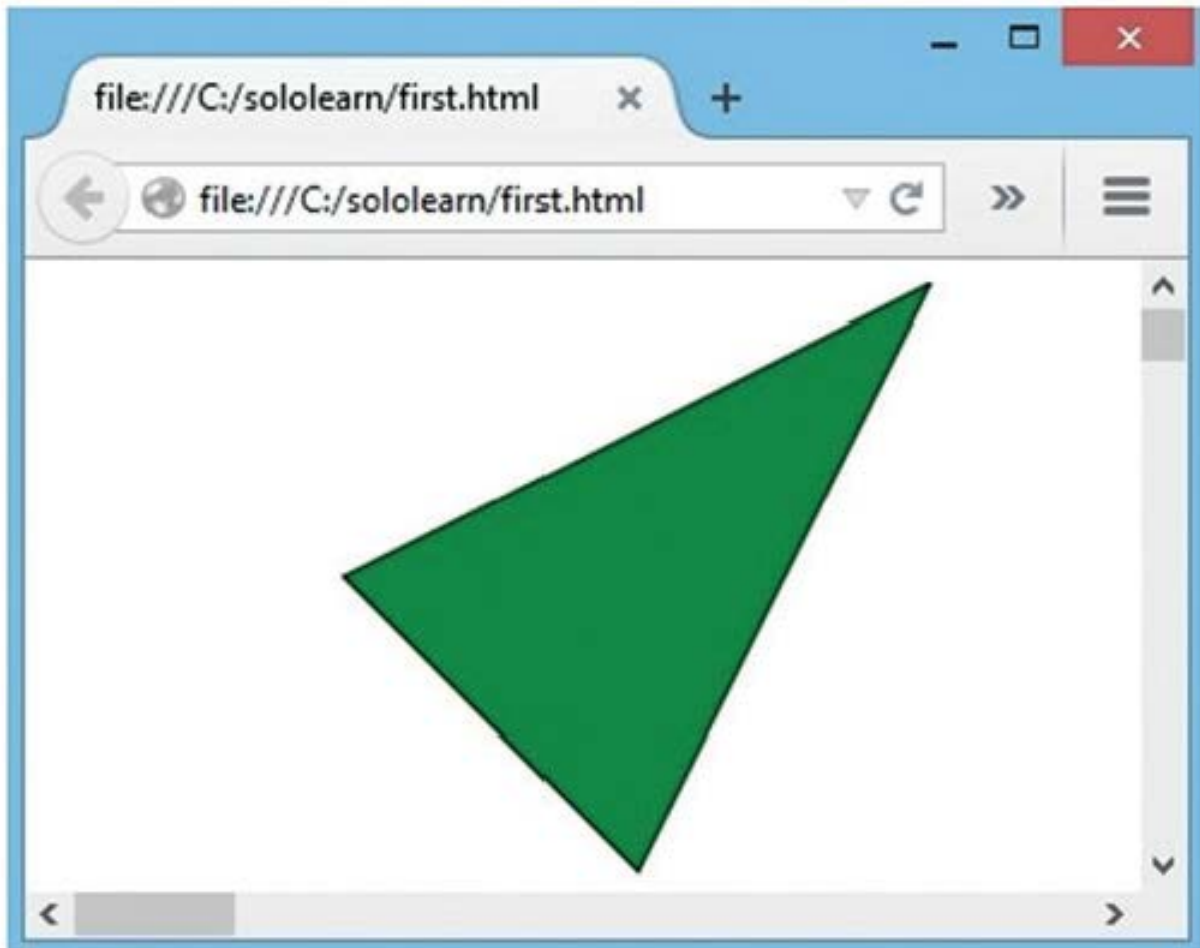
Polygon

The <polygon> element is used to create a graphic with at least three sides. The polygon element is unique because it automatically closes off the shape for you.

```
<svg width="2000" height="2000">
  <polygon points="100 100, 200 200, 300 0"
    style="fill: green; stroke:black;" />
</svg>
```

Try It Yourself

Result:



Polygon comes from Greek. "Poly" means "many" and "gon" means "angle."

332 COMMENTS



Shape Animations

SVG animations can be created using the `<animate>` element.

The example below creates a rectangle that will change its position in 3 seconds and will then repeat the animation twice:

```
<svg width="1000" height="250">
  <rect width="150" height="150" fill="orange">
    <animate attributeName="x" from="0" to="300"
      dur="3s" fill="freeze" repeatCount="2"/>
  </rect>
</svg>
```

Try It Yourself

attributeName: Specifies which attribute will be affected by the animation

from: Specifies the starting value of the attribute

to: Specifies the ending value of the attribute

dur: Specifies how long the animation runs (duration)

fill: Specifies whether or not the attribute's value should return to its initial value when the animation is finished (Values: "remove" resets the value; "freeze" keeps the "to value")

repeatCount: Specifies the repeat count of the animation

In the example above, the rectangle changes its x attribute from 0 to 300 in 3 seconds.

To repeat the animation indefinitely, use the value "indefinite" for the repeatCount attribute.

709 COMMENTS



Paths

The `<path>` element is used to define a path.

The following commands are available for path data:

M: moveto

L: lineto

H: horizontal lineto

V: vertical lineto

C: curveto

S: smooth curveto

Q: quadratic Bézier curve

T: smooth quadratic Bézier curveto

A: elliptical Arc

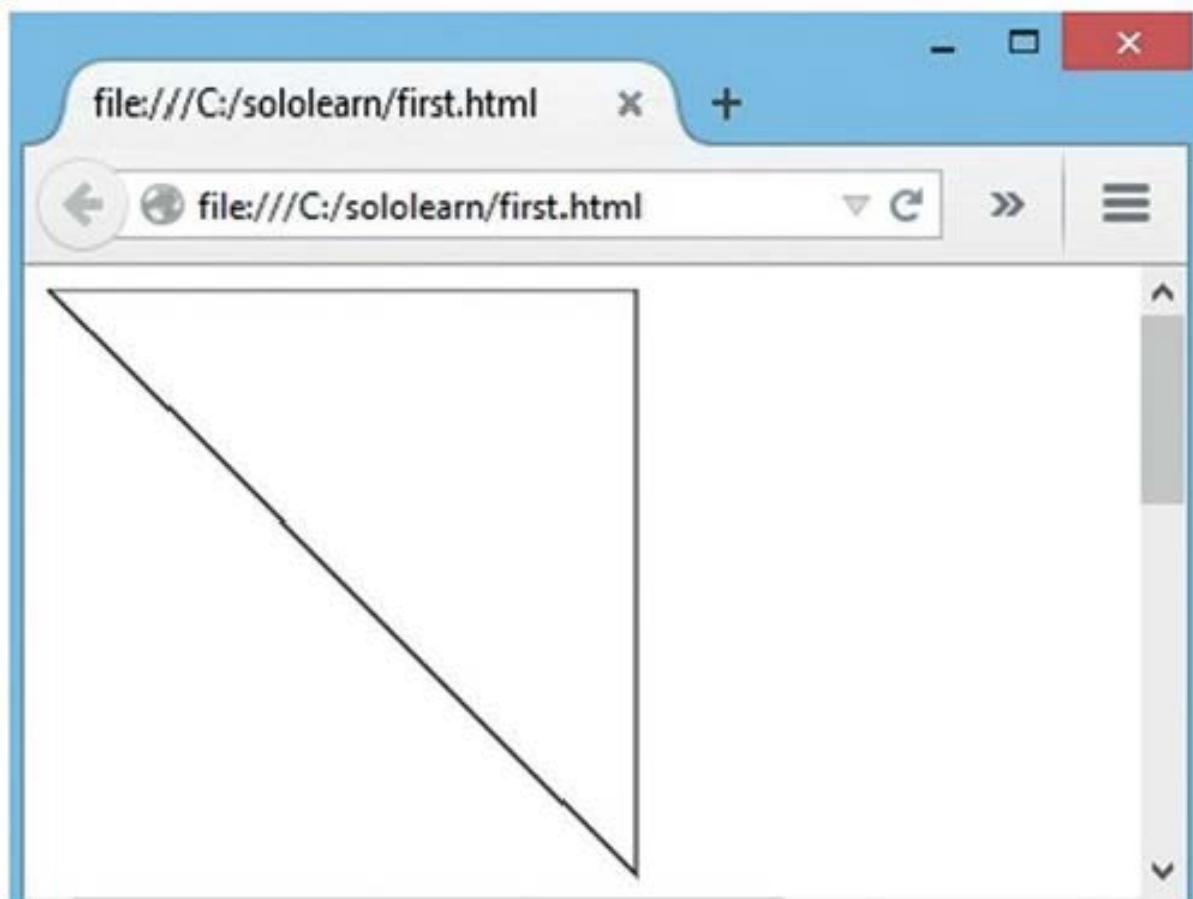
Z: closepath

Define a path using the `d` attribute:

```
<svg width="500" height="500">  
<path d="M 0 0 L200 200 L200 0 Z" style="stroke:#000; fill:none;" />  
</svg>
```

Try It Yourself

M places our "virtual pen" down at the position 0, 0. It then moves 200px down and to the right, then moves up to the position 200, 0. The Z command closes the shape, which results in a hypotenuse:



The <canvas> Element

The HTML canvas is used to draw graphics that include everything from simple lines to complex graphic objects.

The <canvas> element is defined by:

```
<canvas id="canvas1" width="200" height="100">
</canvas>
```

The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics (usually JavaScript).

The <canvas> element must have an **id attribute** so it can be referred to by JavaScript:

```
<html>
  <head></head>
  <body>
    <canvas id="canvas1"
    width="400" height="300"></canvas>

    <script>
      var can = document.getElementById("canvas1");
      var ctx = can.getContext("2d");
    </script>

  </body>
</html>
```

`getContext()` returns a drawing context on the canvas.

Basic knowledge of JavaScript is required to understand and use the Canvas.

250 COMMENTS



Canvas Coordinates

The HTML canvas is a two-dimensional grid.
The upper-left corner of the canvas has the coordinates (0,0).

X coordinate increases to the right.
Y coordinate increases toward the bottom of the canvas.



The `<canvas>` element is only a container for graphics.

96 COMMENTS



Drawing Shapes

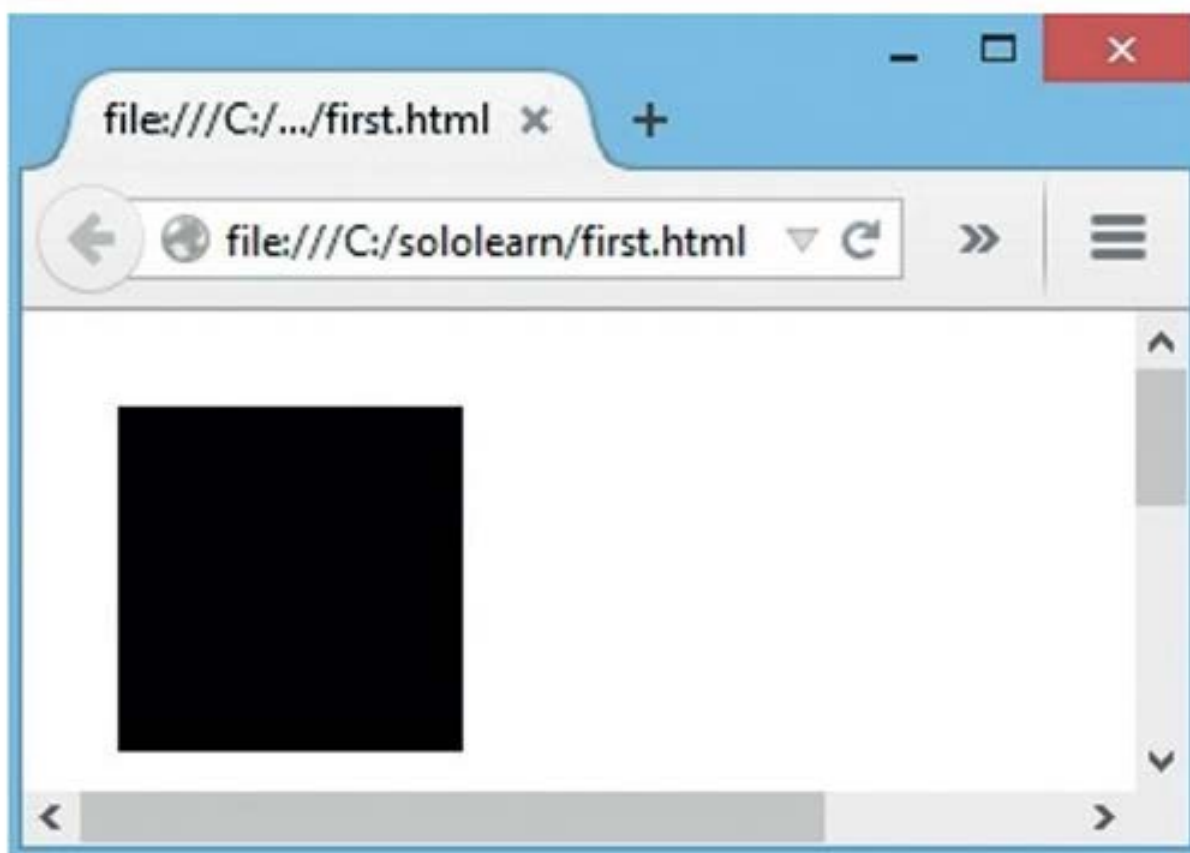
The `fillRect(x, y, w, h)` method draws a "filled" rectangle, in which `w` indicates width and `h` indicates height. The default fill color is black.

A black 100*100 pixel rectangle is drawn on the canvas at the position (20, 20):

```
var c=document.getElementById("canvas1");  
var ctx=c.getContext("2d");  
ctx.fillRect(20,20,100,100);
```

Try It Yourself

Result:

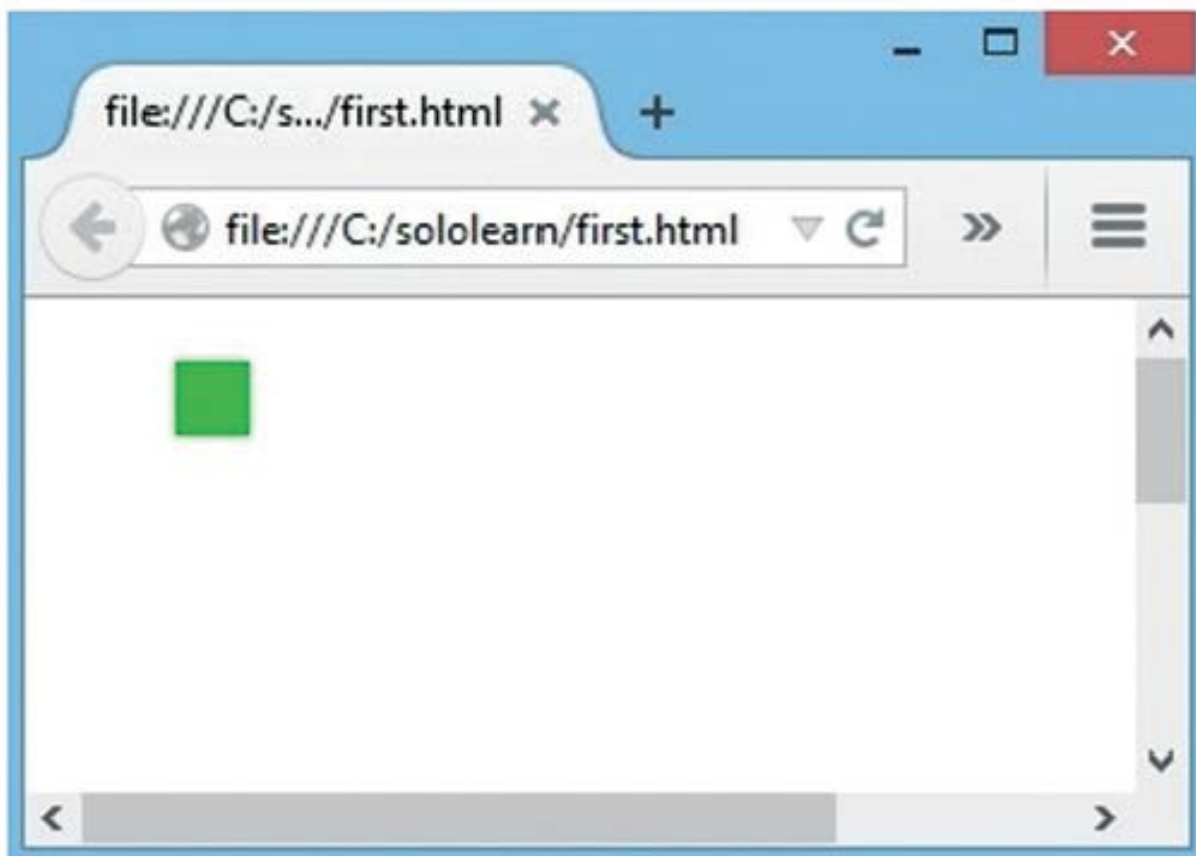


The `fillStyle` property is used to set a color, gradient, or pattern to fill the drawing. Using this property allows you to draw a green-filled rectangle.

```
var canvas=document.getElementById("canvas1");  
var ctx=canvas.getContext("2d");  
ctx.fillStyle="rgba(0,200,0,1)";  
ctx.fillRect(36,10,22,22);
```

Try It Yourself

Result:



The canvas supports various other methods for drawing:

Draw a Line

`moveTo(x,y)`: Defines the starting point of the line.

`lineTo(x,y)`: Defines the ending point of the line.

Draw a Circle

`beginPath()`: Starts the drawing.

`arc(x,y,r,start,stop)`: Sets the parameters of the circle.

`stroke()`: Ends the drawing.

Gradients

`createLinearGradient(x,y,x1,y1)`: Creates a linear gradient.

`createRadialGradient(x,y,r,x1,y1,r1)`: Creates a radial/circular gradient.

Drawing Text on the Canvas

`Font`: Defines the font properties for the text.

`fillText(text,x,y)`: Draws "filled" text on the canvas.

`strokeText(text,x,y)`: Draws text on the canvas (no fill).

There are many other methods aimed at helping to draw shapes and images on the canvas.

299 COMMENTS



Canvas vs. SVG

Canvas

- Elements are drawn programmatically
- Drawing is done with pixels
- Animations are not built in
- High performance for pixels-based drawing operations
- Resolution dependent
- No support for event handlers
- You can save the resulting image as .png or .jpg
- Well suited for graphic-intensive games

SVG

- Elements are part of the page's DOM (Document object model)
- Drawing is done with vectors
- Effects, such as animations are built in
- Based on standard XML syntax, which provides better accessibility
- Resolution independent
- Support for event handlers
- Not suited for game applications
- Best suited for applications with large rendering areas (for example, Google Maps)

You can actually use both SVG and canvas on the same page, if needed. However, you cannot just draw SVG onto a canvas, or vice-versa.

149 COMMENTS



Working with Canvas

The Canvas element can be transformed. As an example, a text is written on the canvas at the coordinates (10, 30).

```
ctx.font="bold 22px Tahoma";  
ctx.textAlign="start";  
ctx.fillText("start", 10, 30);
```

Try It Yourself

Result:



The `translate(x,y)` method is used to move the Canvas. `x` indicates how far to move the grid horizontally, and `y` indicates how far to move the grid vertically.

```
ctx.translate(100, 150);  
ctx.fillText("after translate", 10, 30);
```

Try It Yourself

In this example, the canvas is moved 100px to the right, and 150px down.
Result:



Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

230 COMMENTS



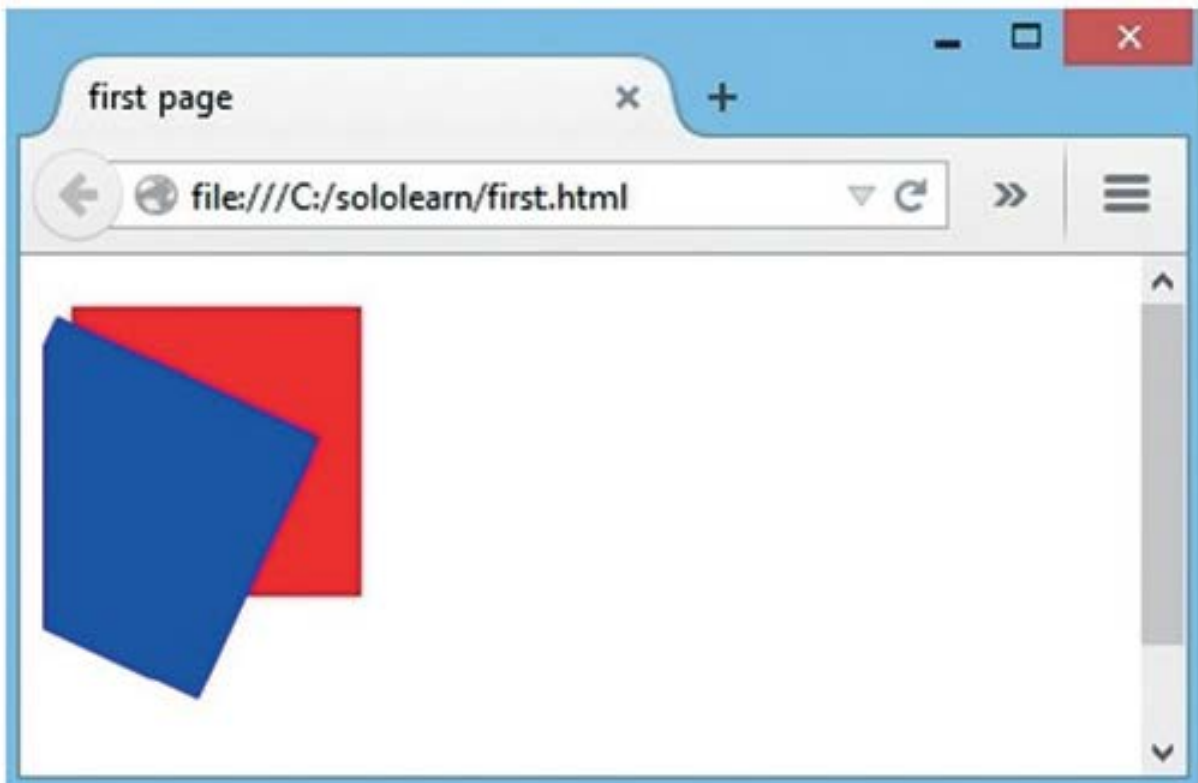
The rotate() Method

The `rotate()` method is used to rotate the HTML5 Canvas. The value must be in radians, not degrees.

Here is an example that draws the same rectangle before and after rotation is set:

```
ctx.fillStyle = "#FF0000";  
ctx.fillRect(10,10, 100, 100);  
  
ctx.rotate( (Math.PI / 180) * 25); //rotate 25 degrees.  
  
ctx.fillStyle = "#0000FF";  
ctx.fillRect(10,10, 100, 100);
```

Try It Yourself



The rotation will only affect drawings made after the rotation is done.

149 COMMENTS



The scale() Method

The `scale()` method scales the current drawing. It takes two parameters:

- The number of times by which the image should be scaled in the X-direction.
- The number of times by which the image should be scaled in the Y-direction.

```
var canvas = document.getElementById('canvas1');
ctx = canvas.getContext('2d');
ctx.font="bold 22px Tahoma";
ctx.textAlign="start";
ctx.fillText("start", 10, 30);
ctx.translate(100, 150);
ctx.fillText("after translate", 0, 0);
ctx.rotate(1);
ctx.fillText("after rotate", 0, 0);
ctx.scale(1.5, 4);
ctx.fillText("after scale", 0,20);
```

Try It Yourself

This will scale the canvas 1.5 times in the X-direction, and 4 times in Y-direction:



This will scale the canvas 1.5 times in the X-direction, and 4 times in Y-direction:



If you scale a drawing, all future drawings will also be scaled.

173 COMMENTS



HTML5 Forms

HTML5 brings many features and improvements to web form creation. There are new attributes and input types that were introduced to help create better experiences for web users.

Form creation is done in HTML5 the same way as it was in HTML4:

```
<form>
  <label>Your name:</label>
  <input id="user" name="username" type="text" />
</form>
```

Try It Yourself

Use the novalidate attribute to avoid form validation on submissions.

259 COMMENTS



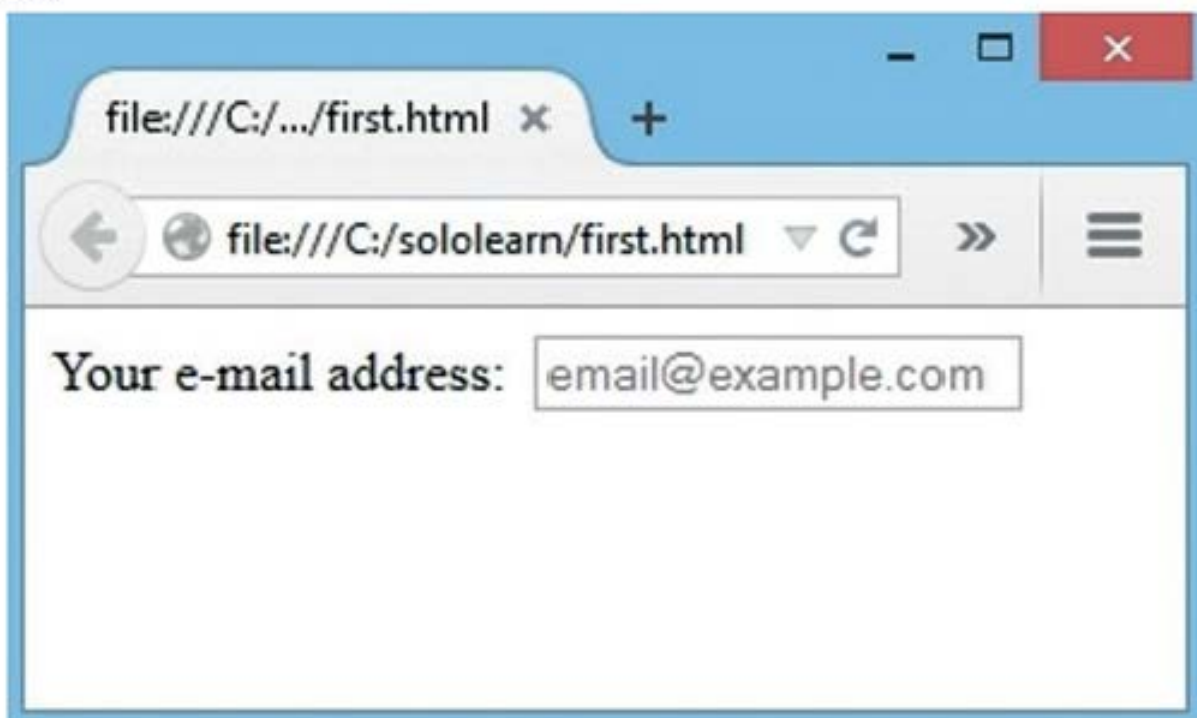
New Attributes

HTML5 has introduced a new attribute called **placeholder**. On `<input>` and `<textarea>` elements, this attribute provides a hint to the user of what information can be entered into the field.

```
<form>
  <label for="email">Your e-mail address: </label>
  <input type="text" name="email" placeholder="email@example.com" />
</form>
```

Try It Yourself

Result:

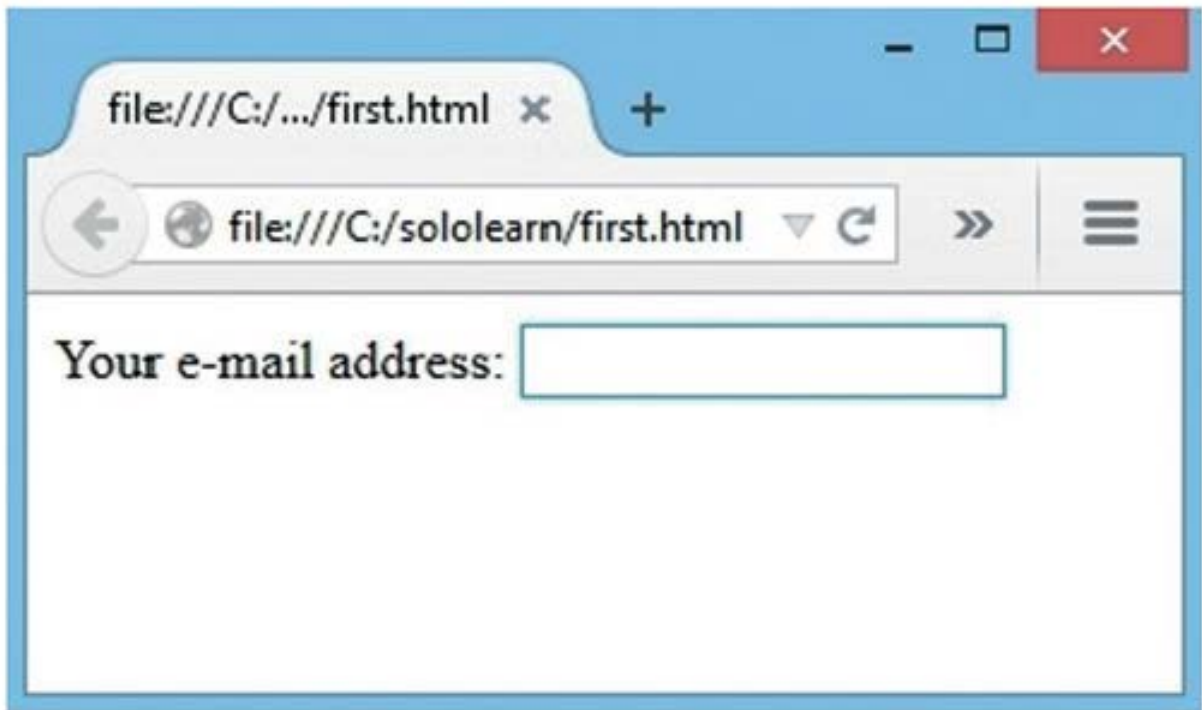


The **autofocus** attribute makes the desired input focus when the form loads:

```
<form>
  <label for="email">Your e-mail address: </label>
  <input type="text" name="email" autofocus/>
</form>
```

Try It Yourself

Result:



The required attribute tells the browser that the input is required.

316 COMMENTS



Forms with Required Fields

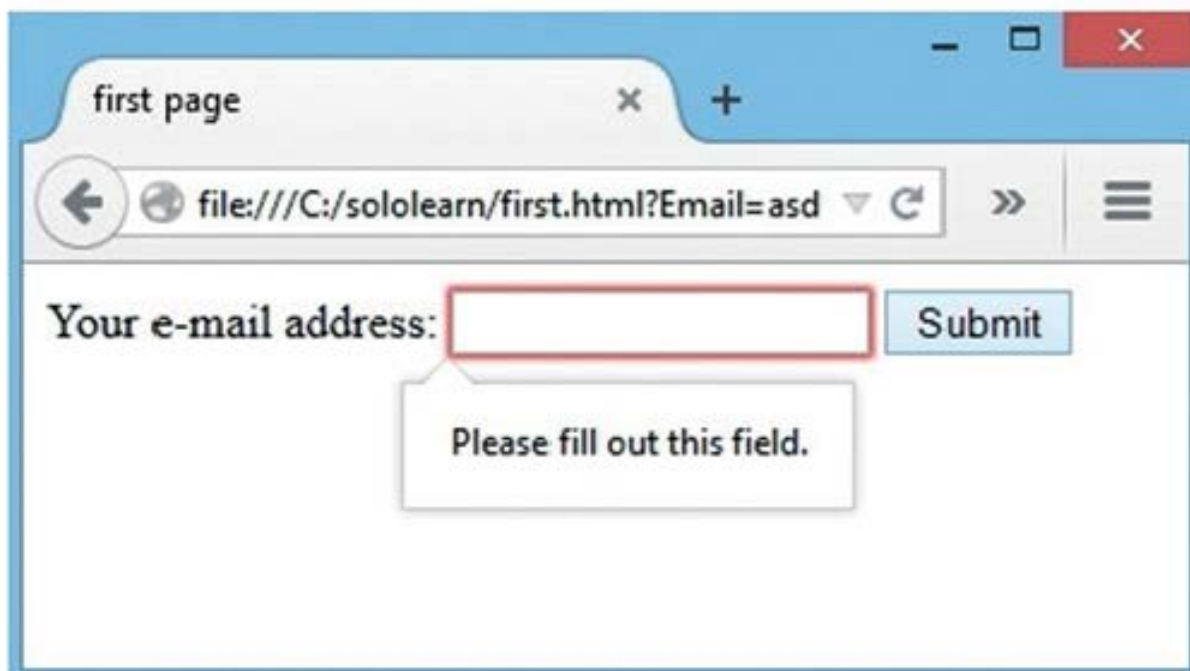
The "required" attribute is used to make the input elements required.

```
<form autocomplete="off">  
  <label for="e-mail">Your e-mail address: </label>  
  <input name="Email" type="text" required />  
  <input type="submit" value="Submit" />  
</form>
```

Try It Yourself

The form will not be submitted without filling in the required fields.

Result:



The **autocomplete** attribute specifies whether a form or input field should have autocomplete turned on or off. When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

HTML5 added several new input types:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

New input attributes in HTML5:

- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

Input types that are not supported by old web browsers, will behave as input type text.

319 COMMENTS



Creating a Search Box

The new `search` input type can be used to create a search box:

```
<input id="mysearch" name="searchitem" type="search" />
```

Try It Yourself

Result:

You must remember to set a name for your input; otherwise, nothing will be submitted.

322 COMMENTS



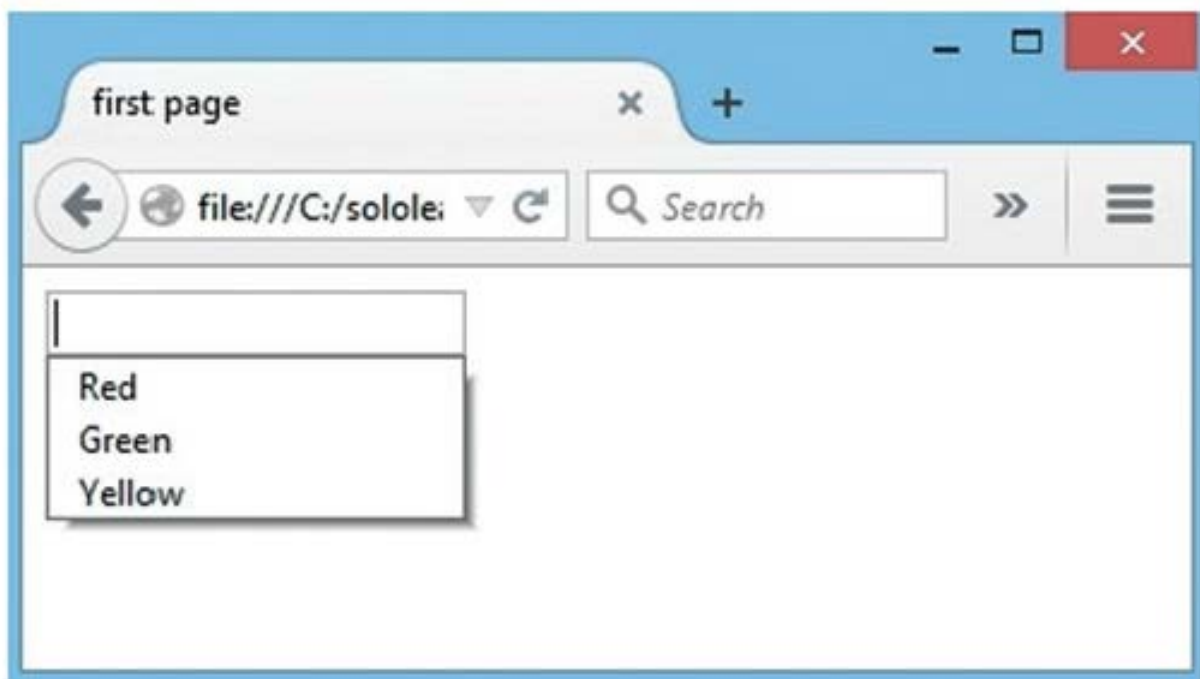
Search Options

The `<datalist>` tag can be used to define a list of pre-defined options for the search field:

```
<input id="car" type="text" list="colors" />
<datalist id="colors">
  <option value="Red">
  <option value="Green">
  <option value="Yellow">
</datalist>
```

Try It Yourself

Result:



`<option>` defines the options in a drop-down list for the user to select.

The ID of the datalist element must match with the list attribute of the input box.

248 COMMENTS



Creating More Fields

Some other new input types include **email**, **url**, and **tel**:

```
<input id="email" name="email" type="email" placeholder="example@example.com" />  
<br />  
<input id="url" name="url" type="url" placeholder="example.com" />  
<br />  
<input id="tel" name="tel" type="tel" placeholder="555.555.1211" />
```

Try It Yourself

These are especially useful when opening a page on a modern mobile device, which recognizes the input types and opens a corresponding keyboard matching the field's type:



These new types make it easier to structure and validate HTML forms.

273 COMMENTS

