

## HTML Introduction

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

## A Simple HTML Document

### Example

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

<....> = Open tag  
</.....>=Close Tag  
<>= Tag

### Example Explained

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

## HTML Element

An HTML element is defined by a start tag, some content, and an end tag:

<tagname>Content goes here...</tagname>

The HTML **element** is everything from the start tag to the end tag:

<h1>My First Heading</h1>

<p>My first paragraph.</p>



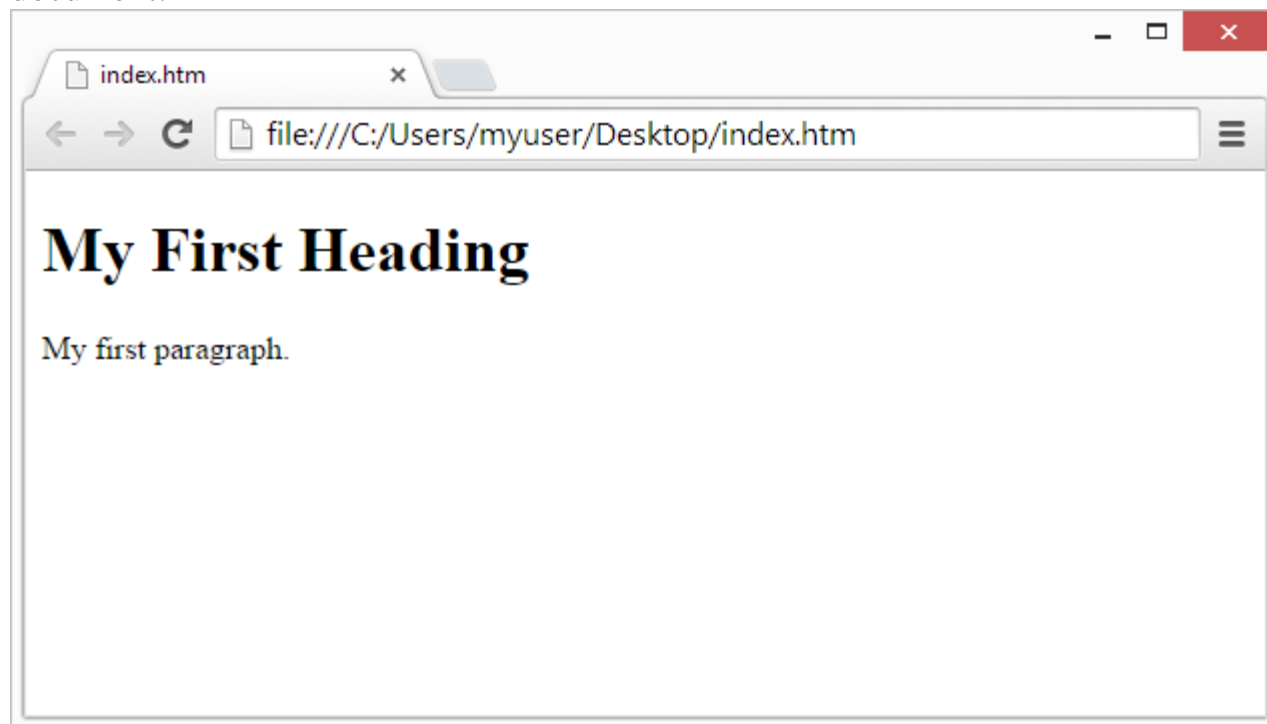
Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
 	<i>None</i>	<i>none</i>

**Note:** Some HTML elements have no content (like the <br> element). These elements are called empty elements. Empty elements do not have an end tag!

## Web Browsers

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags, but uses them to determine how to display the document:



## HTML Page Structure

Below is a visualization of an HTML page structure:

```
<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

**Note:** The content inside the <body> section (the white area above) will be displayed in a browser. The content inside the <title> element will be shown in the browser's title bar or in the page's tab.

## **HTML Editors**

A simple text editor is all you need to learn HTML.

Learn HTML Using Notepad or TextEdit

Web pages can be created and modified by using professional HTML editors.

However, for learning HTML we recommend a simple text editor like Notepad (PC), Notepad++ or TextEdit (Mac).

We believe in that using a simple text editor is a good way to learn HTML.

Follow the steps below to create your first web page with Notepad or TextEdit.

### ***Step 1: Open Notepad (PC)***

#### **Windows 8 or later:**

Open the **Start Screen** (the window symbol at the bottom left on your screen).  
Type **Notepad**.

#### **Windows 7 or earlier:**

Open **Start > Programs > Accessories > Notepad**

### ***Step 2: Write Some HTML***

Write or copy the following HTML code into Notepad:

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

### ***Step 3: Save the HTML Page***

Save the file on your computer. Select **File > Save as** in the Notepad menu with “.html” extension name.

### ***Step 4: View the HTML Page in Your Browser***

Open the saved HTML file in your favorite browser (double click on the file, or right-click - and choose "Open with").

## **HTML Documents**

All HTML documents must start with a document type declaration: <!DOCTYPE html>.

The HTML document itself begins with <html> and ends with </html>.

The visible part of the HTML document is between <body> and </body>.

## **Example**

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The <!DOCTYPE> declaration is not case sensitive.

The <!DOCTYPE> declaration for HTML5 is:

```
<!DOCTYPE html>
```

## **HTML Headings**

HTML headings are titles or subtitles that you want to display on a webpage.

HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

Example

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

**Note:** Browsers automatically add some white space (a margin) before and after a heading.

## **Bigger Headings**

Each HTML heading has a default size. However, you can specify the size for any heading with the style attribute, using the CSS font-size property:

Example

```
<h1 style="font-size:60px;">Heading 1</h1>
```

## **HTML Paragraphs**

The HTML <p> element defines a paragraph.

A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.

Example

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

Example

```
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>
```

```
<p>
This paragraph
contains      a lot of spaces
in the source      code,
but the      browser
ignores it.
</p>
```

## **HTML Horizontal Rules**

The `<hr>` tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The `<hr>` element is used to separate content (or define a change) in an HTML page:  
The `<hr>` tag is an empty tag, which means that it has no end tag.

Example

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```

## **HTML Line Breaks**

The HTML `<br>` element defines a line break.

Use `<br>` if you want a line break (a new line) without starting a new paragraph:

Example

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

## HTML Styles

The HTML style attribute is used to add styles to an element, such as color, font, size, and more.

### **Example**

I am Red

I am Blue

I am Big

### The HTML Style Attribute

Setting the style of an HTML element, can be done with the style attribute.

The HTML style attribute has the following syntax:

```
<tagname style="property:value;">
```

The *property* is a CSS property. The *value* is a CSS value.

### Background Color

The CSS background-color property defines the background color for an HTML element.

Example

Set the background color for a page to powderblue:

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

```
<h1>This is a heading</h1>
```

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

```
</body>
```

### Example

Set background color for two different elements:

```
<body>
```

```
<h1 style="background-color:powderblue;">This is a heading</h1>
```

```
<p style="background-color:tomato;">This is a paragraph.</p>
```

```
</body>
```

### Text Color

The CSS color property defines the text color for an HTML element:

Example

```
<h1 style="color:blue;">This is a heading</h1>
```

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

### Fonts

The CSS font-family property defines the font to be used for an HTML element:

Example

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

## Text Size

The CSS font-size property defines the text size for an HTML element:

### Example

```
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
```

## Text Alignment

The CSS text-align property defines the horizontal text alignment for an HTML element:

### Example

```
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```

## HTML Text Formatting

HTML contains several elements for defining text with a special meaning.

### Example

**This text is bold**

*This text is italic*

This is <sub>subscript</sub> and <sup>superscript</sup>

## HTML Formatting Elements

Formatting elements were designed to display special types of text:

- `<b>` - Bold text
- `<strong>` - Important text
- `<i>` - Italic text
- `<em>` - Emphasized text
- `<mark>` - Marked text
- `<small>` - Smaller text
- `<del>` - Deleted text
- `<ins>` - Inserted text
- `<sub>` - Subscript text
- `<sup>` - Superscript text

## HTML `<b>` and `<strong>` Elements

The HTML `<b>` element defines bold text, without any extra importance.

### Example

```
<b>This text is bold</b>
```

The HTML `<strong>` element defines text with strong importance. The content inside is typically displayed in bold.

### Example



**This text is important!**

### **HTML <i> and <em> Elements**

The HTML <i> element defines a part of text in an alternate voice or mood. The content inside is typically displayed in italic.

**Tip:** The <i> tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

#### **Example**

<i>This text is italic</i>

The HTML <em> element defines emphasized text. The content inside is typically displayed in italic.

**Tip:** A screen reader will pronounce the words in <em> with an emphasis, using verbal stress.

#### **Example**

<em>This text is emphasized</em>

### **HTML <small> Element**

The HTML <small> element defines smaller text:

#### **Example**

<small>This is some smaller text.</small>

### **HTML <mark> Element**

The HTML <mark> element defines text that should be marked or highlighted:

#### **Example**

<p>Do not forget to buy <mark>milk</mark> today.</p>

### **HTML <del> Element**

The HTML <del> element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text:

#### **Example**

<p>My favorite color is <del>blue</del> red.</p>

### **HTML <ins> Element**

The HTML <ins> element defines a text that has been inserted into a document. Browsers will usually underline inserted text:

#### **Example**

<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>

## **HTML <sub> Element**

The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H<sub>2</sub>O:

### **Example**

```
<p>This is <sub>subscripted</sub> text.</p>
```

## **HTML <sup> Element**

The HTML <sup> element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW<sup>[1]</sup>:

### **Example**

```
<p>This is <sup>superscripted</sup> text.</p>
```

## **HTML Comment Tags**

You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here -->
```

Notice that there is an exclamation point (!) in the start tag, but not in the end tag.

**Note:** Comments are not displayed by the browser, but they can help document your HTML source code.

With comments you can place notifications and reminders in your HTML code:

Example

```
<!-- This is a comment -->
```

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

## **HTML Colors**

HTML colors are specified with predefined color names, or with RGB, HEX, HSL, RGBA, or HSLA values.

### **Color Names**

In HTML, a color can be specified by using a color name:

Tomato

Orange

DodgerBlue

MediumSeaGreen

Gray

SlateBlue

Violet

LightGray

HTML supports 140 standard color names

### **Background Color**

We can set the background color for HTML elements:

#### **Example**

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>  
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

### **Text Color**

We can set the color of text:

#### **Example**

```
<h1 style="color:Tomato;">Hello World</h1>  
<p style="color:DodgerBlue;">Lorem ipsum...</p>  
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

### **Border Color**

You can set the color of borders:

#### **Example**

```
<h1 style="border:2px solid Tomato;">Hello World</h1>  
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>  
<h1 style="border:2px solid Violet;">Hello World</h1>
```

## Color Values

In HTML, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values.

The following three <div> elements have their background color set with RGB, HEX, and HSL values:

**rgb(255, 99, 71)**

**#ff6347**

**hsl(9, 100%, 64%)**

The following two <div> elements have their background color set with RGBA and HSLA values, which adds an Alpha channel to the color (here we have 50% transparency):

**rgba(255, 99, 71, 0.5)**

**hsla(9, 100%, 64%, 0.5)**

## Example

<h1 style="background-color:rgb(255, 99, 71);">...</h1>

<h1 style="background-color:#ff6347;">...</h1>

<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>

## HTML Styles - CSS

CSS stands for Cascading Style Sheets.

CSS saves a lot of work. It can control the layout of multiple web pages all at once.

Cascading Style Sheets (CSS) is used to format the layout of a webpage.

With CSS, you can control the color, font, and the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more.

**Tip:** The word **cascading** means that a style applied to a parent element will also apply to all children elements within the parent. So, if you set the color of the body text to "blue", all headings, paragraphs, and other text elements within the body will also get the same color (unless you specify something else)!

## Using CSS

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the style attribute inside HTML elements
- **Internal** - by using a <style> element in the <head> section
- **External** - by using a <link> element to link to an external CSS file

The most common way to add CSS, is to keep the styles in external CSS files. However, in this tutorial we will use inline and internal styles, because this is easier to demonstrate, and easier for you to try it yourself.

## **Inline CSS**

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the style attribute of an HTML element.

The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

Example

```
<h1 style="color:blue;">A Blue Heading</h1>
<p style="color:red;">A red paragraph.</p>
```

## **Internal CSS**

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the <head> section of an HTML page, within a <style> element.

The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to red. In addition, the page will be displayed with a "powderblue" background color:

### **Example**

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1  {color: blue;}
p   {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

## **External CSS**

An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the <head> section of each HTML page:

### **Example**

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
```

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

Here is what the "styles.css" file looks like:

"styles.css":

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

**Tip:** With an external style sheet, you can change the look of an entire web site, by changing one file!

### **CSS Colors, Fonts and Sizes**

The CSS color property defines the text color to be used.

The CSS font-family property defines the font to be used.

The CSS font-size property defines the text size to be used.

### **Example**

Use of CSS color, font-family and font-size properties:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
```

```
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

## **CSS Border**

The CSS border property defines a border around an HTML element.

**Tip:** You can define a border for nearly all HTML elements.

### **Example**

Use of CSS border property:

```
p {
  border: 2px solid powderblue;
}
```

## **CSS Padding**

The CSS padding property defines a padding (space) between the text and the border.

Example

Use of CSS border and padding properties:

```
p {
  border: 2px solid powderblue;
  padding: 30px;
}
```

## **CSS Margin**

The CSS margin property defines a margin (space) outside the border.

### **Example**

Use of CSS border and margin properties:

```
p {
  border: 2px solid powderblue;
  margin: 50px;
}
```

## **HTML Links**

Links are found in nearly all web pages. Links allow users to click their way from page to page.

HTML Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

**Note:** A link does not have to be text. A link can be an image or any other HTML element!

## **HTML Links - Syntax**

The HTML <a> tag defines a hyperlink. It has the following syntax:

```
<a href="url">link text</a>
```

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

The *link text* is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

### **Example**

This example shows how to create a link to W3Schools.com:

```
<a href="https://www.google.com">Visit Google</a>
```

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

**Tip:** Links can of course be styled with CSS, to get another look!

## **HTML Links - The target Attribute**

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

- `_self` - Default. Opens the document in the same window/tab as it was clicked
- `_blank` - Opens the document in a new window or tab
- `_parent` - Opens the document in the parent frame
- `_top` - Opens the document in the full body of the window

### **Example**

Use `target="_blank"` to open the linked document in a new browser window or tab:

```
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

## **HTML Links - Use an Image as a Link**

To use an image as a link, just put the <img> tag inside the <a> tag:

Example

```
<a href="default.asp">
```

```

```

```
</a>
```



## **HTML Links - Different Colors**

An HTML link is displayed in a different color depending on whether it has been visited, is unvisited, or is active.

### **HTML Link Colors**

By default, a link will appear like this (in all browsers):

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

We can change the link state colors, by using CSS:

### **Example**

Here, an unvisited link will be green with no underline. A visited link will be pink with no underline. An active link will be yellow and underlined. In addition, when mousing over a link (a:hover) it will become red and underlined:

```
<style>
a:link {
  color: green;
  background-color: transparent;
  text-decoration: none;
}

a:visited {
  color: pink;
  background-color: transparent;
  text-decoration: none;
}

a:hover {
  color: red;
  background-color: transparent;
  text-decoration: underline;
}

a:active {
  color: yellow;
  background-color: transparent;
  text-decoration: underline;
}
</style>
```

### **Link Buttons**

A link can also be styled as a button, by using CSS:

## Example

```
<style>
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 15px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: red;
}
</style>
```

## HTML Images

Images can improve the design and the appearance of a web page.

### HTML Images Syntax

The HTML `<img>` tag is used to embed an image in a web page.

Images are not technically inserted into a web page; images are linked to web pages.

The `<img>` tag creates a holding space for the referenced image.

The `<img>` tag is empty, it contains attributes only, and does not have a closing tag.

The `<img>` tag has two required attributes:

- `src` - Specifies the path to the image
- `alt` - Specifies an alternate text for the image

### **Syntax**

```

```

### The src Attribute

The required `src` attribute specifies the path (URL) to the image.

**Note:** When a web page loads, it is the browser, at that moment, that gets the image from a web server and inserts it into the page. Therefore, make sure that the image actually stays in the same spot in relation to the web page, otherwise your visitors will get a broken link icon. The broken link icon and the alt text are shown if the browser cannot find the image.

### The alt Attribute

The required `alt` attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the `src` attribute, or if the user uses a screen reader).

The value of the `alt` attribute should describe the image:

## Example

```

```

If a browser cannot find an image, it will display the value of the alt attribute:

Example

```

```

**Tip:** A screen reader is a software program that reads the HTML code, and allows the user to "listen" to the content. Screen readers are useful for people who are visually impaired or learning disabled.

### **Image Size - Width and Height**

You can use the style attribute to specify the width and height of an image.

Example

```

```

Alternatively, you can use the width and height attributes:

Example

```

```

The width and height attributes always define the width and height of the image in pixels.

**Note:** Always specify the width and height of an image. If width and height are not specified, the web page might flicker while the image loads.

### **Width and Height, or Style**

The width, height, and style attributes are all valid in HTML.

However, we suggest using the style attribute. It prevents styles sheets from changing the size of images:

**Example**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
img {  
  width: 100%;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```

```

```

```

```
</body>
```

```
</html>
```

### **Image Floating**

Use the CSS float property to let the image float to the right or to the left of a text:

#### **Example**

```
<p>
```

The image will float to the right of the text.</p>

```
<p>
```

The image will float to the left of the text.</p>

### **HTML Background Images**

A background image can be specified for almost any HTML element.

#### **Background Image on a HTML element**

To add a background image on an HTML element, use the HTML style attribute and the CSS background-image property:

#### **Example**

Add a background image on a HTML element:

```
<div style="background-image: url('img_girl.jpg');">
```

We can also specify the background image in the <style> element, in the <head> section:

#### **Example**

Specify the background image in the <style> element:

```
<style>
div {
  background-image: url('img_girl.jpg');
}
</style>
```

### **Background Image on a Page**

If you want the entire page to have a background image, you must specify the background image on the <body> element:

#### **Example**

Add a background image for the entire page:

```
<style>
body {
  background-image: url('img_girl.jpg');
}
</style>
```

## **Background Repeat**

If the background image is smaller than the element, the image will repeat itself, horizontally and vertically, until it reaches the end of the element:

### **Example**

```
<style>
body {
  background-image: url('example_img_girl.jpg');
}
</style>
```

To avoid the background image from repeating itself, set the background-repeat property to no-repeat.

### **Example**

```
<style>
body {
  background-image: url('example_img_girl.jpg');
  background-repeat: no-repeat;
}
</style>
```

## **Background Cover**

If you want the background image to cover the entire element, you can set the background-size property to cover.

Also, to make sure the entire element is always covered, set the background-attachment property to fixed:

This way, the background image will cover the entire element, with no stretching (the image will keep its original proportions):

### **Example**

```
<style>
body {
  background-image: url('img_girl.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
</style>
```

## **HTML Tables**

HTML tables allow web developers to arrange data into rows and columns.

### **Example**

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany

Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

## **Define an HTML Table**

A table in HTML consists of table cells inside rows and columns

### **Example**

A simple HTML table:

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

## **Table Cells**

Each table cell is defined by a <td> and a </td> tag.

td stands for table data.

Everything between <td> and </td> are the content of the table cell.

### **Example**

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
</table>
```

**Note:** table data elements are the data containers of the table.

They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

### **Table Rows**

Each table row starts with a `<tr>` and end with a `</tr>` tag.

tr stands for table row.

### **Example**

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

We can have as many rows as you like in a table, just make sure that the number of cells are the same in each row.

**Note:** There are times where a row can have less or more cells than another. You will learn about that in a later chapter.

### **Table Headers**

Sometimes you want your cells to be headers, in those cases use the `<th>` tag instead of the `<td>` tag:

### **Example**

Let the first row be table headers:

```
<table>
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
```

```
<td>10</td>
</tr>
</table>
```

By default, the text in `<th>` elements are bold and centered, but you can change that with CSS.

## **HTML Table Borders**

HTML tables can have borders of different styles and shapes.

### **Adding a Border**

When you add a border to a table, you also add borders around each table cell:


To add a border, use the CSS border property on table, th, and td elements:

### **Example**

```
table, th, td {
  border: 1px solid black;
}
```

### **Collapsed Table Borders**

To avoid having double borders like in the example above, set the CSS border-collapse property to collapse.

This will make the borders collapse into a single border:


### **Example**

```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
```

### **Style Table Borders**

If you set a background color of each cell, and give the border a white color (the same as the document background), you get the impression of an invisible border:




### Example

```
table, th, td {  
  border: 1px solid white;  
  border-collapse: collapse;  
}  
th, td {  
  background-color: #96D4D4;  
}
```

### Round Table Borders

With the border-radius property, the borders get rounded corners:


### Example

```
table, th, td {  
  border: 1px solid black;  
  border-radius: 10px;  
}
```

**Skip the border around the table by leaving out table from the css selector:**


### Example

```
th, td {  
  border: 1px solid black;  
  border-radius: 10px;  
}
```

### Dotted Table Borders

With the border-style property, you can set the appearance of the border.


The following values are allowed:

- dotted
- dashed
- solid
- double
- groove
- ridge

- inset
- outset
- none
- hidden

### Example

```
th, td {
  border-style: dotted;
}
```

### Border Color

With the border-color property, you can set the color of the border.


### Example

```
th, td {
  border-color: red;
}
```

### Table Caption

You can add a caption that serves as a heading for the entire table.

Monthly savings

Month	Savings
January	\$100
February	\$50

To add a caption to a table, use the <caption> tag:

### Example

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

## **HTML Table Padding & Spacing**

HTML tables can adjust the padding inside the cells, and also the space between the cells.

With Padding		
hello	hello	hello
hello	hello	hello
hello	hello	hello

With Spacing		
hello	hello	hello
hello	hello	hello
hello	hello	hello

### **HTML Table - Cell Padding**

Cell padding is the space between the cell edges and the cell content.

By default the padding is set to 0.

To add padding on table cells, use the CSS padding property:

#### **Example**

```
th, td {  
    padding: 15px;  
}
```

To add padding only above the content, use the padding-top property.

And the others sides with the padding-bottom, padding-left, and padding-right properties:

#### **Example**

```
th, td {  
    padding-top: 10px;  
    padding-bottom: 20px;  
    padding-left: 30px;  
    padding-right: 40px;  
}
```

### **HTML Table - Cell Spacing**

Cell spacing is the space between each cell.

By default the space is set to 2 pixels.

To change the space between table cells, use the CSS border-space property on the table element:

#### **Example**

```
table {
  border-spacing: 30px;
}
```

## **HTML Table Colspan & Rowspan**

HTML tables can have cells that spans over multiple rows and/or columns.

<b>NAME</b>		

	<b>APRIL</b>		

<b>2022</b>		
<b>FIESTA</b>		

## **HTML Table - Colspan**

To make a cell span over multiple columns, use the colspan attribute:

### **Example**

```
<table>
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>43</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>57</td>
  </tr>
</table>
```

**Note:** The value of the colspan attribute represents the number of columns to span.

## HTML Table - Rowspan

To make a cell span over multiple rows, use the rowspan attribute:

### Example

```
<table>
  <tr>
    <th>Name</th>
    <td>Jill</td>
  </tr>
  <tr>
    <th rowspan="2">Phone</th>
    <td>555-1234</td>
  </tr>
  <tr>
    <td>555-8745</td>
  </tr>
</table>
```

## HTML Table - Zebra Stripes

If you add a background color on every other table row, you will get a nice zebra stripes effect.

### Example

```
tr:nth-child(even) {
  background-color: blue;
}
```

**Note:** If you use (odd) instead of (even), the styling will occur on row 1,3,5 etc. instead of 2,4,6 etc.

## HTML Table - Vertical Zebra Stripes

To make vertical zebra stripes, style every other *column*, instead of every other *row*.

### Example

```
td:nth-child(even), th:nth-child(even) {
  background-color: #D6EEEE;
}
```

**Note:** Put the :nth-child() selector on both th and td elements if you want to have the styling on both headers and regular table cells.

## Horizontal Dividers

First Name	Last Name	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150

First Name	Last Name	Savings
Joe	Swanson	\$300

If you specify borders only at the bottom of each table row, you will have a table with horizontal dividers.

Add the border-bottom property to all tr elements to get horizontal dividers:

### Example

```
tr {
  border-bottom: 1px solid #ddd;
}
```

### Hoverable Table

Use the :hover selector on tr to highlight table rows on mouse over:

### Example

```
tr:hover {background-color: Red;}
```

### HTML Lists

HTML lists allow web developers to group a set of related items in lists.

### Example

An unordered HTML list:

- Item
- Item
- Item
- Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

### Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

### Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

The CSS list-style-type property is used to define the style of the list item marker. It can have one of the following values:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

### Example - Disc

```
<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

### Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag. The list items will be marked with numbers by default:

#### Example

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

### Ordered HTML List - The Type Attribute

The type attribute of the `<ol>` tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

#### Example

```
<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

## **HTML Description Lists**

A description list is a list of terms, with a description of each term.

The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

### **Example**

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

## **HTML class Attribute**

The HTML class attribute is used to specify a class for an HTML element.

Multiple HTML elements can share the same class.

The class attribute is often used to point to a class name in a style sheet. It can also be used by a JavaScript to access and manipulate elements with the specific class name.

In the following example we have three <div> elements with a class attribute with the value of "city". All of the three <div> elements will be styled equally according to the .city style definition in the head section:

### **Example**

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="city">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>

<div class="city">
  <h2>Paris</h2>
```



```
<p>Paris is the capital of France.</p>
</div>

<div class="city">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```

## **HTML id Attribute**

The HTML id attribute is used to specify a unique id for an HTML element.

We cannot have more than one element with the same id in an HTML document.

The id attribute specifies a unique id for an HTML element. The value of the id attribute must be unique within the HTML document.

The id attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

The syntax for id is: write a hash character (#), followed by an id name. Then, define the CSS properties within curly braces {}.

In the following example we have an <h1> element that points to the id name "myHeader". This <h1> element will be styled according to the #myHeader style definition in the head section:

### **Example**

```
<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>
<body>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```

**Note:** The id name is case sensitive!

**Note:** The id name must contain at least one character, cannot start with a number, and must not contain whitespaces (spaces, tabs, etc.).

## **Difference Between Class and ID**

A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page:

### **Example**

```
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>

<!-- An element with a unique id -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple elements with same class -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
```

## **HTML Iframes**

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

### **HTML Iframe Syntax**

The HTML <iframe> tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

### **Syntax**

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

**Tip:** 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.

### **Iframe - Set Height and Width**

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

The height and width are specified in pixels by default:

#### **Example**

```
<iframe src="demo_iframe.htm" height="200" width="300" title="Iframe Example"></iframe>
```

Or we can add the style attribute and use the CSS height and width properties:

#### **Example**

```
<iframe src="demo_iframe.htm" style="height:200px;width:300px;" title="Iframe Example"></iframe>
```

### **Iframe - Remove the Border**

By default, an iframe has a border around it.

To remove the border, add the style attribute and use the CSS border property:

#### **Example**

```
<iframe src="demo_iframe.htm" style="border:none;" title="Iframe Example"></iframe>
```

With CSS, you can also change the size, style and color of the iframe's border:

#### **Example**

```
<iframe src="demo_iframe.htm" style="border:2px solid red;" title="Iframe Example"></iframe>
```

### **Iframe - Target for a Link**

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**

```
<iframe src="demo_iframe.htm" name="iframe_a" title="Iframe Example"></iframe>
```

```
<p><a href="https://www.google.com" target="iframe_a">Click here</a></p>
```

## **HTML Forms**

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

### **The <form> Element**

The HTML <form> element is used to create an HTML form for user input:

```
<form>
```

.

*form elements*

.

```
</form>
```

The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

### **The <input> Element**

The HTML <input> element is the most used form element.

An <input> element can be displayed in many ways, depending on the type attribute.

Here are some examples:

Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

### **Text Fields**

The <input type="text"> defines a single-line input field for text input.

Example

A form with input fields for text:

```
<form>
```

```
  <label for="fname">First name:</label><br>
```

```
  <input type="text" id="fname" name="fname"><br>
```

```
  <label for="lname">Last name:</label><br>
```

```
  <input type="text" id="lname" name="lname">
```

```
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

**Note:** The form itself is not visible. Also note that the default width of an input field is 20 characters.

### **The <label> Element**

Notice the use of the <label> element in the example above.

The <label> tag defines a label for many form elements.

The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

### **Radio Buttons**

The <input type="radio"> defines a radio button.

Radio buttons let a user select ONE of a limited number of choices.

#### **Example**

A form with radio buttons:

<p>Choose your favorite Web language:</p>

```
<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>
```

This is how the HTML code above will be displayed in a browser:

Choose your favorite Web language:

- ☐ HTML
- ☐ CSS
- ☐ JavaScript

### **Checkboxes**

The <input type="checkbox"> defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

#### **Example**

A form with checkboxes:

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
```

```
<label for="vehicle2"> I have a car</label><br>
<input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
<label for="vehicle3"> I have a boat</label>
</form>
```

This is how the HTML code above will be displayed in a browser:

- ☐ I have a bike
- ☐ I have a car
- ☐ I have a boat

## **The Submit Button**

The `<input type="submit">` defines a button for submitting the form data to a form-handler. The form-handler is typically a file on the server with a script for processing input data. The form-handler is specified in the form's action attribute.

## **Example**

A form with a submit button:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

## **HTML JavaScript**

JavaScript makes HTML pages more dynamic and interactive.

### **The HTML <script> Tag**

The HTML <script> tag is used to define a client-side script (JavaScript).

The <script> element either contains script statements, or it points to an external script file through the src attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

To select an HTML element, JavaScript most often uses the document.getElementById() method.

This JavaScript example writes "Hello JavaScript!" into an HTML element with id="demo":

#### **Example**

```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```

#### **Example**

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

<h1>My First JavaScript</h1>

<p>JavaScript can change the content of an HTML element:</p>

<button type="button" onclick="myFunction()">Click Me!</button>

<p id="demo">This is a demonstration.</p>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>

</body>
</html>
```

#### **Example: JavaScript can change the style of an HTML element**

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

<h1>My First JavaScript</h1>

<p id="demo">JavaScript can change the style of an HTML element.</p>
```

```
<script>
function myFunction() {
  document.getElementById("demo").style.fontSize = "25px";
  document.getElementById("demo").style.color = "red";
  document.getElementById("demo").style.backgroundColor = "yellow";
}
</script>

<button type="button" onclick="myFunction()">Click Me!</button>

</body>
</html>
```

### **The HTML <noscript> Tag**

The HTML <noscript> tag defines an alternate content to be displayed to users that have disabled scripts in their browser or have a browser that doesn't support scripts:

Example

```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
<noscript>Sorry, your browser does not support JavaScript!</noscript>
```

### **Example: JavaScript can change HTML content**

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

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello
JavaScript!'">Click Me!</button>

</body>
</html>
```

### **Example: JavaScript can change the style of an HTML element**

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

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change the style of an HTML element.</p>
```



```
<button type="button"
onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!</button>

</body>
</html>
```

### **Example: JavaScript can hide HTML elements**

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

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can hide HTML elements.</p>

<button type="button"
onclick="document.getElementById('demo').style.display='none'">Click Me!</button>

</body>
</html>
```

### **Example: JavaScript can show hidden HTML elements**

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

<h2>What Can JavaScript Do?</h2>

<p>JavaScript can show hidden HTML elements.</p>

<p id="demo" style="display:none">Hello JavaScript!</p>

<button type="button"
onclick="document.getElementById('demo').style.display='block'">Click Me!</button>

</body>
</html>
```