

HTML Introduction

HTML is the standard markup language for creating Web pages.

What is HTML?

- 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>MyFirstHeading</h1>
<p>Myfirstparagraph.</p>

</body>
</html>
```

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

What is an HTML Element?

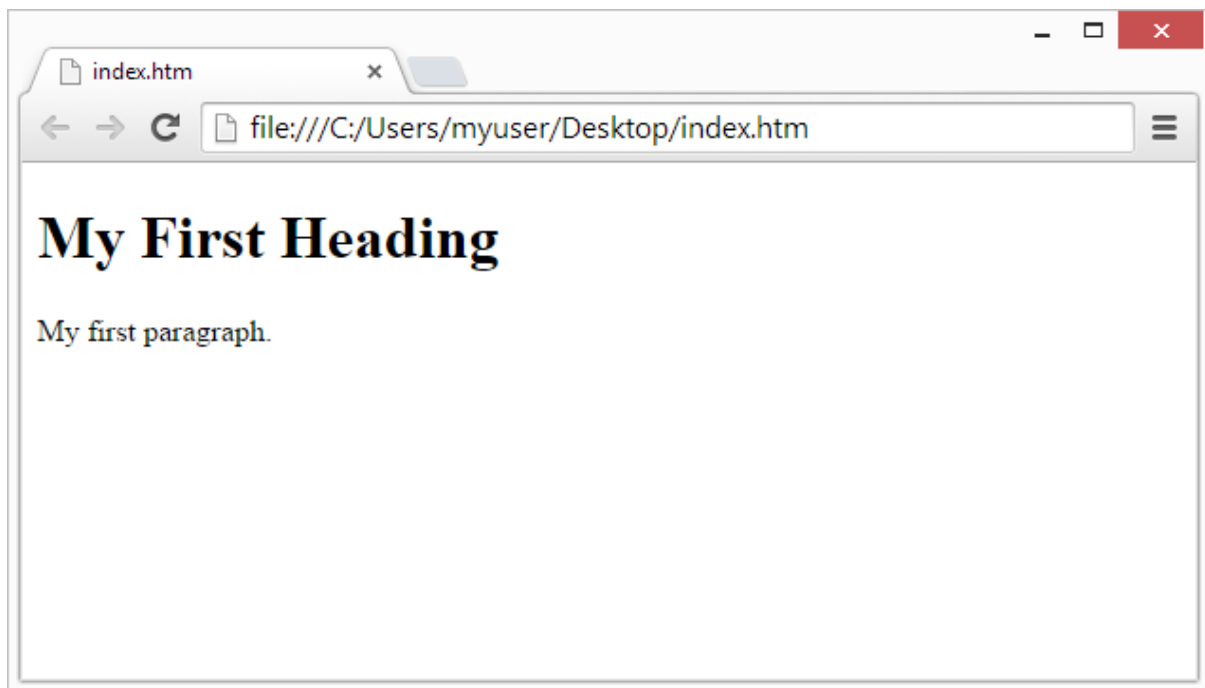
- An HTML element is defined by a start tag, some content, and an end tag:
- `<tagname>Content goes here...</tagname>`
- The HTML **element** is everything from the start tag to the end tag:
- `<h1>My First Heading</h1>`
- `<p>My first paragraph.</p>`

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

Note: Some HTML elements have no content (like the `
` 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: Only the content inside the `<body>` section (the white area above) will be displayed in a browser.

HTML History

Since the early days of the World Wide Web, there have been many versions of HTML:

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0

2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML 5.1
2017	W3C Recommendation: HTML5.1 2nd Edition
2017	W3C Recommendation: HTML5.2

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) 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 1: Open TextEdit (Mac)

Open **Finder > Applications > TextEdit**

Also change some preferences to get the application to save files correctly. In **Preferences > Format** > choose "**Plain Text**"

Then under "Open and Save", check the box that says "Display HTML files as HTML code instead of formatted text".

Then open a new document to place the code.

Step 2: Write Some HTML

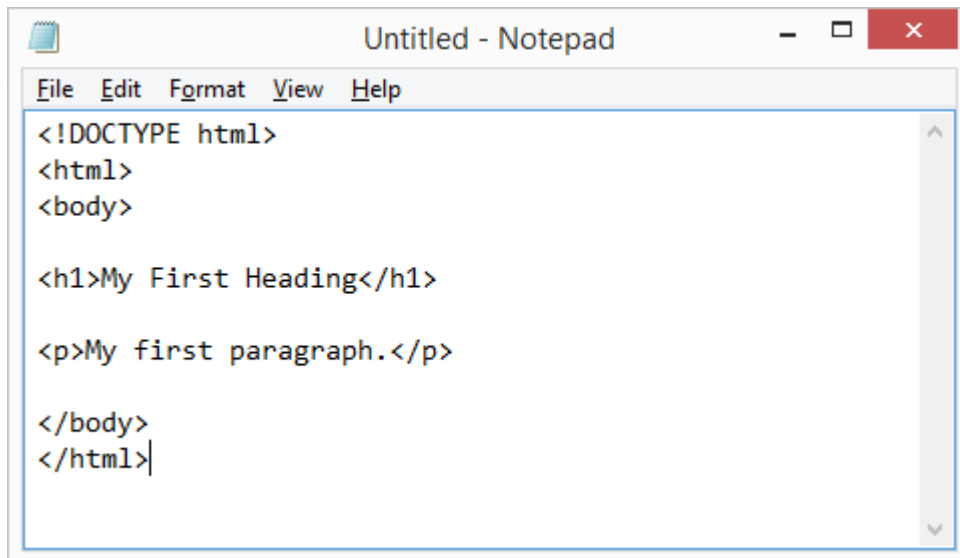
Write or copy the following HTML code into Notepad:

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

<h1>MyFirstHeading</h1>

<p>Myfirstparagraph.</p>

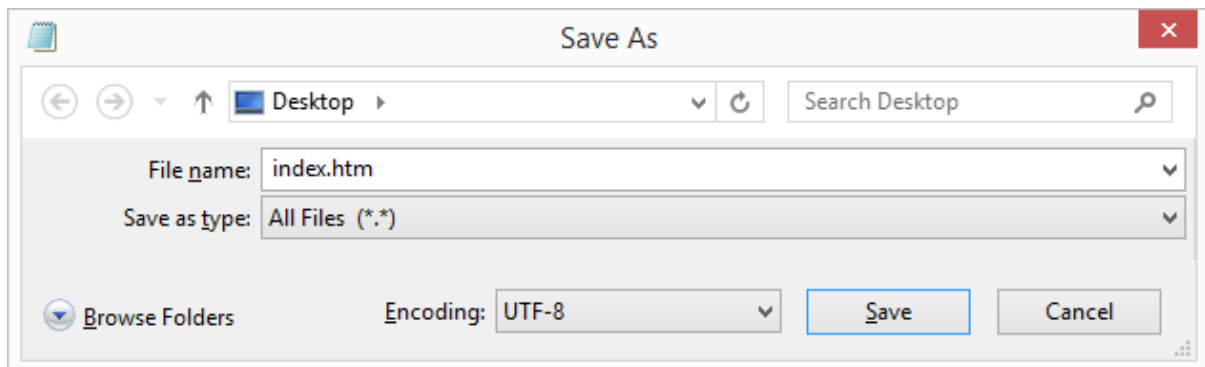
</body>
</html>
```



Step 3: Save the HTML Page

Save the file on your computer. Select **File > Save as** in the Notepad menu.

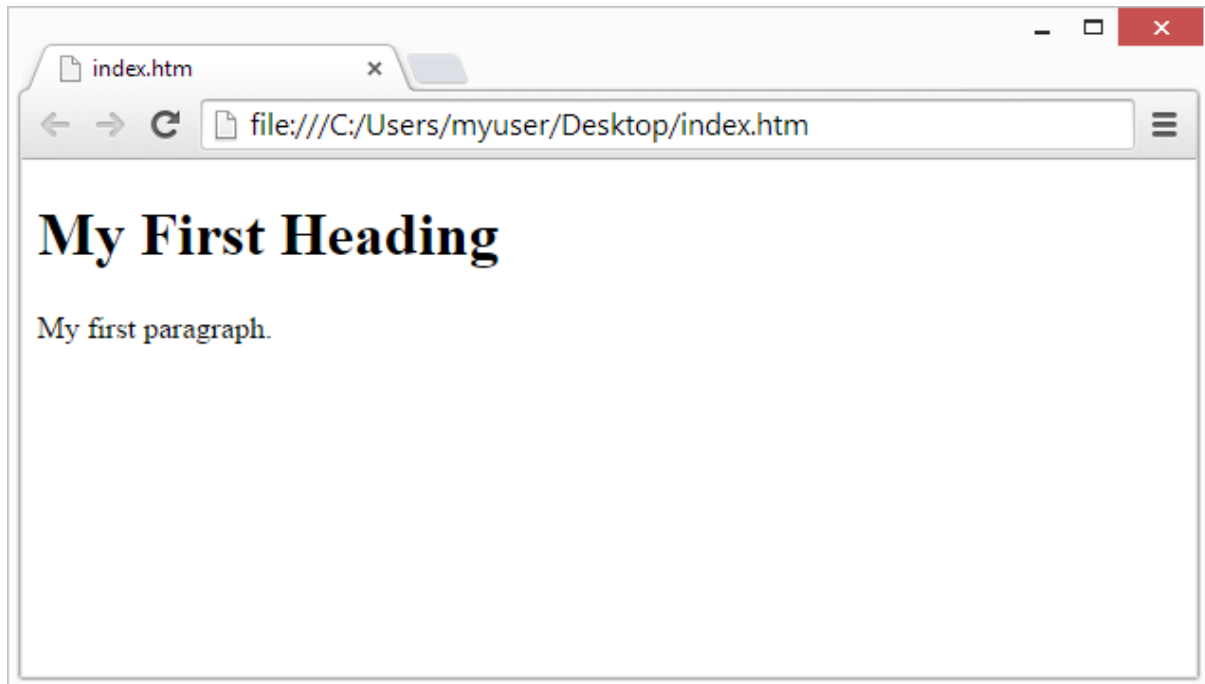
Name the file "**index.htm**" and set the encoding to **UTF-8** (which is the preferred encoding for HTML files).



Tip: You can use either .htm or .html as file extension. There is no difference, it is up to you.

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").
- The result will look much like this:



HTML Basic Examples

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>`.

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 defined with the `<h1>` to `<h6>` tags.
- `<h1>` defines the most important heading. `<h6>` defines the least important heading:

Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

HTML Paragraphs

- HTML paragraphs are defined with the `<p>` tag:

Example

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

HTML Links

- HTML links are defined with the `<a>` tag:

Example

```
<a href="https://www.w3schools.com">This is a link</a>
```

- The link's destination is specified in the `href` attribute.
- Attributes are used to provide additional information about HTML elements.

HTML Images

HTML images are defined with the `` tag.

The source file (`src`), alternative text (`alt`), `width`, and `height` are provided as attributes:

Example

```

```

View HTML Source Code:

Right-click in an HTML page and select "View Page Source" (in Chrome) or "View Source" (in Edge), or similar in other browsers. This will open a window containing the HTML source code of the page.

Inspect an HTML Element:

Right-click on an element (or a blank area), and choose "Inspect" or "Inspect Element" to see what elements are made up of (you will see both the HTML and the CSS). You can also edit the HTML or CSS on-the-fly in the Elements or Styles panel that opens.

HTML Elements

- An HTML element is defined by a start tag, some content, and an end tag.
- 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
<code><h1></code>	My First Heading	<code></h1></code>
<code><p></code>	My first paragraph.	<code></p></code>
<code>
</code>	<i>none</i>	<i>none</i>

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

Nested HTML Elements

- HTML elements can be nested (this means that elements can contain other elements).
- All HTML documents consist of nested HTML elements.

The following example contains four HTML elements (`<html>`, `<body>`, `<h1>` and `<p>`):

Example

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

<h1>MyFirstHeading</h1>
<p>Myfirstparagraph.</p>

</body>
</html>
```

Example Explained

- The `<html>` element is the root element and it defines the whole HTML document.
- It has a start tag `<html>` and an end tag `</html>`.
- Then, inside the `<html>` element there is a `<body>` element:

```
<body>

<h1>MyFirstHeading</h1>
<p>Myfirstparagraph.</p>

</body>
```

- The `<body>` element defines the document's body.
- It has a start tag `<body>` and an end tag `</body>`.

- Then, inside the `<body>` element there are two other elements: `<h1>` and `<p>`:

```
<h1>MyFirstHeading</h1>  
<p>My first paragraph.</p>
```

The `<h1>` element defines a heading.

It has a start tag `<h1>` and an end tag `</h1>`:

```
<h1>My First Heading</h1>
```

The `<p>` element defines a paragraph.

It has a start tag `<p>` and an end tag `</p>`:

```
<p>My first paragraph.</p>
```

Never Skip the End Tag

Some HTML elements will display correctly, even if you forget the end tag:

Example

```
<html>  
<body>  
  
<p>Thisisaparagraph  
<p>Thisisaparagraph  
  
</body>  
</html>
```

However, never rely on this! Unexpected results and errors may occur if you forget the end tag!

Empty HTML Elements

- HTML elements with no content are called empty elements.
- The `
` tag defines a line break, and is an empty element without a closing tag:

Example

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

HTML is Not Case Sensitive

- HTML tags are not case sensitive: `<P>` means the same as `<p>`.
- The HTML standard does not require lowercase tags, but W3C **recommends** lowercase in HTML, and **demand**s lowercase for stricter document types like XHTML.

HTML Tag Reference

W3Schools' tag reference contains additional information about these tags and their attributes.

Tag	Description
<code><html></code>	Defines the root of an HTML document
<code><body></code>	Defines the document's body

HTML Attributes

- HTML attributes provide additional information about HTML elements.

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

The href Attribute

- The `<a>` tag defines a hyperlink. The `href` attribute specifies the URL of the page the link goes to:

Example

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

The src Attribute

- The `` tag is used to embed an image in an HTML page. The `src` attribute specifies the path to the image to be displayed:

Example

```

```

There are two ways to specify the URL in the `src` attribute:

Absolute URL - Links to an external image that is hosted on another website. Example: `src="https://www.w3schools.com/images/img_girl.jpg"`.

Notes: External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.

Relative URL - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page. Example: `src="img_girl.jpg"`. If the URL begins with a slash, it will be relative to the domain. Example: `src="/images/img_girl.jpg"`.

Tip: It is almost always best to use relative URLs. They will not break if you change domain.

The width and height Attributes

The `` tag should also contain the `width` and `height` attributes, which specifies the width and height of the image (in pixels):

Example

```

```


The alt Attribute

The required **alt** attribute for the `` tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the **src** attribute, or if the user uses a screen reader.

Example

```

```

Example

See what happens if we try to display an image that does not exist:

```

```

The style Attribute

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

Example

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

The lang Attribute

- You should always include the **lang** attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers.
- The following example specifies English as the language:

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```

- Country codes can also be added to the language code in the `lang` attribute. So, the first two characters define the language of the HTML page, and the last two characters define the country.

The following example specifies English as the language and United States as the country:

```
<!DOCTYPE html>
<html lang="en-US">
<body>
...
</body>
</html>
```

The title Attribute

- The `title` attribute defines some extra information about an element.
- The value of the title attribute will be displayed as a tooltip when you mouse over the element:

Example

```
<p title="I'm a tooltip">This is a paragraph.</p>
```

We Suggest: Always Use Lowercase Attributes

- The HTML standard does not require lowercase attribute names.
- The title attribute (and all other attributes) can be written with uppercase or lowercase like **title** or **TITLE**.

- However, W3C **recommends** lowercase attributes in HTML, and **demands** lowercase attributes for stricter document types like XHTML.

We Suggest: Always Quote Attribute Values

- The HTML standard does not require quotes around attribute values.
- However, W3C **recommends** quotes in HTML, and **demands** quotes for stricter document types like XHTML.

Good:

```
<a href="https://www.skoda.com/html/">Visit our HTML tutorial</a>
```

Bad:

```
<a href=https://www.skoda.com/html/>Visit our HTML tutorial</a>
```

Sometimes you have to use quotes. This example will not display the title attribute correctly, because it contains a space:

Example

```
<p title>About skoda>
```

we always use quotes around attribute values.

Single or Double Quotes?

- Double quotes around attribute values are the most common in HTML, but single quotes can also be used.
- In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes:

```
<p title='John "ShotGun" Nelson'>
```

Or vice versa:

```
<p title="John 'ShotGun' Nelson">
```

Summary

- All HTML elements can have **attributes**
- The **href** attribute of `<a>` specifies the URL of the page the link goes to
- The **src** attribute of `` specifies the path to the image to be displayed
- The **width** and **height** attributes of `` provide size information for images
- The **alt** attribute of `` provides an alternate text for an image
- The **style** attribute is used to add styles to an element, such as color, font, size, and more
- The **lang** attribute of the `<html>` tag declares the language of the Web page
- The **title** attribute defines some extra information about an element

HTML Attribute Reference

A complete list of all attributes for each HTML element,

HTML Headings

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

Example

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

HTML Headings

- HTML headings are defined with the `<h1>` to `<h6>` tags.
- `<h1>` defines the most important heading. `<h6>` defines the least important heading.

Example

```
<h1>Heading1</h1>
```

```
<h2>Heading2</h2>
```

```
<h3>Heading3</h3>
```

```
<h4>Heading4</h4>
```

```
<h5>Heading5</h5>
```

```
<h6>Heading 6</h6>
```

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

Headings Are Important

- Search engines use the headings to index the structure and content of your web pages.
- Users often skim a page by its headings. It is important to use headings to show the document structure.
- `<h1>` headings should be used for main headings, followed by `<h2>` headings, then the less important `<h3>`, and so on.

Note: Use HTML headings for headings only. Don't use headings to make text BIG or bold.

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 Tag Reference

tag reference contains additional information about these tags and their attributes.

Tag	Description
<html>	Defines the root of an HTML document

`<body>` Defines the document's body

`<h1>` to `<h6>` Defines HTML headings

HTML Paragraphs

- A paragraph always starts on a new line, and is usually a block of text.

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>
```

HTML Display

- You cannot be sure how HTML will be displayed.
- Large or small screens, and resized windows will create different results.
- With HTML, you cannot change the display by adding extra spaces or extra lines in your HTML code.
- The browser will automatically remove any extra spaces and lines when the page is displayed:

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:

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>

- The <hr> tag is an empty tag, which means that it has no end tag.

HTML Line Breaks

- The HTML `
` element defines a line break.
- Use `
` if you want a line break (a new line) without starting a new paragraph:

Example

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

The `
` tag is an empty tag, which means that it has no end tag.

The Poem Problem

This poem will display on a single line:

Example

```
<p>
    My      Bonnie      lies      over      the      ocean.
    My      Bonnie      lies      over      the      sea.
    My      Bonnie      lies      over      the      ocean.
    Oh,     bring      back      my      Bonnie      to      me.
</p>
```

Solution - The HTML `<pre>` Element

- The HTML `<pre>` element defines preformatted text.
- The text inside a `<pre>` element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

Example

```
<pre>
    My      Bonnie      lies      over      the      ocean.

    My      Bonnie      lies      over      the      sea.

    My      Bonnie      lies      over      the      ocean.

Oh,      bring      back      my      Bonnie      to      me.
</pre>
```

HTML Tag Reference

W3Schools' tag reference contains additional information about HTML elements and their attributes.

Tag	Description
<u><p></u>	Defines a paragraph
<u><hr></u>	Defines a thematic change in the content
<u>
</u>	Inserts a single line break
<u><pre></u>	Defines pre-formatted text

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>
```

Chapter Summary

- Use the `style` attribute for styling HTML elements
- Use `background-color` for background color
- Use `color` for text colors
- Use `font-family` for text fonts
- Use `font-size` for text sizes
- Use `text-align` for text alignment

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 _{subscript} and ^{superscript}

HTML Formatting Elements

- Formatting elements were designed to display special types of text:

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

HTML `` and `` Elements

- The HTML `` element defines bold text, without any extra importance.

Example

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

Example

``This text is important! ``

HTML `<i>` and `` 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 `` element defines emphasized text. The content inside is typically displayed in italic.
- **Tip:** A screen reader will pronounce the words in `` with an emphasis, using verbal stress.

Example

``This text is emphasized``

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 `` Element

- The HTML `` 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 ``blue`` 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 ``blue`` `<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 H₂O:

Example

`<p>`This is `_{`subscripted`}` 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^[1]:

Example

`<p>`This is `^{`superscripted`}` text.`</p>`

HTML Text Formatting Elements

Tag	Description
<u></u>	Defines bold text
<u></u>	Defines emphasized text
<u><i></u>	Defines a part of text in an alternate voice or mood
<u><small></u>	Defines smaller text
<u></u>	Defines important text
<u><sub></u>	Defines subscripted text
<u><sup></u>	Defines superscripted text
<u><ins></u>	Defines inserted text
<u></u>	Defines deleted text
<u><mark></u>	Defines marked/highlighted text

HTML Quotation and Citation Elements

- In this module we will go through the `<blockquote>`, `<q>`, `<abbr>`, `<address>`, `<cite>`, and `<bdo>` HTML elements.

Example

Here is a quote from WWF's website:

For nearly 60 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by more than one million members in the United States and close to five million globally.

HTML `<blockquote>` for Quotations

- The HTML `<blockquote>` element defines a section that is quoted from another source.
- Browsers usually indent `<blockquote>` elements.

Example

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

HTML <q> for Short Quotations

- The HTML `<q>` tag defines a short quotation.
- Browsers normally insert quotation marks around the quotation.

Example

`<p>`WWF's goal is to: `<q>`Build a future where people live in harmony with nature. `</q></p>`

HTML <abbr> for Abbreviations

- The HTML `<abbr>` tag defines an abbreviation or an acronym, like "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".
- Marking abbreviations can give useful information to browsers, translation systems and search-engines.
- **Tip:** Use the global title attribute to show the description for the abbreviation/acronym when you mouse over the element.

Example

`<p>`The `<abbr title="World Health Organization">`WHO`</abbr>` was founded in 1948. `</p>`

HTML <address> for Contact Information

- The HTML `<address>` tag defines the contact information for the author/owner of a document or an article.
- The contact information can be an email address, URL, physical address, phone number, social media handle, etc.
- The text in the `<address>` element usually renders in *italic*, and browsers will always add a line break before and after the `<address>` element.

Example

```
<address>
Written by John Doe. <br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>
```

HTML <cite> for Work Title

- The HTML `<cite>` tag defines the title of a creative work (e.g. a book, a poem, a song, a movie, a painting, a sculpture, etc.).

Note: A person's name is not the title of a work.

- The text in the `<cite>` element usually renders in *italic*.

Example

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893. </p>
```

HTML <bdo> for Bi-Directional Override

- BDO stands for Bi-Directional Override.
- The HTML `<bdo>` tag is used to override the current text direction:

Example

```
<bdo dir="rtl">This text will be written from right to left</bdo>
```

HTML Quotation and Citation Elements

Tag	Description
<code><abbr></code>	Defines an abbreviation or acronym
<code><address></code>	Defines contact information for the author/owner of a document
<code><bdo></code>	Defines the text direction
<code><blockquote></code>	Defines a section that is quoted from another source
<code><cite></code>	Defines the title of a work
<code><q></code>	Defines a short inline quotation

HTML Comments

- HTML comments are not displayed in the browser, but they can help document your HTML source code.

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>
```

```
<!-- Remember to add more information here -->
```

- Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors:

Example

```
<!-- Do not display this image at the moment  
  
-->
```

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:





Violet

LightGray

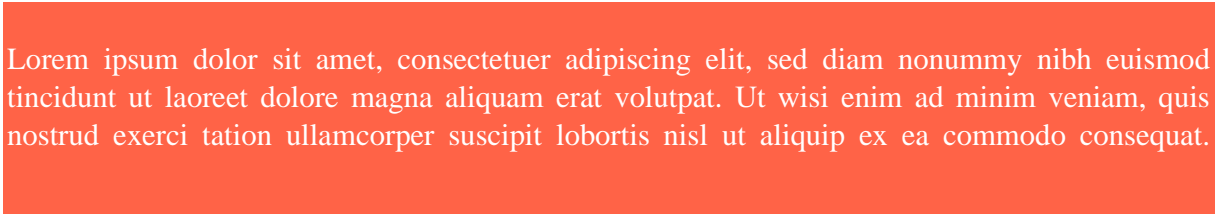
HTML supports [140 standard color names](#).

Background Color

- You can set the background color for HTML elements:



Hello World



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Example

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

Text Color

- You can set the color of text:

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

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:

Hello World

Hello World

Hello World

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):

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 RGB and RGBA Colors

- An RGB color value represents RED, GREEN, and BLUE light sources.
- An RGBA color value is an extension of RGB with an Alpha channel (opacity).

RGB Color Values

- In HTML, a color can be specified as an RGB value, using this formula:

rgb (*red, green, blue*)

- Each parameter (red, green, and blue) defines the intensity of the color with a value between 0 and 255.

- This means that there are $256 \times 256 \times 256 = 16777216$ possible colors!
- For example, rgb (255, 0, 0) is displayed as red, because red is set to its highest value (255), and the other two (green and blue) are set to 0.
- Another example, rgb (0, 255, 0) is displayed as green, because green is set to its highest value (255), and the other two (red and blue) are set to 0.
- To display black, set all color parameters to 0, like this: rgb (0, 0, 0).
- To display white, set all color parameters to 255, like this: rgb (255, 255, 255).
- Experiment by mixing the RGB values below:

rgb (255, 99, 71)

RED

255

GREEN

99

BLUE

71

Example

rgb(255, 0, 0)

rgb(0, 0, 255)

rgb(60, 179, 113)



rgb (238, 130, 238)

rgb (255, 165, 0)

rgb (106, 90, 205)

Shades of Gray

- Shades of gray are often defined using equal values for all three parameters:

Example



rgb (60, 60, 60)

rgb (100, 100, 100)

rgb (140, 140, 140)

rgb (180, 180, 180)

rgb (200, 200, 200)

rgb (240, 240, 240)

RGBA Color Values

- RGBA color values are an extension of RGB color values with an Alpha channel - which specifies the opacity for a color.
- An RGBA color value is specified with:
- **rgba (*red, green, blue, alpha*)**
- The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):
- Experiment by mixing the RGBA values below:

RED

255

GREEN

99

BLUE

71

ALPHA

0.5

Example

`rgba (255, 99, 71, 0)`

`rgba (255, 99, 71, 0.2)`

`rgba (255, 99, 71, 1)`

HTML HEX Colors

- A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color.

HEX Color Values

- In HTML, a color can be specified using a hexadecimal value in the form:

#rrggbb

- Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).
- For example, #ff0000 is displayed as red, because red is set to its highest value (ff), and the other two (green and blue) are set to 00.
- Another example, #00ff00 is displayed as green, because green is set to its highest value (ff), and the other two (red and blue) are set to 00.
- To display black, set all color parameters to 00, like this: #000000.
- To display white, set all color parameters to ff, like this: #ffffff.
- Experiment by mixing the HEX values below:

#ff6347

RED

ff

GREEN

63

BLUE

47

Example

#ff0000



#0000ff

#3cb371

#ee82ee


#ffa500

#6a5acd

Shades of Gray

- Shades of gray are often defined using equal values for all three parameter:

Example



#404040

#686868

#a0a0a0

#bebebe

#dcdcdc

#f8f8f8

HTML HSL and HSLA Colors

- HSL stands for hue, saturation, and lightness.
- HSLA color values are an extension of HSL with an Alpha channel (opacity).

HSL Color Values

- In HTML, a color can be specified using hue, saturation, and lightness (HSL) in the form:
- **hsl (*hue, saturation, lightness*)**
- Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.
- Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.
- Lightness is also a percentage value, 0% is black, and 100% is white.

- Experiment by mixing the HSL values below:

hsl (0, 100%, 50%)

HUE

0

SATURATION

100%

LIGHTNESS

50%

Example

hsl (0, 100%, 50%)

hsl (240, 100%, 50%)

hsl (147, 50%, 47%)

hsl (300, 76%, 72%)

hsl (39, 100%, 50%)



hsl (248, 53%, 58%)

Saturation

- Saturation can be described as the intensity of a color.
- 100% is pure color, no shades of gray
- 50% is 50% gray, but you can still see the color.
- 0% is completely gray, you can no longer see the color.

Example



hsl (0, 100%, 50%)

hsl (0, 80%, 50%)

hsl (0, 60%, 50%)

hsl (0, 40%, 50%)

hsl (0, 20%, 50%)



hsl (0, 0%, 50%)

Lightness

- The lightness of a color can be described as how much light you want to give the color, where 0% means no light (black), 50% means 50% light (neither dark nor light) 100% means full lightness (white).

Example



hsl (0, 100%, 0%)

hsl (0, 100%, 25%)

hsl (0, 100%, 50%)

hsl (0, 100%, 75%)

hsl (0, 100%, 90%)

hsl (0, 100%, 100%)

Shades of Gray

- Shades of gray are often defined by setting the hue and saturation to 0, and adjust the lightness from 0% to 100% to get darker/lighter shades:

Example

hsl (0, 0%, 20%)

hsl (0, 0%, 30%)

hsl (0, 0%, 40%)

hsl (0, 0%, 60%)

hsl (0, 0%, 70%)

hsl (0, 0%, 90%)

HSLA Color Values

- HSLA color values are an extension of HSL color values with an Alpha channel - which specifies the opacity for a color.
- An HSLA color value is specified with:
- **hsla** (*hue, saturation, lightness, alpha*)
- The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):
- Experiment by mixing the HSLA values below:

HUE

0

SATURATION

100%

LIGHTNESS

50%

ALPHA

0.5

Example

hsla(9, 100%, 64%, 0)

hsla (9, 100%, 64%, 0.2)

hsla (9, 100%, 64%, 1)

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.

CSS = Styles and Colors

M a n i p u l a t e T e x t

C o l o r s , **B o x e s**

What is CSS?

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

With CSS, you can control the color, font, 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

- Here, we will demonstrate some commonly used CSS properties. You will learn more about them later.
- 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;  
}
```

Link to External CSS

- External style sheets can be referenced with a full URL or with a path relative to the current web page.

Example

This example uses a full URL to link to a style sheet:

```
<link rel="stylesheet" href="https://www.w3schools.com/html/styles.css">
```

Example

This example links to a style sheet located in the html folder on the current web site:

```
<link rel="stylesheet" href="/html/styles.css">
```

Example

This example links to a style sheet located in the same folder as the current page:

```
<link rel="stylesheet" href="styles.css">
```

Chapter Summary

- Use the HTML `style` attribute for inline styling
- Use the HTML `<style>` element to define internal CSS
- Use the HTML `<link>` element to refer to an external CSS file
- Use the HTML `<head>` element to store `<style>` and `<link>` elements
- Use the CSS `color` property for text colors
- Use the CSS `font-family` property for text fonts
- Use the CSS `font-size` property for text sizes
- Use the CSS `border` property for borders
- Use the CSS `padding` property for space inside the border
- Use the CSS `margin` property for space outside the border.

HTML Style Tags

Tag	Description
<style>	Defines style information for an HTML document
<link>	Defines a link between a document and an external resource

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:

- `link text`
- 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:
- `Visit W3Schools.com! `

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>
```

Absolute URLs vs. Relative URLs

- Both examples above are using an **absolute URL** (a full web address) in the `href` attribute.
- A local link (a link to a page within the same website) is specified with a **relative URL** (without the "https://www" part):

Example

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

HTML Links - Use an Image as a Link

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

Example

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

</a>
```


Link to an Email Address

- Use **mailto:** inside the **href** attribute to create a link that opens the user's email program (to let them send a new email):

Example

```
<a href="mailto:someone@example.com">Send email</a>
```

Button as a Link

- To use an HTML button as a link, you have to add some JavaScript code.
- JavaScript allows you to specify what happens at certain events, such as a click of a button:

Example

```
<button onclick="document. Location='default.asp'">HTML Tutorial</button>
```

Link Titles

- The **title** attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

Example

```
<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML section">Visit our HTML Tutorial</a>
```

More on Absolute URLs and Relative URLs

Example

Use a full URL to link to a web page:

```
<a href="https://www.skoda.com/html/default.asp">HTML</a>
```

Example

Link to a page located in the html folder on the current web site:

```
<a href="/html/default.asp">HTML 1</a>
```

Example

Link to a page located in the same folder as the current page:

```
<a href="default.asp">HTML tutorial</a>
```

Chapter Summary

- Use the `<a>` element to define a link
- Use the `href` attribute to define the link address
- Use the `target` attribute to define where to open the linked document
- Use the `` element (inside `<a>`) to use an image as a link
- Use the `mailto:` scheme inside the `href` attribute to create a link that opens the user's email program

HTML Link Tags

Tag	Description
<u><a></u>	Defines a hyperlink

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

You 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;

```

HTML Link Tags

Tag	Description
<u><a></u>	Defines a hyperlink

HTML Links - Create Bookmarks

- HTML links can be used to create bookmarks, so that readers can jump to specific parts of a web page.

Create a Bookmark in HTML

- Bookmarks can be useful if a web page is very long.
- To create a bookmark - first create the bookmark, then add a link to it.
- When the link is clicked, the page will scroll down or up to the location with the bookmark.

Example

- First, use the **id** attribute to create a bookmark:
- `<h2 id="C4">Chapter 4</h2>`
- Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

Example

`Jump to Chapter 4`

You can also add a link to a bookmark on another page:

`Jump to Chapter 4`

Chapter Summary

- Use the **id** attribute (`id="value"`) to define bookmarks in a page
- Use the **href** attribute (`href="#value"`) to link to the bookmark

HTML Link Tags

Tag	Description
<code><a></code>	Defines a hyperlink

HTML Images

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

Example

```

```

Example

```

```

Example

```

```

HTML Images Syntax

- The HTML `` 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 `` tag creates a holding space for the referenced image.
- The `` tag is empty, it contains attributes only, and does not have a closing tag.
- The `` 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.

Example

- ```

```

## 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>
```

## Images in Another Folder

- If you have your images in a sub-folder, you must include the folder name in the `src` attribute:

### Example

```

```

## Images on Another Server/Website

- Some web sites point to an external image on another server.

- To point to an image on another server, you must specify an absolute (full) URL in the `src` attribute:

## Example

```

```

- **Notes on external images:** External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.

## Animated Images

- HTML allows animated GIFs:

## Example

- ``

## Image as a Link

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

## Example

```



```

# 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>

## Common Image Formats

- Here are the most common image file types, which are supported in all browsers (Chrome, Edge, Firefox, Safari, Opera):

Abbreviation	File Format	File Extension
APNG	Animated Portable Network Graphics	. apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	. icon, .cur
JPEG	Joint Photographic Expert Group image	. jpg,. jpeg,. jfif,. pjpeg, .jpg
PNG	Portable Network Graphics	.png

## Chapter Summary

- Use the HTML `<img>` element to define an image
- Use the HTML `src` attribute to define the URL of the image
- Use the HTML `alt` attribute to define an alternate text for an image, if it cannot be displayed
- Use the HTML `width` and `height` attributes or the CSS `width` and `height` properties to define the size of the image
- Use the CSS `float` property to let the image float to the left or to the right

Note: Loading large images takes time, and can slow down your web page. Use images carefully.

## HTML Image Tags

Tag	Description
<code>&lt;img&gt;</code>	Defines an image
<code>&lt;map&gt;</code>	Defines an image map
<code>&lt;area&gt;</code>	Defines a clickable area inside an image map

<picture>

Defines a container for multiple image resources

## HTML Image Maps

- With HTML image maps, you can create clickable areas on an image.

### Image Maps

- The HTML `<map>` tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more `<area>` tags.
- Try to click on the computer, phone, or the cup of coffee in the image below:



## Example

Here is the HTML source code for the image map above:

```


<map name="workmap">
 <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
 <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
 <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>
```

## How Does it Work?

- The idea behind an image map is that you should be able to perform different actions depending on where in the image you click.
- To create an image map you need an image, and some HTML code that describes the clickable areas.

## The Image

- The image is inserted using the `<img>` tag. The only difference from other images is that you must add a `usemap` attribute:
- ``

The `usemap` value starts with a hash tag `#` followed by the name of the image map, and is used to create a relationship between the image and the image map.

## Create Image Map

- Then, add a `<map>` element.
- The `<map>` element is used to create an image map, and is linked to the image by using the required `name` attribute:
- `<map name="workmap">`
- The `name` attribute must have the same value as the `<img>`'s `usemap` attribute.

## The Areas

- Then, add the clickable areas.
- A clickable area is defined using an `<area>` element.

## Shape

You must define the shape of the clickable area, and you can choose one of these values:

- `rect` - defines a rectangular region
- `circle` - defines a circular region
- `poly` - defines a polygonal region
- `default` - defines the entire region

You must also define some coordinates to be able to place the clickable area onto the image.



## Shape="rect"

- The coordinates for `shape="rect"` come in pairs, one for the x-axis and one for the y-axis.
- So, the coordinates `34,44` is located 34 pixels from the left margin and 44 pixels from the top:



- The coordinates `270,350` is located 270 pixels from the left margin and 350 pixels from the top:



Now we have enough data to create a clickable rectangular area:

## Example

- `<area shape="rect" coords="34, 44, 270, 350" href="computer.htm">`
- This is the area that becomes clickable and will send the user to the page "computer.htm":



## Shape="circle"

- To add a circle area, first locate the coordinates of the center of the circle:

337,300



- Then specify the radius of the circle:

44 pixels



- Now you have enough data to create a clickable circular area:

## Example

- `<area shape="circle" coords="337, 300, 44" href="coffee.htm">`
- This is the area that becomes clickable and will send the user to the page "coffee.htm":



## Shape="poly"

- The `shape="poly"` contains several coordinate points, which creates a shape formed with straight lines (a polygon).
- This can be used to create any shape.
- Like maybe a croissant shape!
- How can we make the croissant in the image below become a clickable link?





- We have to find the x and y coordinates for all edges of the croissant:



- The coordinates come in pairs, one for the x-axis and one for the y-axis:

### Example

```
<area shape="poly" coords="140,121,181,116,204,160,204,222,191,270,140,329,85,355,58,352,37,322,40,259,103,161,128,147" href="croissant.htm">
```

This is the area that becomes clickable and will send the user to the page "croissant.htm":



## Image Map and JavaScript

- A clickable area can also trigger a JavaScript function.
- Add a **click** event to the **<area>** element to execute a JavaScript function:



## Example

Here, we use the onclick attribute to execute a JavaScript function when the area is clicked:

```
<map name="workmap">
 <area shape="circle" coords="337,300,44" onclick="myFunction()">
</map>

<script>
function myFunction() {
 alert("You clicked the coffee cup!");
}
</script>
```

## Summary

- Use the HTML `<map>` element to define an image map
- Use the HTML `<area>` element to define the clickable areas in the image map
- Use the HTML `usemap` attribute of the `<img>` element to point to an image map

## HTML Image Tags

Tag	Description
<a href="#"><code>&lt;img&gt;</code></a>	Defines an image
<a href="#"><code>&lt;map&gt;</code></a>	Defines an image map
<a href="#"><code>&lt;area&gt;</code></a>	Defines a clickable area inside an image map

`<picture>`

Defines a container for multiple image resources

# 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');">
```

- You 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>
```

## Background Stretch

- If you want the background image to stretch to fit the entire element, you can set the **background-size** property to **100% 100%**:
- Try resizing the browser window, and you will see that the image will stretch, but always cover the entire element.

### Example

```
<style>
body {
 background-image: url('img_girl.jpg');
 background-repeat: no-repeat;
 background-attachment: fixed;
 background-size: 100% 100%;
}
</style>
```

# HTML <picture> Element

- The HTML `<picture>` element allows you to display different pictures for different devices or screen sizes.



## The HTML <picture> Element

- The HTML `<picture>` element gives web developers more flexibility in specifying image resources.
- The `<picture>` element contains one or more `<source>` elements, each referring to different images through the `srcset` attribute. This way the browser can choose the image that best fits the current view and/or device.
- Each `<source>` element has a `media` attribute that defines when the image is the most suitable.

## Example

Show different images for different screen sizes:

```
<picture>
 <source media="(min-width: 650px)" srcset="img_food.jpg">
 <source media="(min-width: 465px)" srcset="img_car.jpg">

</picture>
```

**Note:** Always specify an `<img>` element as the last child element of the `<picture>` element. The `<img>` element is used by browsers that do not support the `<picture>` element, or if none of the `<source>` tags matches.

## When to use the Picture Element

There are two main purposes for the `<picture>` element:

### 1. Bandwidth

If you have a small screen or device, it is not necessary to load a large image file. The browser will use the first `<source>` element with matching attribute values, and ignore any of the following elements.

### 2. Format Support

Some browsers or devices may not support all image formats. By using the `<picture>` element, you can add images of all formats, and the browser will use the first format it recognizes, and ignore any of the following elements.

## Example

The browser will use the first image format it recognizes:

```
<picture>
 <source srcset="img_avatar.png">
 <source srcset="img_girl.jpg">

</picture>
```

**Note:** The browser will use the first `<source>` element with matching attribute values, and ignore any following `<source>` elements.

## HTML Image Tags

Tag	Description
<a href="#"><code>&lt;img&gt;</code></a>	Defines an image
<a href="#"><code>&lt;map&gt;</code></a>	Defines an image map
<a href="#"><code>&lt;area&gt;</code></a>	Defines a clickable area inside an image map
<a href="#"><code>&lt;picture&gt;</code></a>	Defines a container for multiple image resources