# HTML

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

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

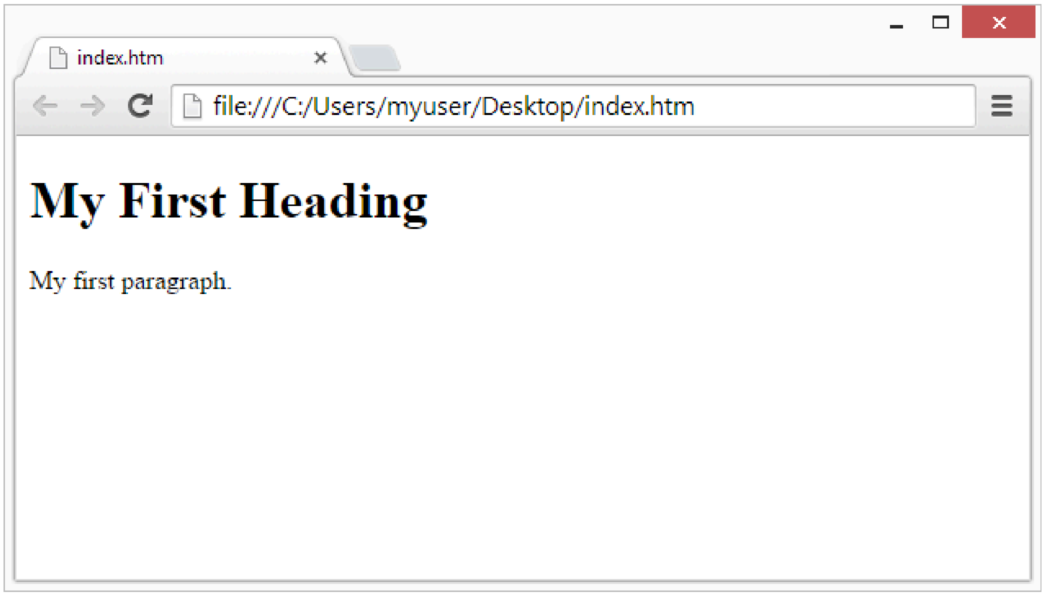
| Start tag | Element content | End tag |
|---|---|---|
| <h1> | My First Heading | </h1> |
| <p> | My first paragraph. | </p> |
| <br> | *none* | *none* |

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:

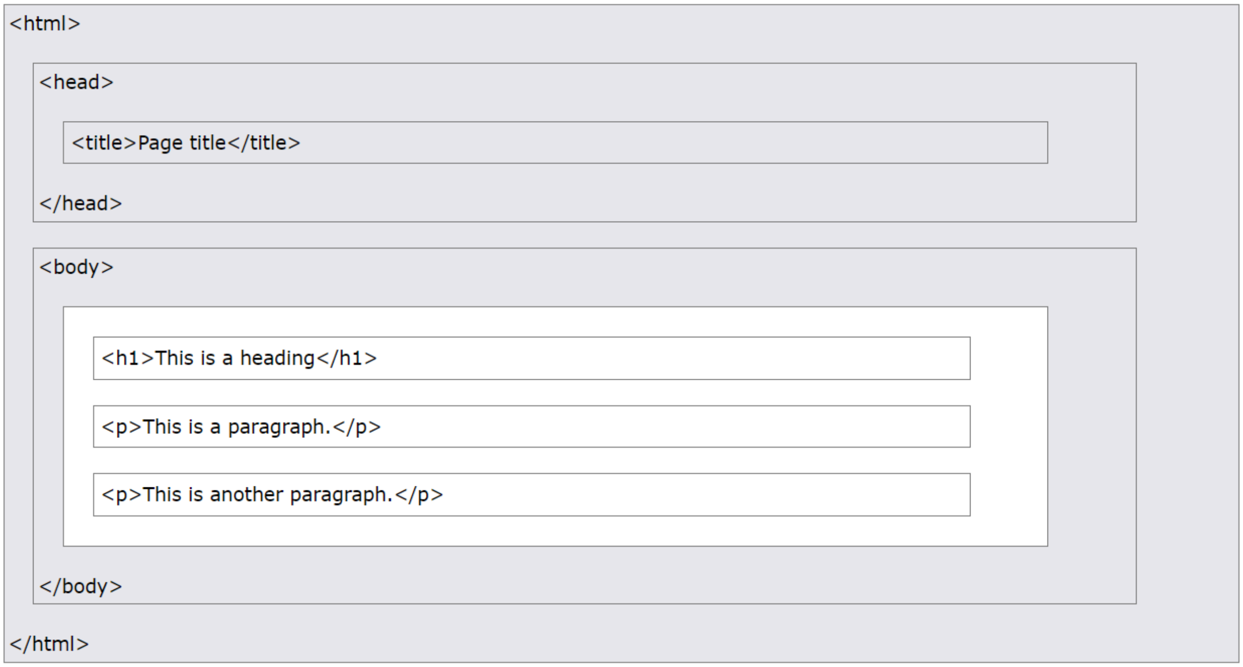


**My First Heading**

My first paragraph.

# 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 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 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 |

## Empty HTML Elements

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

### Example

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

## HTML is Not Case Sensitive

HTML tags are not case sensitive: `<P>` means the same as `<p>`.

# 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.jetking.com/">Visit Jetking</a>
```

## The src Attribute

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

### Example

```
<img src="img_girl.jpg">
```

# HTML Headings

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

### Example

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

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

</body>
</html>
```

# 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

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

<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>

<p>
```

```
The number of lines in a paragraph depends on the size of the browser window. If you resize
the browser window, the number of lines in this paragraph will change.
</p>

</body>
</html>
```

# 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

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

<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>

<h2>This is heading 2</h2>
<p>This is some other text.</p>

</body>
</html>
```

# 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

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

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

</body>
</html>
```

# The Poem Problem

This poem will display on a single line:

## Example

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

<p>In HTML, spaces and new lines are ignored:</p>

<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>

</body>
</html>
```

# 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

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

<p>The pre tag preserves both spaces and line breaks:</p>

<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>

</body>
</html>
```

# HTML Styles

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

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

<p>I am normal</p>
<p style="color:red;">I am red</p>
<p style="color:blue;">I am blue</p>
<p style="font-size:50px;">I am big</p>

</body>
</html>
```

# 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:

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

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

</body>
</html>
```

## Example

Set background color for two different elements:

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

<h1 style="background-color:powderblue;">This is a heading</h1>
<p style="background-color:tomato;">This is a paragraph.</p>

</body>
</html>
```

# 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.

## 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

```
<strong>This text is important!</strong>
```

## 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 $H_2O$:

### 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[1]:

### Example

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

# HTML Quotation and Citation Elements

```
<!DOCTYPE html>
<html>
<body>
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 60 years, WWF has worked to help people and nature thrive. As the
world's leading conservation organization, WWF works in nearly 100
countries. At every level, we collaborate with people around the world to
develop and deliver innovative solutions that protect communities,
wildlife, and the places in which they live.
</blockquote>
</body>
</html>
```

## 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 Comment Tag

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.

# HTML Colors

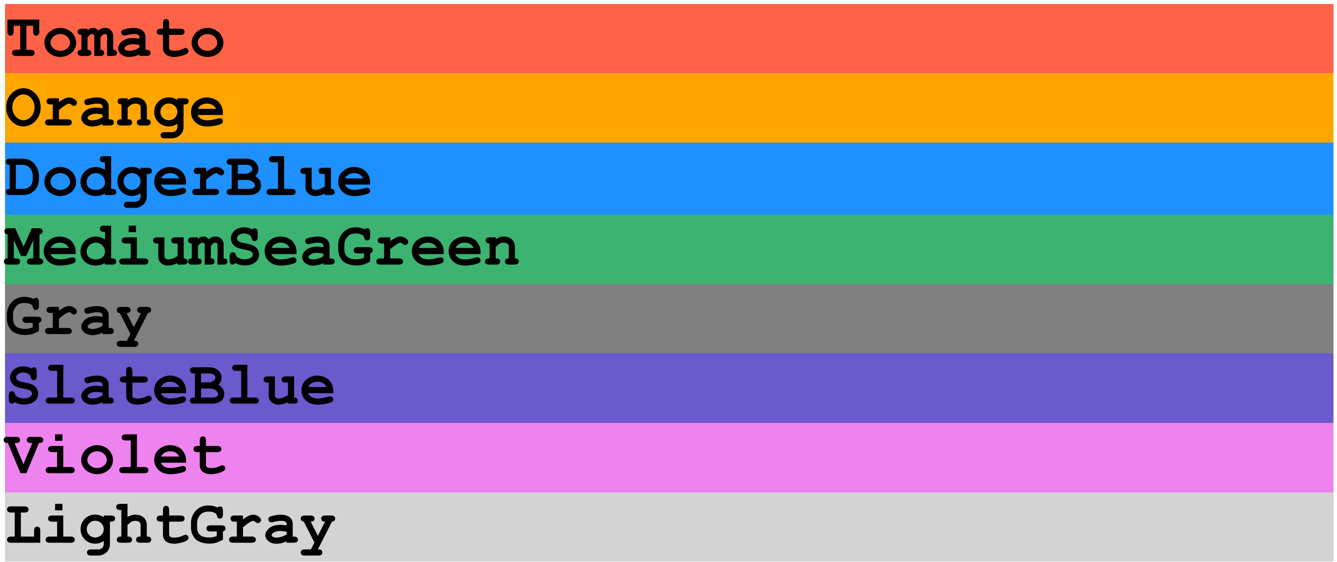
HTML colors are specified with predefined color names, or with RGB, HEX, HSL, RGBA, or HSLA values.

---

## Color Names

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

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:Tomato;">Tomato</h1>
<h1 style="background-color:Orange;">Orange</h1>
<h1 style="background-color:DodgerBlue;">DodgerBlue</h1>
<h1 style="background-color:MediumSeaGreen;">MediumSeaGreen</h1>
<h1 style="background-color:Gray;">Gray</h1>
<h1 style="background-color:SlateBlue;">SlateBlue</h1>
<h1 style="background-color:Violet;">Violet</h1>
<h1 style="background-color:LightGray;">LightGray</h1>
</body>
</html>
```



```
Tomato
Orange
DodgerBlue
MediumSeaGreen
Gray
SlateBlue
Violet
LightGray
```

## Background Color

You 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

You 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.

# 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

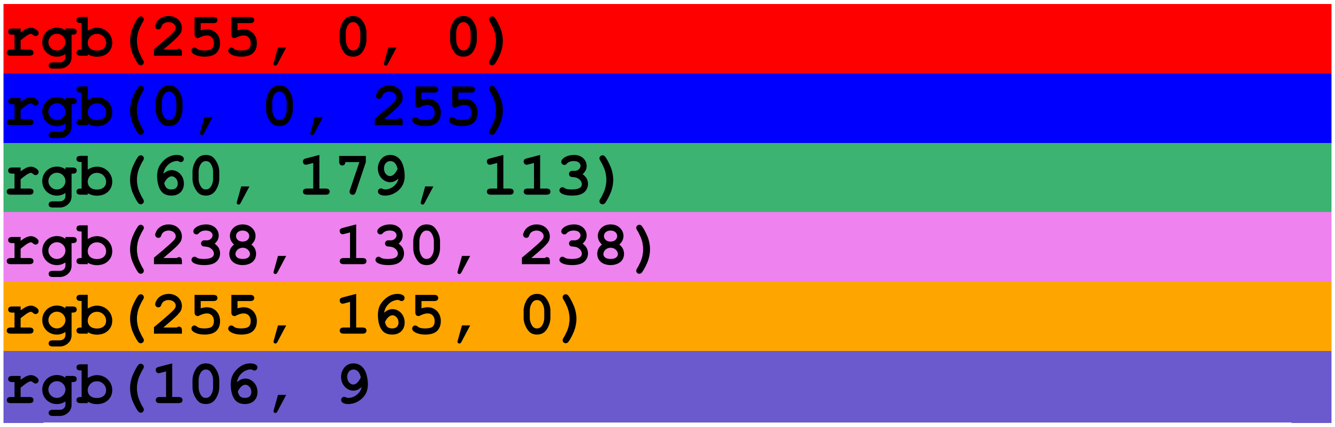
In HTML, a color can be specified as an RGB value, using this formula:

**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 x 256 x 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).

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:rgb(255, 0, 0);">rgb(255, 0, 0)</h1>
<h1 style="background-color:rgb(0, 0, 255);">rgb(0, 0, 255)</h1>
```

```
<h1 style="background-color:rgb(60, 179, 113);">rgb(60, 179, 113)</h1>
<h1 style="background-color:rgb(238, 130, 238);">rgb(238, 130, 238)</h1>
<h1 style="background-color:rgb(255, 165, 0);">rgb(255, 165, 0)</h1>
<h1 style="background-color:rgb(106, 90, 205);">rgb(106, 90, 205)</h1>
</body>
</html>
```



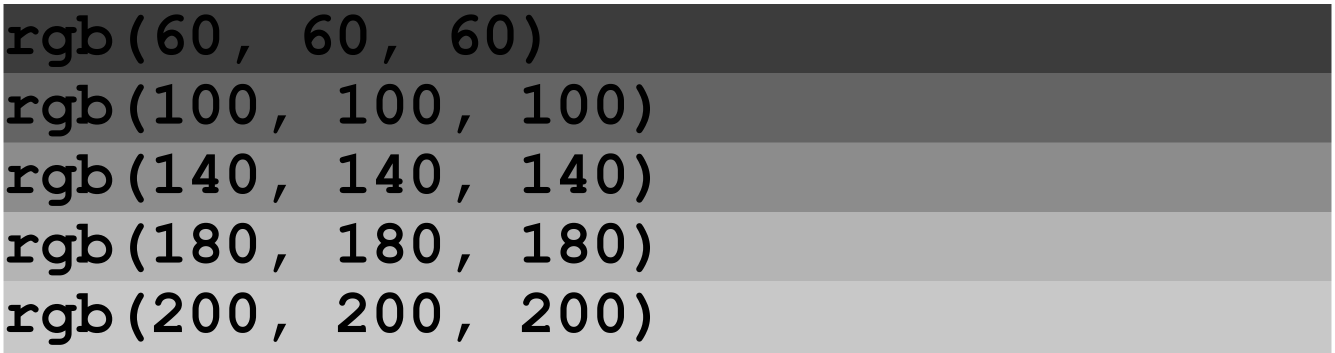
```
rgb(255, 0, 0)
rgb(0, 0, 255)
rgb(60, 179, 113)
rgb(238, 130, 238)
rgb(255, 165, 0)
rgb(106, 9
```

## Shades of Gray

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

### Example

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:rgb(60, 60, 60);">rgb(60, 60, 60)</h1>
<h1 style="background-color:rgb(100, 100, 100);">rgb(100, 100, 100)</h1>
<h1 style="background-color:rgb(140, 140, 140);">rgb(140, 140, 140)</h1>
<h1 style="background-color:rgb(180, 180, 180);">rgb(180, 180, 180)</h1>
<h1 style="background-color:rgb(200, 200, 200);">rgb(200, 200, 200)</h1>
<h1 style="background-color:rgb(240, 240, 240);">rgb(240, 240, 240)</h1>
</body>
</html>
```



```
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):

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:rgba(255, 99, 71, 0);">rgba(255, 99, 71,
0)</h1>
<h1 style="background-color:rgba(255, 99, 71, 0.2);">rgba(255, 99, 71,
0.2)</h1>
<h1 style="background-color:rgba(255, 99, 71, 0.4);">rgba(255, 99, 71,
0.4)</h1>
<h1 style="background-color:rgba(255, 99, 71, 0.6);">rgba(255, 99, 71,
0.6)</h1>
<h1 style="background-color:rgba(255, 99, 71, 0.8);">rgba(255, 99, 71,
0.8)</h1>
<h1 style="background-color:rgba(255, 99, 71, 1);">rgba(255, 99, 71,
1)</h1>
</body>
</html>
```

# rgba(255, 99, 71, 0)

# rgba(255, 99, 71, 0.2)

# rgba(255, 99, 71, 0.4)

# rgba(255, 99, 71, 0.6)

# rgba(255, 99, 71, 0.8)

# 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).

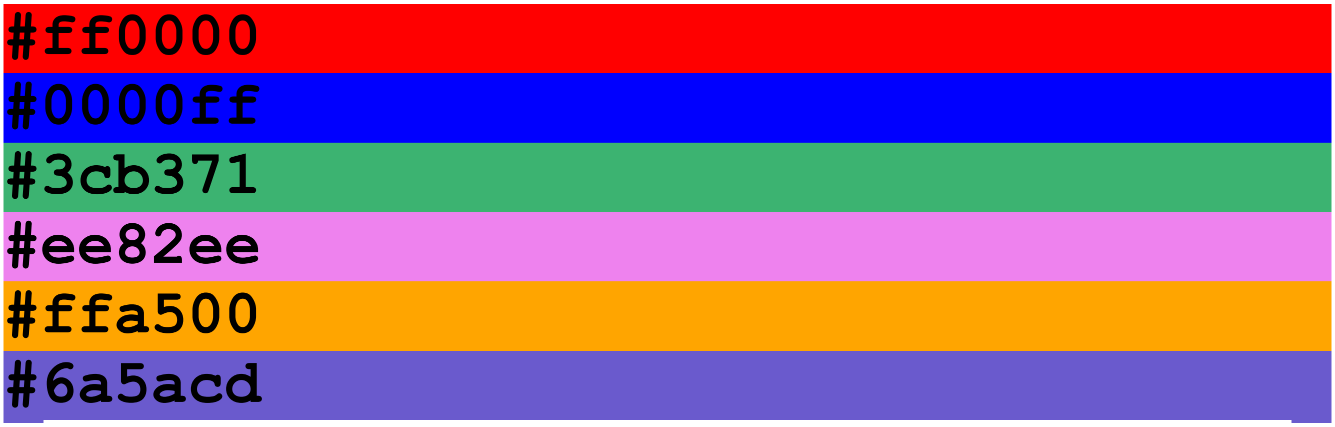
For example, #ff0000 is displayed as red, because red is set to its highest value (ff), and the other two (green and blue) are set to 00.

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.

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:#ff0000;">#ff0000</h1>
<h1 style="background-color:#0000ff;">#0000ff</h1>
<h1 style="background-color:#3cb371;">#3cb371</h1>
<h1 style="background-color:#ee82ee;">#ee82ee</h1>
<h1 style="background-color:#ffa500;">#ffa500</h1>
<h1 style="background-color:#6a5acd;">#6a5acd</h1>
</body>
</html>
```



# #ff0000
# #0000ff
# #3cb371
# #ee82ee
# #ffa500
# #6a5acd

## Shades of Gray

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

### Example

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:#404040;">#404040</h1>
<h1 style="background-color:#686868;">#686868</h1>
<h1 style="background-color:#a0a0a0;">#a0a0a0</h1>
<h1 style="background-color:#bebebe;">#bebebe</h1>
<h1 style="background-color:#dcdcdc;">#dcdcdc</h1>
<h1 style="background-color:#f8f8f8;">#f8f8f8</h1>
</body>
</html>
```

**#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

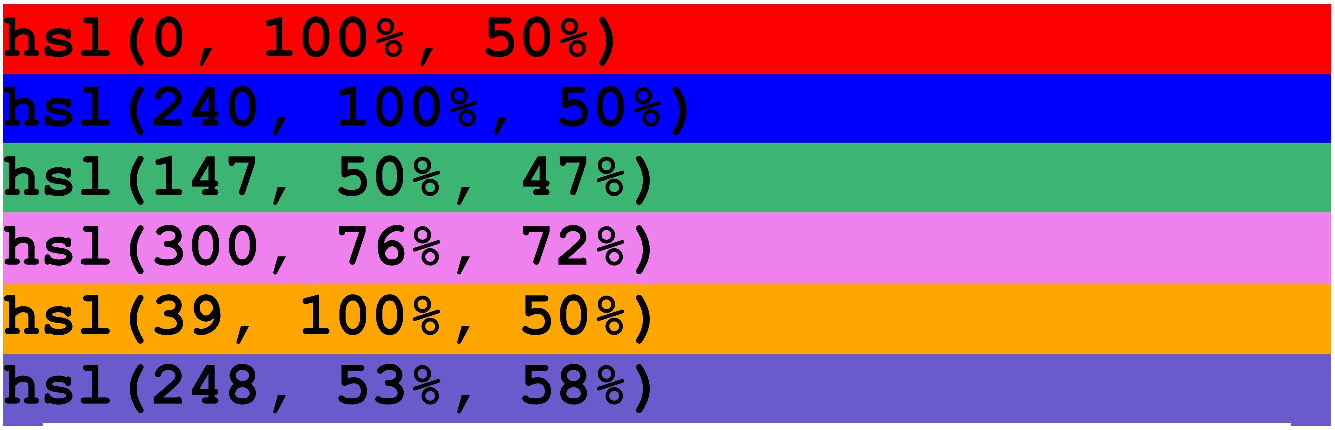
In HTML, a color can be specified using hue, saturation, and lightness (HSL) in the form:

**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.

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>
<h1 style="background-color:hsl(240, 100%, 50%);">hsl(240, 100%,
50%)</h1>
<h1 style="background-color:hsl(147, 50%, 47%);">hsl(147, 50%, 47%)</h1>
<h1 style="background-color:hsl(300, 76%, 72%);">hsl(300, 76%, 72%)</h1>
<h1 style="background-color:hsl(39, 100%, 50%);">hsl(39, 100%, 50%)</h1>
<h1 style="background-color:hsl(248, 53%, 58%);">hsl(248, 53%, 58%)</h1>
</body>
</html>
```



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

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>
<h1 style="background-color:hsl(0, 80%, 50%);">hsl(0, 80%, 50%)</h1>
<h1 style="background-color:hsl(0, 60%, 50%);">hsl(0, 60%, 50%)</h1>
<h1 style="background-color:hsl(0, 40%, 50%);">hsl(0, 40%, 50%)</h1>
<h1 style="background-color:hsl(0, 20%, 50%);">hsl(0, 20%, 50%)</h1>
<h1 style="background-color:hsl(0, 0%, 50%);">hsl(0, 0%, 50%)</h1>
<p>With HSL colors, less saturation mean less color. 0% is completely
gray.</p>
</body>
</html>
```

### 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), and 100% means full lightness (white).

### Example

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:hsl(0, 100%, 0%);">hsl(0, 100%, 0%)</h1>
<h1 style="background-color:hsl(0, 100%, 25%);">hsl(0, 100%, 25%)</h1>
<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>
<h1 style="background-color:hsl(0, 100%, 75%);">hsl(0, 100%, 75%)</h1>
<h1 style="background-color:hsl(0, 100%, 90%);">hsl(0, 100%, 90%)</h1>
<h1 style="background-color:hsl(0, 100%, 100%);">hsl(0, 100%, 100%)</h1>
<p>With HSL colors, 0% lightness means black, and 100 lightness means
white.</p>
</body>
</html>
```

## Shades of Gray

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

### Example

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:hsl(0, 0%, 20%);">hsl(0, 0%, 20%)</h1>
<h1 style="background-color:hsl(0, 0%, 30%);">hsl(0, 0%, 30%)</h1>
<h1 style="background-color:hsl(0, 0%, 40%);">hsl(0, 0%, 40%)</h1>
<h1 style="background-color:hsl(0, 0%, 60%);">hsl(0, 0%, 60%)</h1>
<h1 style="background-color:hsl(0, 0%, 70%);">hsl(0, 0%, 70%)</h1>
<h1 style="background-color:hsl(0, 0%, 90%);">hsl(0, 0%, 90%)</h1>
</body>
</html>
```

**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):

```
<!DOCTYPE html>
<html>
<body>
<h1 style="background-color:hsla(9, 100%, 64%, 0);">hsla(9, 100%, 64%,
0)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.2);">hsla(9, 100%, 64%,
0.2)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.4);">hsla(9, 100%, 64%,
0.4)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.6);">hsla(9, 100%, 64%,
0.6)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.8);">hsla(9, 100%, 64%,
0.8)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 1);">hsla(9, 100%, 64%,
1)</h1>
</body>
</html>
```

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

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

**hsla(9, 100%, 64%, 0.4)**

**hsla(9, 100%, 64%, 0.6)**

**hsla(9, 100%, 64%, 0.8)**

**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.

## 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!

## 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

## 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;
}
```

# 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:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 2px solid powderblue;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

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

</body>
</html>
```

## CSS Padding

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

### Example

Use of CSS border and padding properties:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 2px solid powderblue;
  padding: 30px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

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

</body>
</html>
```

## CSS Margin

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

### Example

Use of CSS border and margin properties:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 2px solid powderblue;
  margin: 50px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

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

</body>
</html>
```

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

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

# 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

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

<h1>HTML Links</h1>

<p><a href="https://www.jetking.com/">Visit Jetking</a></p>

</body>
</html>
```

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

# 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:

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

<h2>The target Attribute</h2>

<a href="https://www.jetking.com/" target="_blank">Visit Jeetking!</a>

<p>If target="_blank", the link will open in a new browser window or
tab.</p>

</body>
</html>
```

# 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

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

<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>

</body>
</html>
```

# 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

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

<h2>Image as a Link</h2>
```

```
<p>The image below is a link. Try to click on it.</p>

<a href="https://www.jetking.com/"><img src="smiley.gif" alt="HTML
tutorial" style="width:42px;height:42px;"></a>

</body>
</html>
```

## 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

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

<h2>Link to an Email Address</h2>

<p>To create a link that opens in the user's email program (to let them
send a new email), use mailto: inside the href attribute:</p>

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

</body>
</html>
```

# 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:

```
<!DOCTYPE html>
<html>
<head>
<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>
</head>
<body>

<h2>Link Colors</h2>

<p>You can change the default colors of links</p>

<a href="html_images.asp" target="_blank">HTML Images</a>

</body>
</html>
```

# Link Buttons

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

This is a link

### Example

```
<!DOCTYPE html>
<html>
<head>
<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>
</head>
<body>

<h2>Link Button</h2>
<p>A link styled as a button:</p>
<a href="default.asp" target="_blank">This is a link</a>

</body>
</html>
```

# 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

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

<p><a href="#C4">Jump to Chapter 4</a></p>
<p><a href="#C10">Jump to Chapter 10</a></p>

<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2 id="C4">Chapter 4</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 7</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 8</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 9</h2>
<p>This chapter explains ba bla bla</p>

<h2 id="C10">Chapter 10</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 11</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 12</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 13</h2>
<p>This chapter explains ba bla bla</p>
```

```
<h2>Chapter 14</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 15</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 16</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 17</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 18</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 19</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 20</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 21</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 22</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 23</h2>
<p>This chapter explains ba bla bla</p>

</body>
</html>
```

# HTML Images

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

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

<h2>HTML Image</h2>
<img src="pic_trulli.jpg" alt="Trulli" width="500" height="333">

</body>
</html>
```

# Image Size - Width and Height

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

## Example

```
<img src="img_girl.jpg" alt="Girl in a jacket"
style="width:500px;height:600px;">
```

Alternatively, you can use the `width` and `height` attributes:

## Example

```
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
```

# Image Floating

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

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

<h2>Floating Images</h2>
<p><strong>Float the image to the right:</strong></p>

<p>
<img src="smiley.gif" alt="Smiley face"
style="float:right;width:42px;height:42px;">
A paragraph with a floating image. A paragraph with a floating image. A
paragraph with a floating image.
</p>
<p><strong>Float the image to the left:</strong></p>
<p>
<img src="smiley.gif" alt="Smiley face"
style="float:left;width:42px;height:42px;">
A paragraph with a floating image. A paragraph with a floating image. A
paragraph with a floating image.
</p>
```

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

# 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 | .ico, .cur |
| JPEG | Joint Photographic Expert Group image | .jpg, .jpeg, .jfif, .pjpeg, .pjp |
| PNG | Portable Network Graphics | .png |
| SVG | Scalable Vector Graphics | .svg |

# 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:

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

## 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

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url('img_girl.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
</style>
</head>
<body>
<h2>Background Cover</h2>
<p>Set the background-size property to "cover" and the background image
will cover the entire element, in this case the body element.</p>
</body>
</html>
```

# 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">
  <img src="img_girl.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 match.

# HTML Favicon

A favicon is a small image displayed next to the page title in the browser tab.

# How To Add a Favicon in HTML

You can use any image you like as your favicon. You can also create your own favicon on sites like https://www.favicon.cc.

Tip: A favicon is a small image, so it should be a simple image with high contrast.

A favicon image is displayed to the left of the page title in the browser tab, like this:

## Example

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
<body>
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

Now, save the "index.html" file and reload it in your browser. Your browser tab should now display your favicon image to the left of the page title.

# HTML Page Title

Every web page should have a page title to describe the meaning of the page.

---

The `<title>` element adds a title to your page:

## Example

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Tutorial</title>
</head>
<body>
The content of the document......
</body>
</html>
```

# HTML Tables

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

## Define an HTML Table

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

### Example

A simple HTML table:

```
<!DOCTYPE html>
<html>
<style>
table, th, td {
  border:1px solid black;
}
</style>
<body>

<h2>A basic HTML table</h2>

<table style="width:100%">
  <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>

<p>To understand the example better, we have added borders to the
table.</p>
```

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

# 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: A table cell can contain all sorts of HTML elements: text, images, lists, links, other tables, etc.

# Table Rows

Each table row starts with a `<tr>` and ends 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>
```

# Table Headers

Sometimes you want your cells to be table header cells. In those cases use the `<th>` tag instead of the `<td>` tag:

`th` stands for table header.

## Example

Let the first row be table header cells:

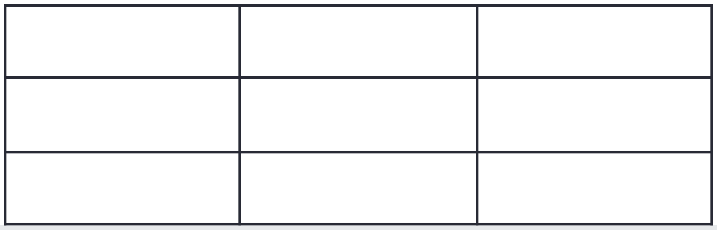
```
<!DOCTYPE html>
<html>
<style>
table, th, td {
  border:1px solid black;
}
</style>
<body>

<h2>TH elements define table headers</h2>

<table style="width:100%">
  <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>

<p>To understand the example better, we have added borders to the
table.</p>

</body>
</html>
```

# HTML Table Borders

HTML tables can have borders of different styles and shapes.

## How To Add a Border

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



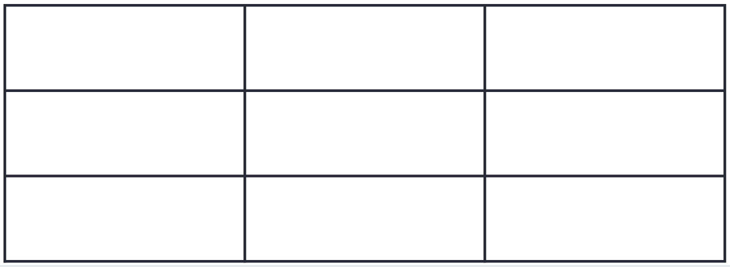
### Example

```
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:

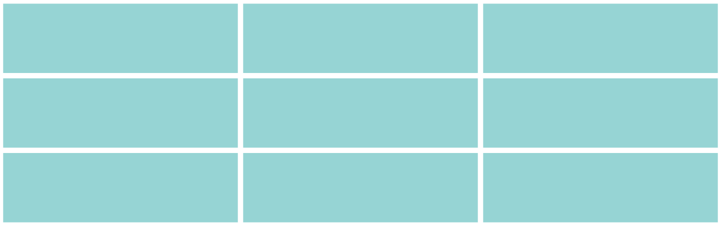


### Example

```
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

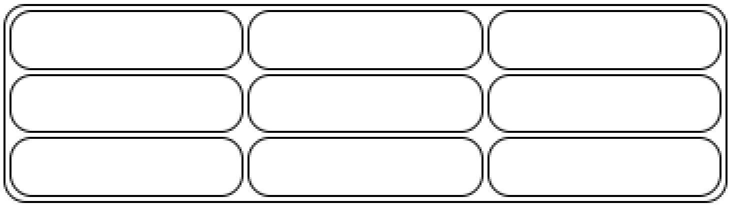
```
table, th, td {
  border: 1px solid white;
  border-collapse: collapse;
}
th, td {
  background-color: #96D4D4;
}
```

# Round Table Borders

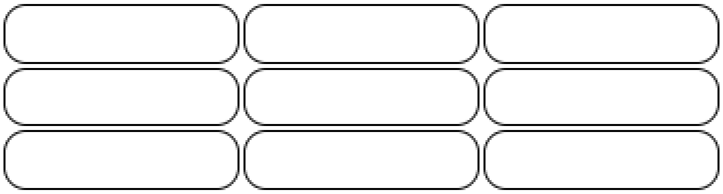
With the `border-radius` property, the borders get rounded corners:

## 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:

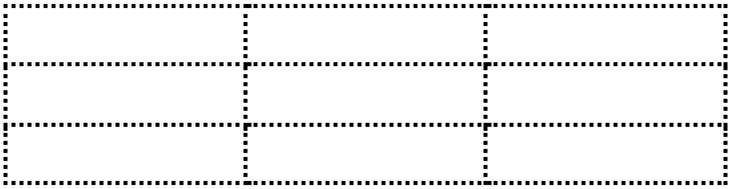


## Example

```
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

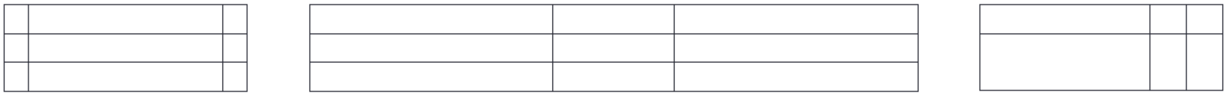
With the `border-color` property, you can set the color of the border.

### Example

```
th, td {
  border-color: #96D4D4;
}
```

# HTML Table Sizes

HTML tables can have different sizes for each column, row or the entire table.



Use the `style` attribute with the `width` or `height` properties to specify the size of a table, row or column.

---

## HTML Table Width

To set the width of a table, add the `style` attribute to the `<table>` element:

### Example

Set the width of the table to 100%:

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Note: Using a percentage as the size unit for a width means how wide will this element be compared to its parent element, which in this case is the <body> element.

# HTML Table Column Width

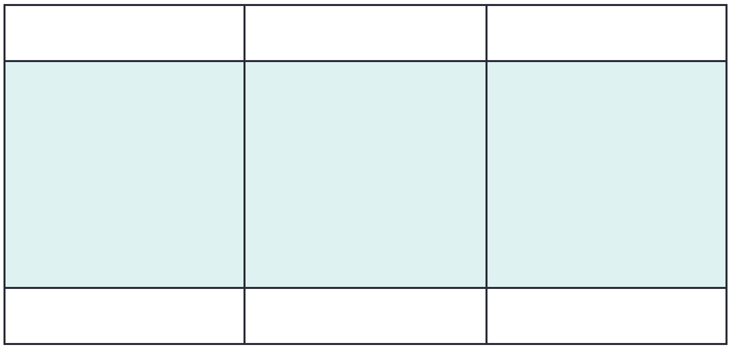
To set the size of a specific column, add the style attribute on a <th> or <td> element:

## Example

Set the width of the first column to 70%:

```
<table style="width:100%">
  <tr>
    <th style="width:70%">Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

# HTML Table Row Height



To set the height of a specific row, add the `style` attribute on a table row element:

## Example

Set the height of the second row to 200 pixels:

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr style="height:200px">
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

# HTML Table Headers

HTML tables can have headers for each column or row, or for many columns/rows.

## HTML Table Headers

Table headers are defined with `th` elements. Each `th` element represents a table cell.

### Example

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>

<h2>Table Headers</h2>

<p>Use the TH element to define table headers.</p>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>

</body>
</html>
```

## Vertical Table Headers

To use the first column as table headers, define the first cell in each row as a `<th>` element:

### Example

```
<table>
  <tr>
    <th>Firstname</th>
    <td>Jill</td>
    <td>Eve</td>
  </tr>
  <tr>
    <th>Lastname</th>
    <td>Smith</td>
    <td>Jackson</td>
  </tr>
  <tr>
    <th>Age</th>
    <td>94</td>
    <td>50</td>
  </tr>
</table>
```

## Align Table Headers

By default, table headers are bold and centered:

| Firstname | Lastname | Age |
|---|---|---|
| Jill | Smith | 50 |
| Eve | Jackson | 94 |

To left-align the table headers, use the CSS `text-align` property:

## Example

```
th {
  text-align: left;
}
```

# Header for Multiple Columns

You can have a header that spans over two or more columns.

| Name | | Age |
|---|---|---|
| Jill | Smith | 50 |
| Eve | Jackson | 94 |

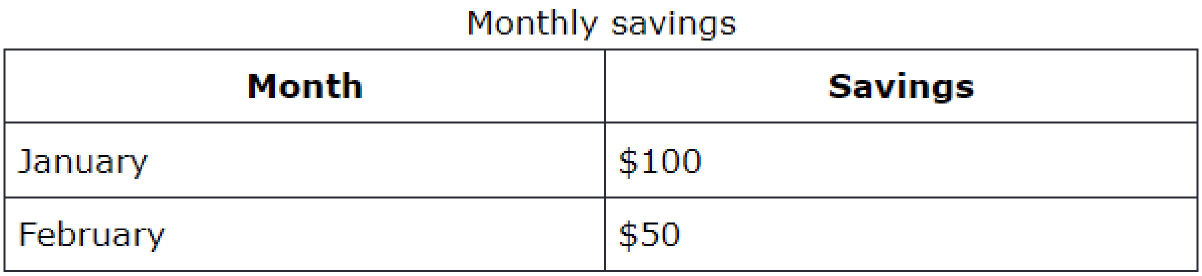
To do this, use the `colspan` attribute on the `<th>` element:

## Example

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>

<h2>A header that spans two columns</h2>

<p>Use the colspan attribute to have a header span over multiple
columns.</p>

<table style="width:100%">
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
</body>
</html>
```

# 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:

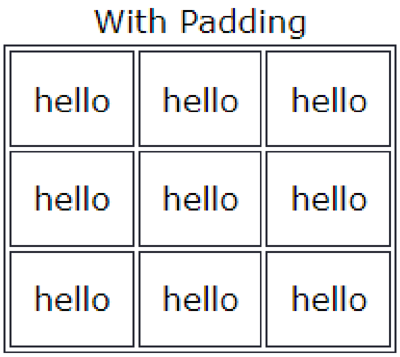
## Example

```
<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>
```

Note: The <caption> tag should be inserted immediately after the <table> tag.

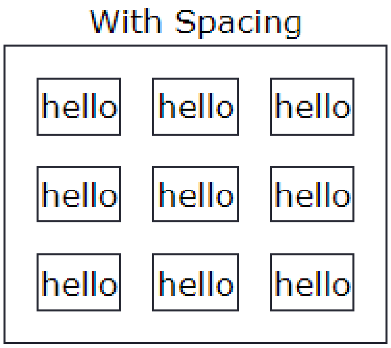
# HTML Table Padding & Spacing

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

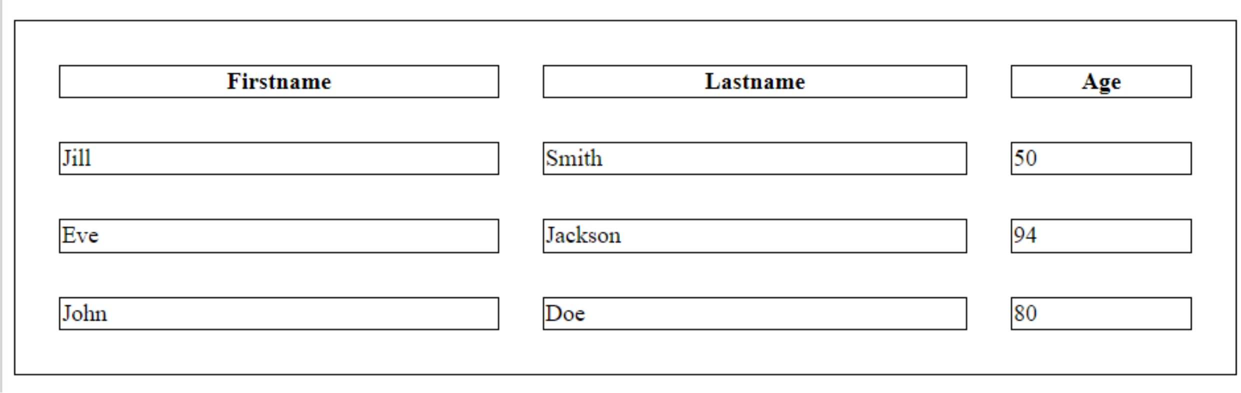
```
th, td {
  padding-top: 10px;
  padding-bottom: 20px;
  padding-left: 30px;
  padding-right: 40px;
}
```

# 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-spacing` property on the `table` element:

## Example

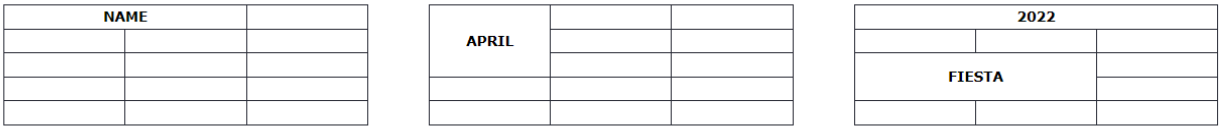
```
table {
  border-spacing: 30px;
}
```



| Firstname | Lastname | Age |
|---|---|---|
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

# HTML Table Colspan & Rowspan

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



## 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>
```

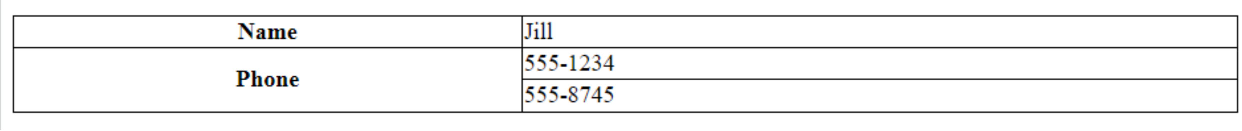
Note: The value of the `colspan` attribute represents the number of columns to span.

## 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>
```



| Name | Jill |
|---|---|
| Phone | 555-1234 |
| | 555-8745 |

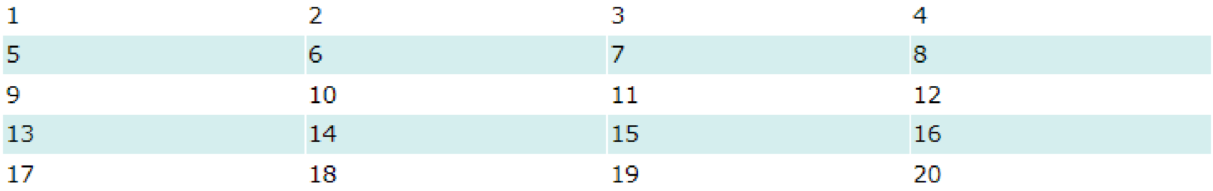
Note: The value of the `rowspan` attribute represents the number of rows to span.

# HTML Table Styling

Use CSS to make your tables look better.

## HTML Table - Zebra Stripes

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



| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 |

To style every other table row element, use the `:nth-child(even)` selector like this:

### Example

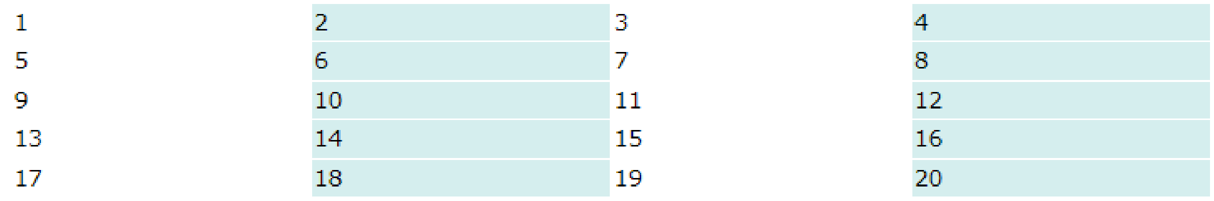
```
<!DOCTYPE html>
<html>
<head>
<style>
table {
  border-collapse: collapse;
  width: 100%;
}

th, td {
  text-align: left;
  padding: 8px;
}
```

```
tr:nth-child(even) {
  background-color: #D6EEEE;
}
</style>
</head>
<body>

<h2>Zebra Striped Table</h2>
<p>For zebra-striped tables, use the nth-child() selector and add a
background-color to all even (or odd) table rows:</p>

<table>
  <tr>
  <th>First Name</th>
  <th>Last Name</th>
  <th>Points</th>
  </tr>
  <tr>
  <td>Peter</td>
  <td>Griffin</td>
  <td>$100</td>
  </tr>
  <tr>
  <td>Lois</td>
  <td>Griffin</td>
  <td>$150</td>
  </tr>
  <tr>
  <td>Joe</td>
  <td>Swanson</td>
  <td>$300</td>
  </tr>
  <tr>
  <td>Cleveland</td>
  <td>Brown</td>
  <td>$250</td>
  </tr>
</table>

</body>
</html>
```

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*.



| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 |

## Example

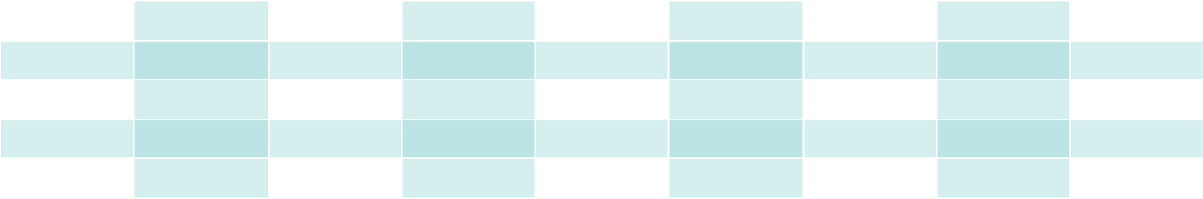
```
td:nth-child(even), th:nth-child(even) {
  background-color: #D6EEEE;
}
```

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.

# Combine Vertical and Horizontal Zebra Stripes

You can combine the styling from the two examples above and you will have stripes on every other row and every other column.
If you use a transparent color you will get an overlapping effect.

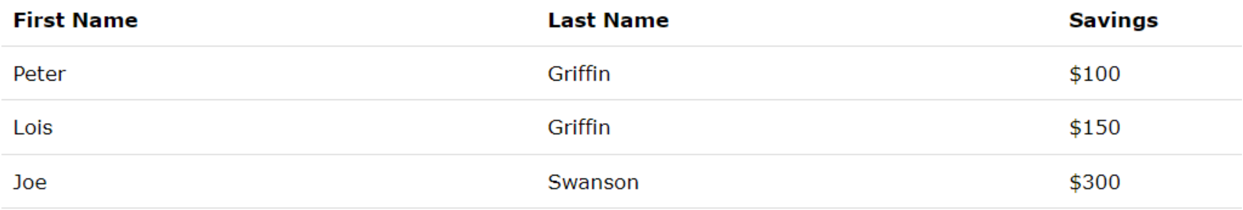
Use an `rgba()` color to specify the transparency of the color:



## Example

```
tr:nth-child(even) {
  background-color: rgba(150, 212, 212, 0.4);
}

th:nth-child(even),td:nth-child(even) {
  background-color: rgba(150, 212, 212, 0.4);
}
```

# Horizontal Dividers



| First Name | Last Name | Savings |
|---|---|---|
| Peter | Griffin | $100 |
| Lois | Griffin | $150 |
| 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:

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
table {
  border-collapse: collapse;
  width: 100%;
}

th, td {
  padding: 8px;
  text-align: left;
  border-bottom: 1px solid #DDD;
}

tr:hover {background-color: #D6EEEE;}
</style>
</head>
<body>

<h2>Hoverable Table</h2>
<p>Move the mouse over the table rows to see the effect.</p>

<table>
  <tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
  <tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
  </tr>
</table>

</body>
</html>
```

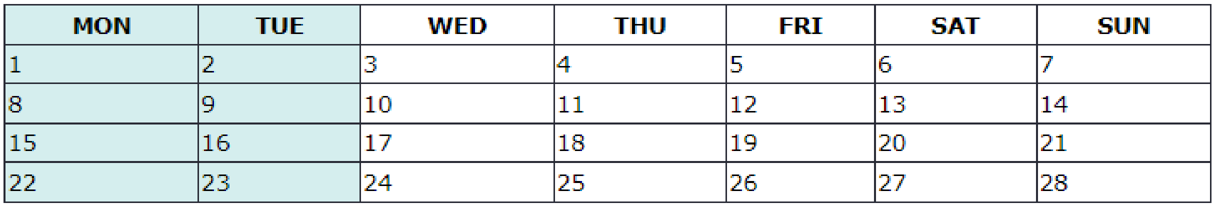
# HTML Table Colgroup

The <colgroup> element is used to style specific columns of a table.

---

## HTML Table Colgroup

If you want to style the two first columns of a table, use the <colgroup> and <col> elements.



| MON | TUE | WED | THU | FRI | SAT | SUN |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |

The <colgroup> element should be used as a container for the column specifications.
Each group is specified with a <col> element.
The span attribute specifies how many columns that get the style.
The style attribute specifies the style to give the columns.

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>
```

```
<h2>Colgroup</h2>
<p>Add the a colgroup with a col element that spans over two columns to
define a style for the two columns:</p>

<table style="width: 100%;">
<colgroup>
  <col span="2" style="background-color: #D6EEEE">
</colgroup>
<tr>
<th>MON</th>
<th>TUE</th>
<th>WED</th>
<th>THU</th>
<th>FRI</th>
<th>SAT</th>
<th>SUN</th>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
```

```
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
</table>

</body>
</html>
```

# Multiple Col Elements

If you want to style more columns with different styles, use more `<col>` elements inside the `<colgroup>`:

## Example

```
<table>
  <colgroup>
    <col span="2" style="background-color: #D6EEEE">
    <col span="3" style="background-color: pink">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
…
```

# Empty Colgroups

If you want to style columns in the middle of a table, insert a "empty" `<col>` element (with no styles) for the columns before:

## Example

```
<table>
  <colgroup>
    <col span="3">
    <col span="2" style="background-color: pink">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
…
```

## Hide Columns

You can hide columns with the `visibility: collapse` property:

### Example

```
<table>
  <colgroup>
    <col span="2">
    <col span="3" style="visibility: collapse">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
...
```

# 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>
```

## 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>
```

## HTML Description Lists

HTML also supports 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 Unordered Lists

The HTML <ul> tag defines an unordered (bulleted) list.

---

## 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>
```

## Unordered HTML List - Choose List Item Marker

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>
```

# Nested HTML Lists

Lists can be nested (list inside list):

## Example

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
    <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

Note: A list item (`<li>`) can contain a new list, and other HTML elements, like images and links, etc.

# Horizontal List with CSS

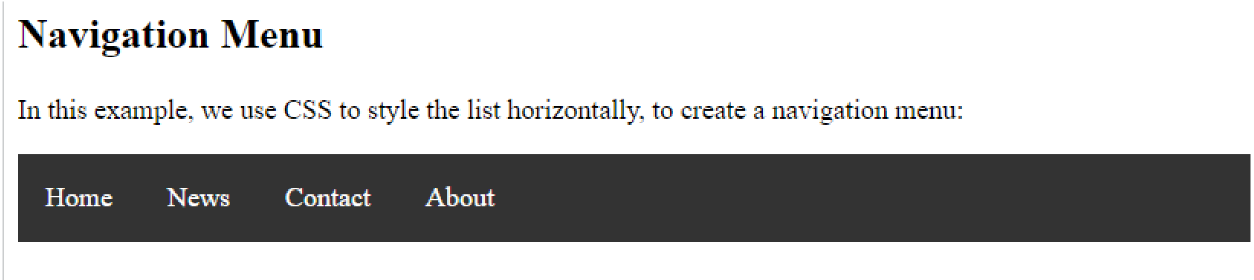
HTML lists can be styled in many different ways with CSS.

One popular way is to style a list horizontally, to create a navigation menu:

## Example

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
  text-decoration: none;
}

li a:hover {
  background-color: #111111;
}
</style>
</head>
<body>

<h2>Navigation Menu</h2>
<p>In this example, we use CSS to style the list horizontally, to
create a navigation menu:</p>

<ul>
```

```
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```



## Navigation Menu

In this example, we use CSS to style the list horizontally, to create a navigation menu:

# HTML Ordered Lists

The HTML `<ol>` tag defines an ordered list. An ordered list can be numerical or alphabetical.

## 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 |

### Numbers:

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

## Control List Counting

By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the `start` attribute:

### Example

```
<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

## Nested HTML Lists

Lists can be nested (list inside list):

### Example

```
<ol>
  <li>Coffee</li>
  <li>Tea
    <ol>
    <li>Black tea</li>
      <li>Green tea</li>
    </ol>
  </li>
  <li>Milk</li>
</ol>
```

Note: A list item (<li>) can contain a new list, and other HTML elements, like images and links, etc.

# HTML Other Lists

HTML also supports description lists.

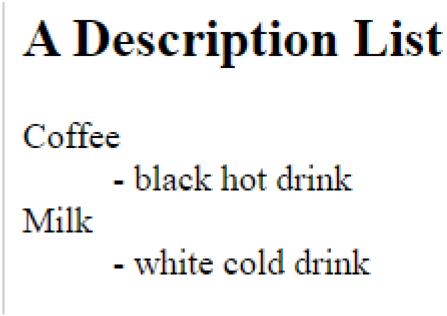
---

## 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>
```



## A Description List

Coffee

    - black hot drink

Milk

    - white cold drink

# HTML Block and Inline Elements

Every HTML element has a default display value, depending on what type of element it is.
The two most common display values are block and inline.

---

## Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.
A block-level element always takes up the full width available (stretches out to the left and right as far as it can).
Two commonly used block elements are: `<p>` and `<div>`.

The `<p>` element defines a paragraph in an HTML document.

The `<div>` element defines a division or a section in an HTML document.

The <p> element is a block-level element.

The <div> element is a block-level element.

### Example

```
<p>Hello World</p>
<div>Hello World</div>
```

Here are the block-level elements in HTML:

`<address>`

`<article>`

`<aside>`

`<blockquote>`

`<canvas>`

`<dd>`

`<div>`

`<dl>`

`<dt>`

`<fieldset>`

`<figcaption>`

`<figure>`

<footer>

<form>

<h1>-<h6>

<header>

<hr>

<li>

<main>

<nav>

<noscript>

<ol>

<p>

<pre>

<section>

<table>

<tfoot>

<ul>

<video>

# Inline Elements

An inline element does not start on a new line.
An inline element only takes up as much width as necessary.
This is a <span> element inside a paragraph.

## Example

```
<span>Hello World</span>
```

Here are the inline elements in HTML:

<a>

<abbr>

<acronym>

<b>

<bdo>

<big>

<br>

<button>

<cite>

<code>

<dfn>

<em>

<i>

<img>

<input>

<kbd>

<label>

<map>

<object>

<output>

<q>

<samp>

<script>

<select>

<small>

<span>

<strong>

<sub>

<sup>

<textarea>

<time>

<tt>

<var>

Note: An inline element cannot contain a block-level element!

# The <div> Element

The `<div>` element is often used as a container for other HTML elements.

The `<div>` element has no required attributes, but `style`, `class` and `id` are common.

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

## Example

```
<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous city
in the United Kingdom, with a metropolitan area of over 13 million
inhabitants.</p>
</div>
```

You will learn more about the `<div>` element in the next chapter.

---

# The <span> Element

The `<span>` element is an inline container used to mark up a part of a text, or a part of a document.

The `<span>` element has no required attributes, but `style`, `class` and `id` are common.

When used together with CSS, the `<span>` element can be used to style parts of the text:

## Example

```
<p>My mother has <span style="color:blue;font-weight:bold;">blue</span>
eyes and my father has <span
style="color:darkolivegreen;font-weight:bold;">dark green</span>
eyes.</p>
```

# HTML Div Element

The `<div>` element is used as a container for other HTML elements.

## The <div> Element

The `<div>` element is by default a block element, meaning that it takes all available width, and comes with line breaks before and after.

### Example

A <div> element takes up all available width:

```
Lorem Ipsum <div>I am a div</div> dolor sit amet.
```

### Result

Lorem Ipsum

I am a div

dolor sit amet.

The `<div>` element has no required attributes, but `style`, `class` and `id` are common.

## <div> as a container

The `<div>` element is often used to group sections of a web page together.

### Example

A <div> element with HTML elements:

```
<div>
  <h2>London</h2>
  <p>London is the capital city of England.</p>
  <p>London has over 13 million inhabitants.</p>
</div>
```

### Result

## London

London is the capital city of England.

London has over 13 million inhabitants.

## Center align a <div> element

If you have a <div> element that is not 100% wide, and you want to center-align it, set the CSS margin property to auto.

### Example

```
<style>
div {
  width:300px;
  margin:auto;
}
</style>
```

### Result

London

London is the capital city of England.

London has over 13 million inhabitants.

## Multiple <div> elements

You can have many <div> containers on the same page.

### Example

```
<div>
  <h2>London</h2>
  <p>London is the capital city of England.</p>
  <p>London has over 13 million inhabitants.</p>
</div>

<div>
  <h2>Oslo</h2>
  <p>Oslo is the capital city of Norway.</p>
  <p>Oslo has over 600.000 inhabitants.</p>
</div>

<div>
  <h2>Rome</h2>
  <p>Rome is the capital city of Italy.</p>
  <p>Rome has almost 3 million inhabitants.</p>
</div>
```

### Result

## London

London is the capital city of England.
London has over 13 million inhabitants.

## Oslo

Oslo is the capital city of Norway.
Oslo has over 600.000 inhabitants.

## Rome

Rome is the capital city of Italy.
Rome has almost 3 million inhabitants.

# Aligning <div> elements side by side

When building web pages, you often want to have two or more `<div>` elements side by side, like this:

## London

London is the capital city of England.
London has over 13 million inhabitants.

## Oslo

Oslo is the capital city of Norway.
Oslo has over 600.000 inhabitants.

## Rome

Rome is the capital city of Italy.
Rome has almost 3 million inhabitants.

There are different methods for aligning elements side by side, all include some CSS styling. We will look at the most common methods:

---

# Float

The CSS `float` property was not originally meant to align `<div>` elements side-by-side, but has been used for this purpose for many years.

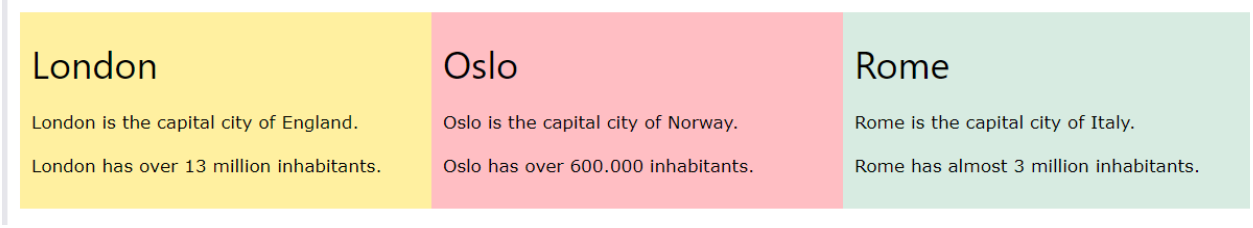
The CSS `float` property is used for positioning and formatting content and allow elements float next to each other instead of on top of each other.

## Example

How to use float to align div elements side by side:

```
<style>
.mycontainer {
  width:100%;
  overflow:auto;
}
.mycontainer div {
  width:33%;
  float:left;
}
</style>
```

## Result



London

London is the capital city of England.

London has over 13 million inhabitants.

Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

# Inline-block

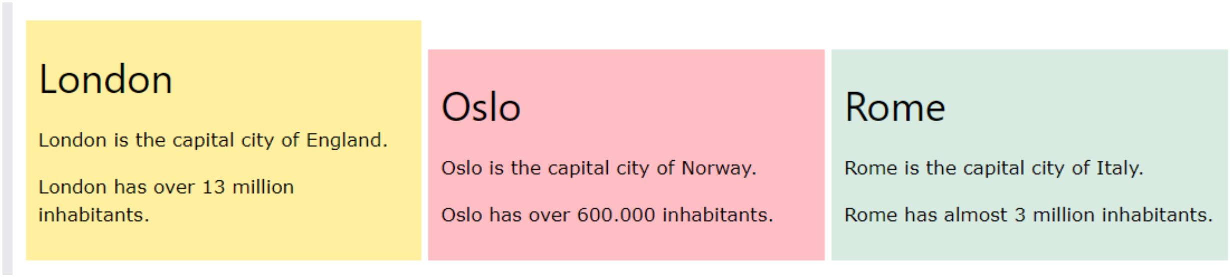
If you change the `<div>` element's `display` property from `block` to `inline-block`, the `<div>` elements will no longer add a line break before and after, and will be displayed side by side instead of on top of each other.

## Example

How to use display: inline-block to align div elements side by side:

```
<style>
div {
  width: 30%;
  display: inline-block;
}
</style>
```

## Result



London

London is the capital city of England.

London has over 13 million inhabitants.

Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

```
<!DOCTYPE html>
<html>
<style>
div {
  width:30%;
  display:inline-block;
}
</style>
<body>

<div style="background-color:#FFF4A3;">
  <h2>London</h2>
  <p>London is the capital city of England.</p>
  <p>London has over 13 million inhabitants.</p>
</div>

<div style="background-color:#FFC0C7;">
  <h2>Oslo</h2>
  <p>Oslo is the capital city of Norway.</p>
  <p>Oslo has over 600.000 inhabitants.</p>
</div>

<div style="background-color:#D9EEE1;">
  <h2>Rome</h2>
  <p>Rome is the capital city of Italy.</p>
  <p>Rome has almost 3 million inhabitants.</p>
</div>

</body>
</html>
```

# Flex

The CSS Flexbox Layout Module was introduced to make it easier to design flexible responsive layout structure without using float or positioning.
To make the CSS flex method work, surround the `<div>` elements with another `<div>` element and give it the status as a flex container.

## Example

How to use flex to align div elements side by side:

```
<style>
.mycontainer {
  display: flex;
}
.mycontainer > div {
  width:33%;
}
</style>
```

## Result

```
<!DOCTYPE html>
<html>
<head>
<style>
.mycontainer {
  display: flex;
}
.mycontainer > div {
  width:33%;
}
</style>
</head>
<body>

<h1>Flexbox Example</h1>

<p>Align three DIV elements side by side.</p>
```

```
<div class="mycontainer">

  <div style="background-color:#FFF4A3;">
    <h2>London</h2>
    <p>London is the capital city of England.</p>
    <p>London has over 13 million inhabitants.</p>
  </div>

  <div style="background-color:#FFC0C7;">
    <h2>Oslo</h2>
    <p>Oslo is the capital city of Norway.</p>
    <p>Oslo has over 600.000 inhabitants.</p>
  </div>

  <div style="background-color:#D9EEE1;">
    <h2>Rome</h2>
    <p>Rome is the capital city of Italy.</p>
    <p>Rome has almost 3 million.</p>
  </div>

</div>

</body>
</html>
```

# Grid

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

Sounds almost the same as flex, but has the ability to define more than one row and position each row individually.

The CSS grid method requires that you surround the `<div>` elements with another `<div>` element and give the status as a grid container, and you must specify the width of each column.

## Example

How to use grid to align <div> elements side by side:

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: 33% 33% 33%;
}
</style>
```

## Result

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: 33% 33% 33%;
}
</style>
</head>
<body>

<h1>Grid Example</h1>

<p>Align three DIV elements side by side.</p>

<div class="grid-container">

<div style="background-color:#FFF4A3;">
  <h2>London</h2>
  <p>London is the capital city of England.</p>
  <p>London has over 13 million inhabitants.</p>
</div>

<div style="background-color:#FFC0C7;">
  <h2>Oslo</h2>
  <p>Oslo is the capital city of Norway.</p>
  <p>Oslo has over 600.000 inhabitants.</p>
</div>

<div style="background-color:#D9EEE1;">
  <h2>Rome</h2>
  <p>Rome is the capital city of Italy.</p>
  <p>Rome has almost 3 million inhabitants.</p>
</div>

</div>

</body>
</html>
```

# 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.

## Using The class Attribute

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>
```

In the following example we have two `<span>` elements with a `class` attribute with the value of "note". Both `<span>` elements will be styled equally according to the `.note` style definition in the head section:

## Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.note {
  font-size: 120%;
  color: red;
}
</style>
</head>
<body>

<h1>My <span class="note">Important</span> Heading</h1>
<p>This is some <span class="note">important</span> text.</p>

</body>
</html>
```

Tip: The `class` attribute can be used on any HTML element.

Note: The class name is case sensitive!

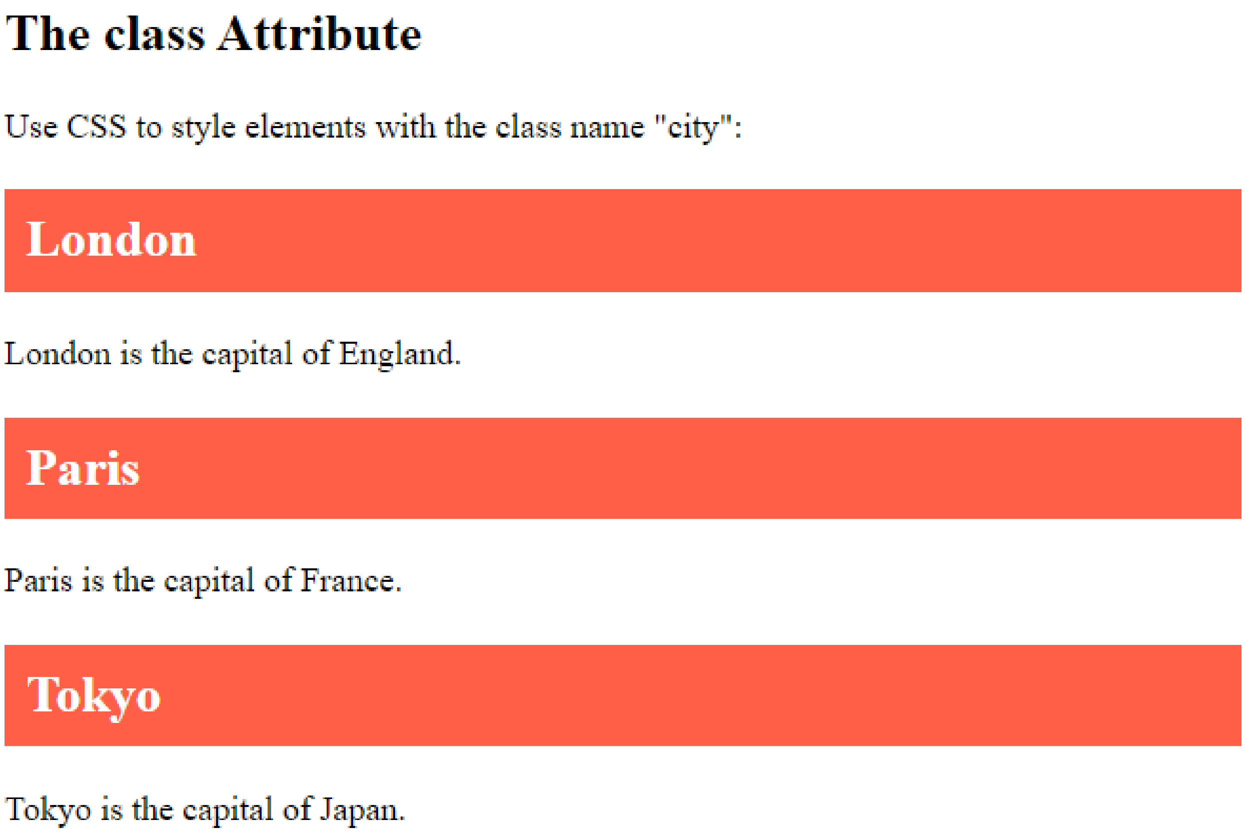
# The Syntax For Class

To create a class; write a period (.) character, followed by a class name. Then, define the CSS properties within curly braces {}:

## Example

Create a class named "city":

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<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>

</body>
</html>
```



## The class Attribute

Use CSS to style elements with the class name "city":

## London

London is the capital of England.

## Paris

Paris is the capital of France.

## Tokyo

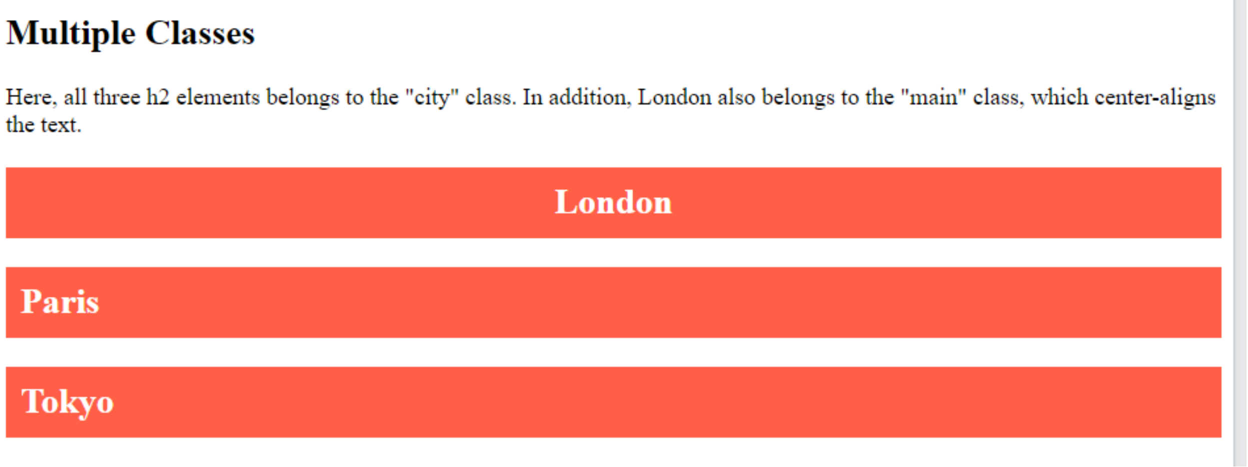
Tokyo is the capital of Japan.

# Multiple Classes

HTML elements can belong to more than one class.
To define multiple classes, separate the class names with a space, e.g. <div class="city main">. The element will be styled according to all the classes specified.
In the following example, the first `<h2>` element belongs to both the `city` class and also to the `main` class, and will get the CSS styles from both of the classes:

## Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}

.main {
  text-align: center;
}
</style>
</head>
<body>

<h2>Multiple Classes</h2>
<p>Here, all three h2 elements belongs to the "city" class. In addition, London
also belongs to the "main" class, which center-aligns the text.</p>

<h2 class="city main">London</h2>
<h2 class="city">Paris</h2>
<h2 class="city">Tokyo</h2>

</body>
</html>
```



Multiple Classes

Here, all three h2 elements belongs to the "city" class. In addition, London also belongs to the "main" class, which center-aligns the text.

London

Paris

Tokyo

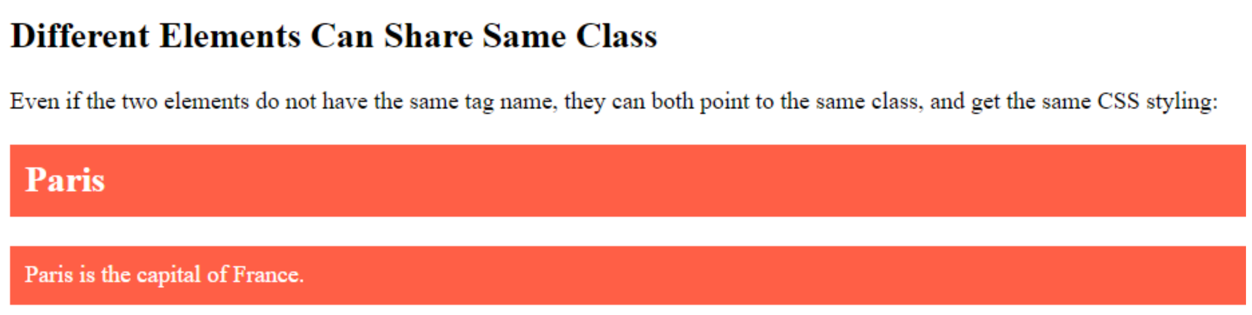
# Different Elements Can Share Same Class

Different HTML elements can point to the same class name.

In the following example, both `<h2>` and `<p>` point to the "city" class and will share the same style:

## Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<h2>Different Elements Can Share Same Class</h2>

<p>Even if the two elements do not have the same tag name, they can both
point to the same class, and get the same CSS styling:</p>

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

</body>
</html>
```



**Different Elements Can Share Same Class**

Even if the two elements do not have the same tag name, they can both point to the same class, and get the same CSS styling:

**Paris**

Paris is the capital of France.

# HTML id Attribute

The HTML `id` attribute is used to specify a unique id for an HTML element. You cannot have more than one element with the same id in an HTML document.

## Using The id Attribute

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

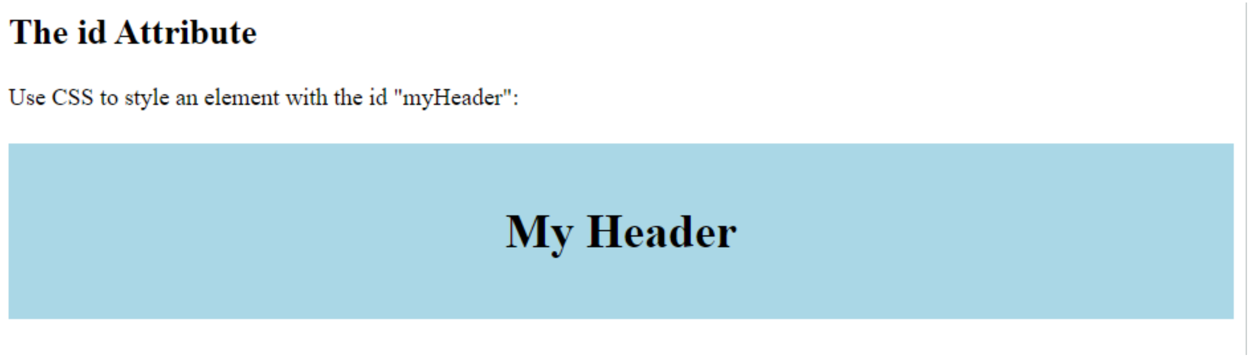
The `id` attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

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>
```



## The id Attribute

Use CSS to style an element with the id "myHeader":

# My Header

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.).

---

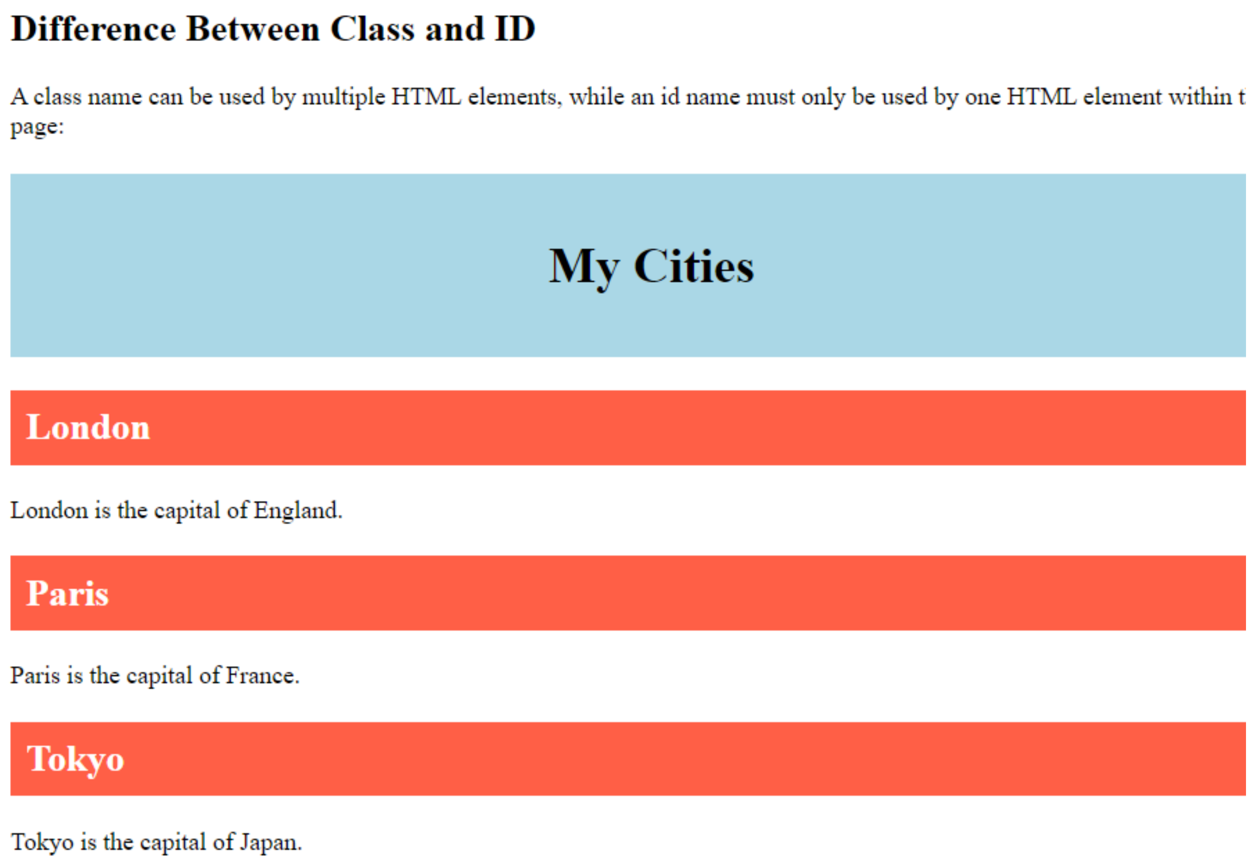
# Difference Between Class and ID

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>
```



## 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 t page:

# My Cities

## London

London is the capital of England.

## Paris

Paris is the capital of France.

## Tokyo

Tokyo is the capital of Japan.