

# HTML: HyperText Markup Language

**HTML** (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (**CSS**) or functionality/behavior (**JavaScript**).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, `<section>`, `<p>`, `<div>`, `<span>`, `<img>`, `<aside>`, `<audio>`, `<canvas>`, `<datalist>`, `<details>`, `<embed>`, `<nav>`, `<search>`, `<output>`, `<progress>`, `<video>`, `<ul>`, `<ol>`, `<li>` and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by `<` and `>`. The name of an element inside a tag is case-insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the `<title>` tag can be written as `<Title>`, `<TITLE>`, or in any other way. However, the convention and recommended practice is to write tags in lowercase.

The articles below can help you learn more about HTML.

## Beginner's tutorials

### [Your first website: Creating the content](#)

This article provides a brief tour of what HTML is and how to use it, aimed at people who are completely new to web development.

### [Structuring content with HTML](#)

Our [Learn web development](#) section's HTML module teaches all the HTML fundamentals from the ground up.

## Guides

The [HTML guides](#) help you build with HTML on the web, covering topics such as forms, CORS, content preloading, and responsive images.

### [HTML forms](#)

Forms are a very important part of the Web — these provide much of the functionality you need for interacting with websites, e.g., registering and logging in, sending feedback, buying products, and more. This module gets you started with creating the client-side/front-end parts of forms.

### [CORS enabled image](#)

The `crossorigin` attribute, in combination with an appropriate `CORS` header, allows images defined by the `<img>` element to be loaded from foreign origins and used in a `<canvas>` element as if they were being loaded from the current origin.

### CORS settings attributes

Some HTML elements that provide support for `CORS`, such as `<img>` or `<video>`, have a `crossorigin` attribute (`crossOrigin` property), which lets you configure the `CORS` requests for the element's fetched data.

### Preloading content with `rel="preload"`

The `preload` value of the `<link>` element's `rel` attribute allows you to write declarative fetch requests in your HTML `<head>`, specifying resources that your pages will need very soon after loading, which you therefore want to start preloading early in the lifecycle of a page load, before the browser's main rendering machinery kicks in. This ensures that they are made available earlier and are less likely to block the page's first render, leading to performance improvements. This article provides a basic guide to how `preload` works.

### Responsive images

In this article, we'll learn about the concept of responsive images — images that work well on devices with widely differing screen sizes, resolutions, and other such features — and look at what tools HTML provides to help implement them. This helps to improve performance across different devices.

## Reference

### HTML reference

HTML consists of **elements**, each of which may be modified by some number of **attributes**. HTML documents are connected to each other with **links**.

### HTML element reference

Browse a list of all [HTML elements](#).

### HTML attribute reference

Elements in HTML have **attributes**. These are additional values that configure the elements or adjust their behavior in various ways.

### Global attributes

Global attributes may be specified on all [HTML elements](#), *even those not specified in the standard*. This means that any non-standard elements must still permit these attributes, even though those elements make the document HTML5-noncompliant.

### Inline-level elements and block-level elements

HTML elements are usually "inline-level" or "block-level" elements. An inline-level element occupies only the space bounded by the tags that define it. A block-level element occupies the entire space of its parent element (container), thereby creating a "block box".

### HTML comments

HTML comments are used to add explanatory notes to the markup or to prevent the browser from interpreting specific parts of the document.

### Guide to media types and formats on the web

The `<audio>` and `<video>` elements allow you to play audio and video media natively within your content without the need for external software support.

### HTML content categories

HTML is comprised of several kinds of content, each of which is allowed to be used in certain contexts and is disallowed in others. Similarly, each context has a set of other content categories it can contain and elements that can or can't be used in them. This is a guide to these categories.

### Quirks mode and standards mode

Historical information on quirks mode and standards mode.

## Related topics

### Applying color to HTML elements using CSS

This article covers most of the ways you use CSS to add color to HTML content, listing what parts of HTML documents can be colored and what CSS properties to use when doing so.



Your blueprint for a better internet.