

At its heart, HTML is a fairly simple language made up of **elements**, which can be applied to pieces of text to give them different meaning in a document (Is it a paragraph? Is it a bulleted list? Is it part of a table?), structure a document into logical sections (Does it have a header? Three columns of content? A navigation menu?), and embed content such as images and videos into a page. This module will introduce the first two of these and introduce fundamental concepts and syntax you need to know to understand HTML.

Looking to become a front-end web developer?

We have put together a course that includes all the essential information you need to work towards your goal.

[Get started](#)

Prerequisites

Before starting this module, you don't need any previous HTML knowledge, but you should have at least basic familiarity with using computers and using the web passively (i.e., just looking at it and consuming content). You should have a basic work environment set up (as detailed in [Installing basic software](#)), and understand how to create and manage files (as detailed in [Dealing with files](#)). Both are parts of our [Getting started with the web](#) complete beginner's module.

Note: If you are working on a computer/tablet/other devices that doesn't let you create your own files, you can try out (most of) the code examples in an online coding program such as JSBin or Glitch.

Guides

This module contains the following articles, which will take you through all the basic theory of HTML and provide ample opportunity for you to test out some skills.

Getting started with HTML

Covers the absolute basics of HTML, to get you started — we define elements, attributes, and other important terms, and show where they fit in the language. We also show how a typical

HTML page is structured and how an HTML element is structured, and explain other important basic language features. Along the way, we'll play with some HTML to get you interested!

What's in the head? Metadata in HTML

The head of an HTML document is the part that **is not** displayed in the web browser when the page is loaded. It contains information such as the page `<title>`, links to CSS (if you want to style your HTML content with CSS), links to custom favicons, and metadata (data about the HTML, such as who wrote it, and important keywords that describe the document).

HTML text fundamentals

One of HTML's main jobs is to give text meaning (also known as semantics), so that the browser knows how to display it correctly. This article looks at how to use HTML to break up a block of text into a structure of headings and paragraphs, add emphasis/importance to words, create lists, and more.

Creating hyperlinks

Hyperlinks are really important — they are what makes the web a web. This article shows the syntax required to make a link and discusses best practices for links.

Advanced text formatting

There are many other elements in HTML for formatting text that we didn't get to in the [HTML text fundamentals](#) article. The elements here are less well-known, but still useful to know about. In this article, you'll learn about marking up quotations, description lists, computer code and other related text, subscript and superscript, contact information, and more.

Document and website structure

As well as defining individual parts of your page (such as "a paragraph" or "an image"), HTML is also used to define areas of your website (such as "the header," "the navigation menu," or "the main content column.") This article looks into how to plan a basic website structure and how to write the HTML to represent this structure.

Debugging HTML

Writing HTML is fine, but what if something goes wrong, and you can't work out where the error in the code is? This article will introduce you to some tools that can help.