

# Survey of HTML

## Relevant Links

- Semantic Code<sup>1</sup>
- devdocs<sup>2</sup>
- Learn HTML&CSS<sup>3</sup>
- MDN's introduction to HTML<sup>4</sup>
- "Can I use" site<sup>5</sup>
- Free online books<sup>6</sup>

## ACM Learning Center books

- HTML: A beginner's tutorial<sup>7</sup>.
- Beginning HTML5 and CSS3<sup>8</sup>.
- HTML and CSS Visual Quickstart<sup>9</sup>.
- HTML5 Foundations<sup>10</sup>
- HTML5 Programmer's Reference<sup>11</sup>

## Notes

An HTML document consists of nested "tags", each tag representing a certain "structure" for its contents.

For instance we will see tags for lists, tables, headings and so on.

This is important:

HTML dictates the structure of the page's content. It describes structure, semantics, not appearance. It is used to describe "what something is", not "how it should look". CSS is responsible for appearance.

This semantic information is important, as it is all that visually impaired users or tools like search engines will see.

---

<sup>1</sup><https://boagworld.com/dev/semantic-code-what-why-how/>

<sup>2</sup><http://devdocs.io/>

<sup>3</sup><http://learn.shayhowe.com/html-css/>

<sup>4</sup><https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Introduction>

<sup>5</sup><http://caniuse.com/>

<sup>6</sup><https://github.com/vhf/free-programming-books/blob/master/free-programming-books.md#html-css>

<sup>7</sup>[http://learning.acm.org/books/book\\_detail.cfm?id=2830919&type=24](http://learning.acm.org/books/book_detail.cfm?id=2830919&type=24)

<sup>8</sup>[http://learning.acm.org/books/book\\_detail.cfm?id=2414773&type=24](http://learning.acm.org/books/book_detail.cfm?id=2414773&type=24)

<sup>9</sup>[http://learning.acm.org/books/book\\_detail.cfm?id=2663416&type=safari](http://learning.acm.org/books/book_detail.cfm?id=2663416&type=safari)

<sup>10</sup>[http://learning.acm.org/books/book\\_detail.cfm?id=2825957&type=24](http://learning.acm.org/books/book_detail.cfm?id=2825957&type=24)

<sup>11</sup>[http://learning.acm.org/books/book\\_detail.cfm?id=2811310&type=24](http://learning.acm.org/books/book_detail.cfm?id=2811310&type=24)

There is entirely too much legacy information on HTML and various standards. We will be focusing on the current standard, usually termed HTML5. A number of old tags have been deprecated.

HTML5 is actually a collection of standards, some with very varying levels of adoption. The can-i-use site<sup>12</sup> can tell you more information about the state of adoption of the particular technology you want to use.

What we will cover in this class is standard and widely adopted. We start with an overview of some key features:

- An HTML document is a tree structure consisting of **nodes**. Each node is either an **element** node or a **text** node. Element nodes are identified by pairs of tags, an **open tag** like `<html ...>` and a corresponding **close tag** `</html>`. The **content** of the node is all the children elements that are found between the open and close tags.
- The opening tag may also contain a number of **attributes**. These are key-value pairs where the value is always a string. Example: `<div id="myDiv" class="bold item">`.
- There are a few elements that are self-closing: `<hr />`. These are roughly equivalent to open-close tags back-to-back.
- Tags must be well-nested, namely this is not allowed: `<tag1><tag2></tag1></tag2>`. In other words, if an open tag is contained in a node, the corresponding close tag must also be contained in that node.

Let us define some key terms:

**doctype** The first line of any HTML document needs to be its doctype. For HTML5, this is `<!DOCTYPE html>`.

**html tag** The next thing in any HTML document is an `<html>` tag. Everything else will be nested inside this tag.

**head tag** The head tag is the first thing inside the HTML document. It contains meta-data information about the document (keywords, author, description, title etc), as well as links to CSS spreadsheets and fonts.

**body tag** The body tag follows the head tag, and contains all the actual elements in the page.

**attributes** properties attached to tags. They are contained within the open tag like so `<tag attr1="..." attr2="...">`. We will see examples later.

**block tags** Some tags are meant to represent “block” elements. These tend to occupy space, as opposed to “inline” tags. Examples: `div`, `section`, `nav`, `table`.

**inline tags** These tags are meant to identify a structure within an inline text. For example, the `em` tag is used to indicate its content should be “emphasized”.

We will now describe in some detail some attributes and elements. We start with **attributes**:

---

<sup>12</sup><http://caniuse.com/>

**id** A string uniquely identifying an element. No two elements can have the same id.

**class** A string containing space-separated “classes” for the element. These classes have no meaning on their own, but they can be targetted by Javascript and CSS. For example `class="hidden cartItem"` identifies this element as being a “cartItem” and also being “hidden”. (This does not actually mean it is hidden, it means that it appears however CSS has specified “hidden” elements should appear). We might have some Javascript code that does something to all elements with a class “cartItem”, or we might have some CSS instructions that say how all elements of class “hidden” should appear.

**alt** alternative text for images and inputs.

**title** used for tooltips.

**data-\*** Used to add custom data to the element, that Javascript can use. Usually set programmatically.

**disabled** For inputs, whether they are “editable”.

**hidden** Prevents rendering of the item.

**name** Used in form elements, to target labels to them or to group radio buttons together.

**value** Used in form elements.

There are many more attributes, most more specialized to certain elements. You will encounter them as you find them.

Here are some of the most important **tags/elements** from standard HTML:

**h1, h2, h3** 6 tags for 6 different heading levels

**ol, ul, li** ordered and unordered lists, and list items

**dl, dt, dd** definition lists, terms and definitions

**p** paragraph

**form, fieldset, input** controls for creating forms

**pre, code** preformatted text and code

**blockquote, quote** quotes, block/inline

**a** anchor links (to other documents, or within page)

**em, strong** “emphasized” and strongly emphasized text

**hr** thematic break between paragraph-level elements. Typically displayed as a horizontal ruler

**img** embedded image

**div, span** generic block-level/inline elements

**b, i** to denote text that would normally be shown as bold/italicized

**pre, code** Used for “preformatted” fixed-width blocks, and inline code snippets

**select, option** for creating pull-down lists of items

**script** used for embedding scripts

**table, tr, td, th, thead, tbody** used to make tables

Here are some elements added in **HTML5**:

**audio, video** used to embed sound/video content

**canvas** one of the two ways of introducing graphics, the other being SVG<sup>13</sup>.

**datalist** used with option tags to form list of options for other input elements.

**figure** meant for self-contained content typically with a figcaption. Its position is meant to not be part of the normal flow.

**header, footer** meant to identify the content at the header/footer of the page

**main, article, section** meant to identify key sections of the document

**nav** identifies the section of the document meant for navigation within the page or to other pages

An example of an HTML document can be seen in the WebAppsTODO application page<sup>14</sup>.

In-class activity. Think of the elements that this page contains.

---

<sup>13</sup><https://developer.mozilla.org/en-US/docs/Web/SVG>

<sup>14</sup><https://github.com/skiadas/WebAppsTodo/blob/master/index.html>