Front-end web developer

Learn web development | MDN

Edgar Quiroz

July 25, 2021

Contents

1 Semantics and structure

1.1 HTML Basics

1.1.1 Elements

HTML language is made up of elements with either of the following syntax

```
<tag>
<tag> content </tag>
```

Tags give a meaning to the content. The content can either be nothing, text, or more elements. The resulting element tree defines the website structure. Elements with no content are called void.

Elements have categories that define their visual representation. Visually, they can be more simply divided into block elements, wich create new lines, and inline elements.

Attributes add information to elements, though it won't affect their HTML meaning. Those attributes can be used by other tools for styling or interactivity. They have the following syntax.

```
<tag attr1="value1" attr2="value2" boolattr> content </tag>
```

The last attributes would be boolean. They must have their own name as a value. As a shorthand, the value can be omitted.

1.1.2 File structure

All HTML files should have the same root structure.

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="utf-8">
    <title>Title</title>
  </head>
  <body>
```

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

Whitespace is ignored. It is nice for readability to indent nested elements, among other things.

- `<!DOCTYPE html>`: for backwards compatibility. It used to be a link to HTML specification.
- `<html>`: harcoded root. Defining langage will improve indexing.
- `<head>`: configuration, i.e. everytinh that is not content.
- `<meta charset>`: defines charset. Not mandatory, but it will solve common bugs.
- `<title>`: title to show in bookmarks and browser tabs.
- `<body>`: parent of all content.

1.1.3 Special characters

`<, >, ", '`,& are HTML reserved characters. As long as UTF8 encoding is used, any other character in content shouldn't cause any problem. To have these symbols as content, a special reference is needed.

Character	Reference
`<`	`&lt;`
`>`	`&gt;`
`"`	`&quot;`
`'`	`&apos;`
`&`	`&amp;`

Other than elements, there can be comments. These are merely for readability.

```
<!-- comment -->
<!--
    Anything inside will be ignored!
    -->
```

1.1.4 Metadata

In the previous section, a `<meta>` tag was used to specify the character set. This tag is used to specify the document metadata. Other than the chararcter set, it can have a `name` an `content` attributes. This is used for descriptions, authors, or other data to help classify the file.

```
<meta name="description" content="Google search description">
```

The description is actually used by search engines. Different sites use matadata for custom purposes. Facebook's Open Graph protocol allows a nicer rendering while linking the site on Facebook. Twitter has something similar.

1.1.5 External resources

The `<link>` tag allows to specify the location of external elements to be mae available to the file. It has a `rel` attribut to specify the type of resource, and a `href` to specify the location.

For example, to a specify the icon to be used in bookmarks ("favorites" icon, *favicon*), the rel value must be `icon`.

```
<link rel="icon" href="path/to/icon">
```

Another common usage is to specify stylesheets. In this case, `rel="stylesheet"`. The `href` attribute must then point to a CSS file.

Finally, the last common resource is a script, with the following tag

```
<script>
code();
</script>
<script src="script_file.js" defer></script>
```

The script tag may contain the script directly, or just point to the script file. In the later case, the `defer` attribute signals the script must be loaded last. This is to prevent the script from using things before they are loaded.

1.1.6 Text

Structure in text gives order and improves readability and indexing. This structure is also used for styling. In HTML, basic hierarchical structure is given by headings.

```
<h1> Main heading </h1>
<h2> Sub heading </h2>
<h3> Less important heading </h3>
<h4> The pattern continues </h4>
<h5> How long? </h5>
<h6> Not much </h6>
```

Text is processed as one long line. To give a nicer structure, it can be diveded into paragraphs. They usually insert a new line.

```
All these
words will be a
single line
<p> But this will be a new line </p>
```

Other than headings and text, HTML supports lists. Both orered and unordered. List elements must be inside a `<li>` (list element) tag. Ordered list use the tag `<ol>` and unordered ones use `<ul>`.

```
<ul>
  <li> some </li>
  <li> elements </li>
  <li> with </li>
  <li> no </li>
  <li> order </li>
</ul>

<ol>
  <li> now </li>
  <li> there </li>
  <li> is </li>
  <li> order </li>
</ol>
```

Linebreaks are forced with the void tag `<br>`, and horizontal lines with `<hr>`. Finally, to give emphasis to a phrase, there are the inline tags `<em>` and `<strong>`.

```
This is something you <em> should </em> know. <br>
But that is quite <strong>dangerous</strong>.
<hr>
In other new, ...
```

The `<u>`, `<i>`, `<b>` were used before CSS. In general they shouldn't be used. The exceptions are where their styles are generally accepte as semantic. For example, italized foreign words, bold keywords or underlines misspellings. When in doubt, do not use them.

1.1.7 Links

Up to now, the pages are just nicely structured text. But what makes the internet a network are its links. They allow to move and fetch content from anywhere. Naturally, they are elements.

```
<a
  href="path/to/content"
  title="Best website ever"
  target="_blank">
  Text to be used as link. Any element can be a link.
</a>
```

`href` is the URL of the content, and `title` just a description. `target` attribute indicates how the link should be opened. `"_blank"` will open it in a new tab. If the content is to be downloaded, an attribute `download` can give it a default name.

If the content needs to be opene with an external program, the browser will try to open it. For example, prefixing `mailto:` will open an email client with the rest of the link as address. Links can also point to elements. To allow this, the element must have an `id` attribute. Then its references by `"document#id"`.

```
<h2 id="mail" href="mailto:some@one.com">Mailing address</h2>
<p> Please direct all your doubts to our <a href="#mail"> mailing address
```

As a general rule, make links self explenatory yet concise even without context. Be explicit when the content is not HTML. This will make them effective for indexing or screen readers.

1.1.8 More text

There are other tags to structure text. They are more specialized and less common, but nevertheless useful. First, description lists allow you the enumerate concepts and their definitions. The list must be enclosed by `<dl>` tags. Each concept uses a `<dt>` (description term) and their definitions a `<dd>`.

```
<dl>
  <dt> Item</dt>
  <dd> Definition</dd>
  <dd> Other definition</dd>
  <dt> Other item</dt>
  <dd> Another definition</dd>
</dl>
```

There are also quotes. Both paragraphs and inline text can be quoted. The attribute `cite` isn't used in any standard way. The `<cite>` element is meat only to wrap the title of the resouce. Finally, abbreviations are used to expand acronyms (given by their `title` attribute) by hovering over them.

```
It is important to note that according to
<cite>someone</cite>.

<blockquote cite="path/to/source">
  Someone else said this
</blockquote>

But someone else said <q cite="path">something</q>
shorter.
Wich <abbr title="In My Humble Opinion">IMHO</abbr>,
is correct.
```

Code can be represented in text with `<code>`, and whitespace is preserved inside `<pre>`. Other than this, `<var>` references variables (more control over using `<code>`?), `<kbd>` is used for key codes and `<samp>` to wrap program output.

```
<pre><code>
var para = document.querySelector('p');
</code></pre>

<var>para</var> represents a paragraph element.

Select all the text with
<kbd>Ctrl</kbd>/<kbd>Cmd</kbd> + <kbd>A</kbd>.

<pre>$ <code>ping mozilla.org</code>
<samp>PING mozilla.org (63.245.215.20):<br>
56 data bytes<br>
64 bytes from 63.245.215.20: <br>
icmp_seq=0 ttl=40 time=158.233 ms</samp></pre>
```

`<time>` allows a unabiguous way to specify dates using a `date-time` attribute with a valid date format.

```
<time
datetime="2016-01-20T19:30">
7.30pm, 20 January 2016
</time>
```

Finally, there are some non-semantic tags used for styling. That is, tags that aren't used by HTML, but can be used by CSS or Javascript. `<div>` wraps blocks and `<span>` an inline element. They have a `class` attribute to make some kind of classification used by the other tools. They say nothing about the elements, so their usage should be minimize. Use only if there is no other appropriate tag.

1.1.9 Website structure

HTML provides tags to create common strucutural elements of website bodies.

- `<header>`: big strip of text on the top of the page, unchanged between different pages of the same website. Usually has a logo and a slogan.
- `<nav>`: navigation menu. Fast access to other pages in the website. Usually it is part of the header, but having it separated can help accessibility.
- `<aside>`: secondary menu. Usually for related link depending on the content of each page.
- `<main>`: wraps all the unique content of the page, once per page. Should be a child of `<body>`. This can be further divided using the following tags
  - `<article>`: extra hierarchy inside `<main>`. Should be self contained.
  - `<section>`: grups elements by functionality, like minimaps or a set of summaries.
- `<footer>`: strip of text at the bottom of each page. Doesn't change. Usually has useful but non-critical information about the website. Copyright, contact information, link to FAQ, about page.

1.2 Media Embedding

1.3 Tables

2 Styling and layout

3 Interactivity

4 Forms

5 Accesibility

6 Tooling