HTML basics

2019-05-06 22:54
1a0f14a

Contents

Background / history

- HTML = HyperText Markup Language
- HyperText means text with *links* to other text.
- HyperLinks = links
- HTML is (or at least began life as) an *instance* of a document format called SGML = Standard Generalized Markup Language.
- Another popular instance is DocBook, for writing technical manuals.
- "Markup" refers originally to the marks a human editor makes on a document.
- HTTP/HTML invented together (early 1990s) by Tim Berners-Lee. He was working at CERN and wanted a way for researchers to share results. First web browser was just called WorldWideWeb, implemented on a NeXT Computer (which is now in the Computer History Museum[1] in Mountain View, California).
- Other organizations started introducing HTTP/HTML-compatible browsers: Mosaic, Netscape (Mozilla → Firefox). Later: IE, Opera, Safari, Chrome, Edge, etc.
- Some of these competed on introducing new innovations to HTTP/HTML, leading to "browser wars." Cross-browser compatibility is still a concern, but not nearly as problematic as it was in earlier days.

Spaces in HTML

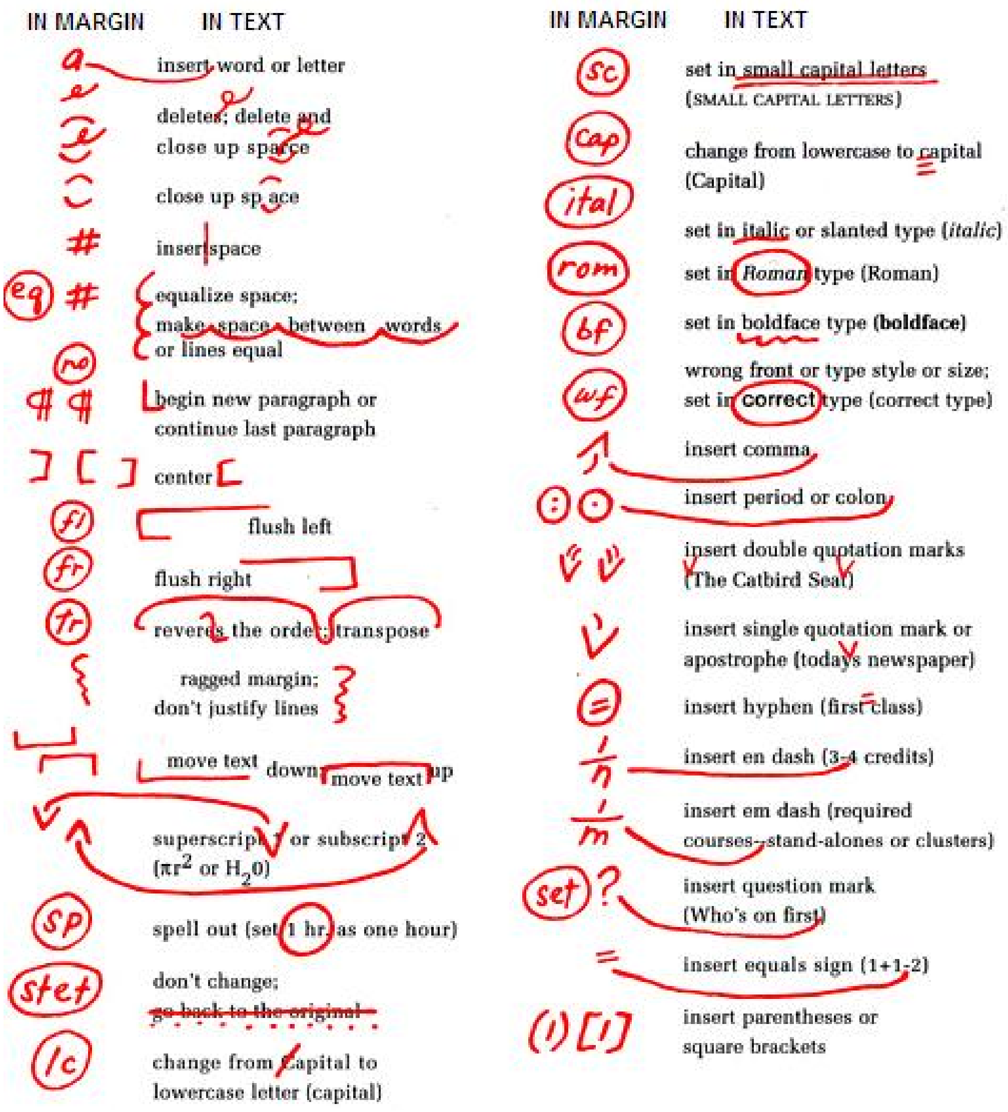
- A plain text file is also valid HTML; however rules about *spacing* are special.
- HTML does not obey more than one space. Sequences of various kinds of space (including tab, newline, etc.) are collapsed into just one space.
- Here is an example demonstrating the spacing issue. The first box is the HTML source, the second is how it looks rendered in a browser.

```
These        words           are

       really

spaced                      out!
```

[1] www.computerhistory.org/



insert word or letter

deletes; delete and close up sparce

close up sp ace

inserspace

equalize space; make space between words or lines equal

begin new paragraph or continue last paragraph

center

flush left

flush right

reverds the order; transpose

ragged margin; don't justify lines

move text down

move text up

superscript 1 or subscript 2 (πr^2 or H_20)

spell out (set 1 hr. as one hour)

don't change; ~~go back to the original~~

change from Capital to lowercase letter (capital)

set in small capital letters (SMALL CAPITAL LETTERS)

change from lowercase to capital (Capital)

set in italic or slanted type (*italic*)

set in *Roman* type (Roman)

set in boldface type (**boldface**)

wrong front or type style or size; set in correct type (correct type)

insert comma

insert period or colon

insert double quotation marks (The Catbird Seat)

insert single quotation mark or apostrophe (todays newspaper)

insert hyphen (first class)

insert en dash (3 4 credits)

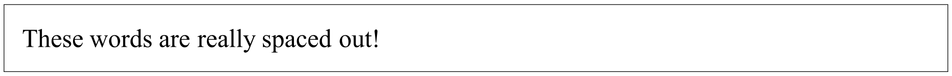
insert em dash (required courses--stand-alones or clusters)

insert question mark (Who's on first)

insert equals sign (1+1 2)

insert parentheses or square brackets

Figure 1: Proofreading and copyediting markup

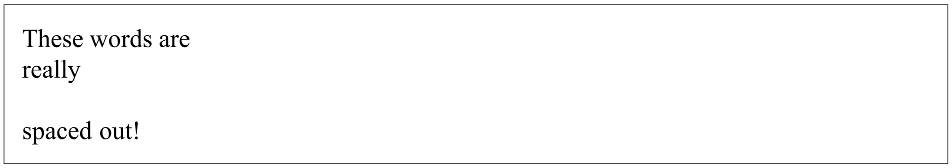


```
These words are really spaced out!
```

- So our first bit of markup will be to enforce *vertical* spacing (a line break. It is written `<br>`, with a normal ASCII less-than and greater-than sign. This syntax makes it a **tag.** Tag names like `br` are *not* case-sensitive.

- Here is a similar example, with break tags added:

```
These        words          are
<br>
       really
<br><br>
spaced                      out!
```

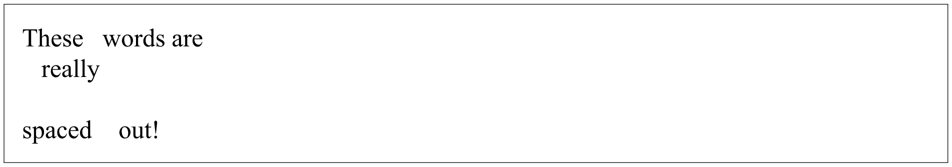


```
These words are
really

spaced out!
```

- Enforced horizontal spaces use a different syntax: `&nbsp;` where the ampersand and semi-colon are the delimiters for an **entity.** The name of the entity, `nbsp`, stands for non-breaking space. Example:

```
These &nbsp;  words           are
<br>
&nbsp;&nbsp;&nbsp;really
<br><br>
spaced   &nbsp;&nbsp;         out!
```



```
These   words are
   really

spaced    out!
```

- The main purpose of `&nbsp;` is to indicate a space that will not become a line-break when wrapping words at the right margin. For example, I might want "CS 120 Spring 2019" to appear on my page, but I don't want the numbers to wrap to the next line unless the corresponding letters go with them.

- Here's a version *without* non-breaking spaces, set at different widths so you can probably see some line-breaks appear between "CS" and "120" or "Spring" and "2019".

My favorite course to teach
is probably CS 120 in
Spring 2019.

My favorite course to teach is probably CS
120 in Spring 2019.

My favorite course to teach is probably CS 120 in Spring
2019.

My favorite course to teach is probably CS 120 in Spring 2019.

- Now here is a version *with* non-breaking spaces, set at the same widths. The browser will only break at regular spaces, not at `&nbsp;`.

```
My favorite course to teach is probably
CS&nbsp;120 in Spring&nbsp;2019.
```

My favorite course to teach
is probably CS 120 in
Spring 2019.

My favorite course to teach is probably
CS 120 in Spring 2019.

My favorite course to teach is probably CS 120 in
Spring 2019.

My favorite course to teach is probably CS 120 in Spring 2019.

Entities

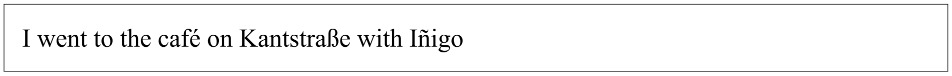
- The entity syntax `&____;` can be used to insert all kinds of special characters into the text.

- Most notably, if you need to write the characters that make up the syntax of HTML tags and entities (such as ampersand, less-than, greater-than), you should use these:

`&lt;`	<
`&gt;`	>
`&amp;`	&

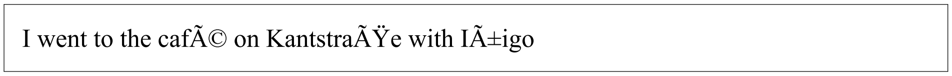
- There are also entities for most non-ASCII characters. This helps with issues of character encodings, so that it's possible to transmit the HTTP file as pure ASCII, even if it contains words like these:

```
I went to the caf&eacute; on
Kantstra&szlig;e with I&ntilde;igo
```



I went to the café on Kantstraße with Iñigo

- It's also possible just to write those characters directly in your editor, and save it using a particular text encoding, ideally UTF-8. The trouble is that if the user's browser uses a different encoding by default, the characters could get corrupted:



I went to the cafÃ© on KantstraÃŸe with IÃ±igo

- To ensure that the browser is also using UTF-8, you want to add this metadata tag near the top of your file:

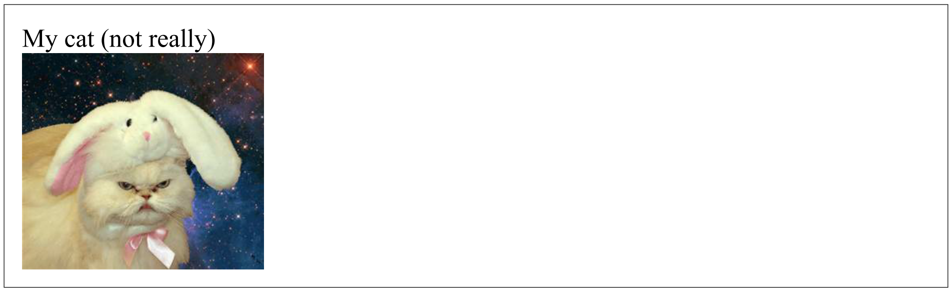
```
<meta charset="utf-8">
I went to the café on
Kantstraße with Iñigo
```

Tags with attributes

- The `meta` tag just introduced is an example of a tag with an **attribute** value. Attributes are keywords that appear *within* the angle brackets (less-than/greater-than) that make up the tag. They usually have an equals sign, and then quoted content.
- Like tag names, attribute names are case-insensitive. There can be any number of spaces or newlines around the attribute name, around the equals sign, or between attributes.
- The attribute value must be quoted if it contains spaces or other special characters. Probably it's a good idea to quote every value, but some coders omit quotes for very simple values like `width=32`.

- You can use ASCII double " or single ' quotes, as long as they match. You cannot, however, use any kind of “smart” (curly) quotes around attribute values. If your editor is inserting those for you, make it stop.
- Besides `meta`, here’s a very commonly used tag for including images:

```
My cat (not really)<br>
<img width="150" height="134"
 src="./data/17/2d394e-dbdb-4366-ba0a-64a7a6210cfb/kitty.jpg">
```

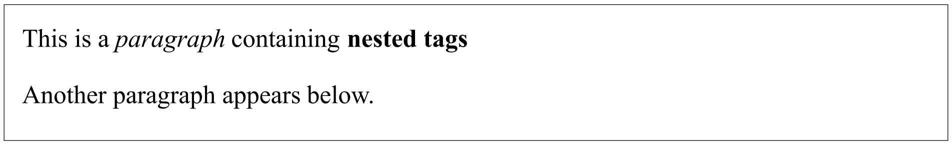


My cat (not really)

Open/close (begin/end) tags

- This is a syntax for tags that can nest and be intermingled with text. For example:

```
<p>This is a <i>paragraph</i>
  containing <b>nested tags</b>
</p>
<p>Another paragraph appears
  below.</p>
```



This is a *paragraph* containing **nested tags**

Another paragraph appears below.

- These tag names stand for paragraph (p), italic (i), and bold (b).

Recommended document structure

```
<!DOCTYPE HTML>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>My first page</title>
  </head>
  <body>
    <h1>My first page</h1>
    <p>This is a paragraph about me.</p>
  </body>
</html>
```

☑ valid HTML

- The `<title>` that appears within the `<head>` section is only used as metadata. It may appear in the browser tab or in bookmarks, but it does not appear on the page itself. That’s why we duplicate that title as an `<h1>` tag (top-level heading) in the `<body>` section.