All together now

All together now!

One key to understanding HTML, or any computer language, is to be sure that you avoid ambiguity, because computers generally are not good at judgement calls. For example, you could streamline HTML so that whenever you see a <p> tag, you start a new paragraph, no close tag needed. That might work most of the time, but that would prevent you from nesting one element inside another, as the computer could not know if you meant the new element to be nested or a successor.
A human designer might be able to tell what you meant from the context, and knowing that mistakes happen choose the one she thinks is best suited in that case. A computer, on the other hand, has difficulty with a task like that, so it is helpful to use a close tag that matches the open tag to make things absolutely clear.

```
<p>
    The old lady pulled her
spectacles
down and looked over them about the
room; then she put them up and
looked
out under them.

    There was a slight noise behind
her
and she turned just in time to seize
a
small boy by the slack of his
roundabout
and arrest his flight.
</p>
```

The old lady pulled her spectacles down and looked over them about the room; then she put them up and looked out under them. There was a slight noise behind her and she turned just in time to seize a small boy by the slack of his roundabout and arrest his flight.

A human reader could easily detect that two paragraphs were intended and that the writer probably just forgot to terminate one and start the other. The computer, on the other hand, will only see one paragraph and layout accordingly.

On the other hand, you might think that since a computer always knows exactly what tag it is working with (eidetic memory), you could provide a sort of "universal close tag" that doesn't specify the type that it's closing. It would know to close the current tag. While that's technically true, it's handy to have the close tag there for people reading the code. The close tag makes it easier to remember what tag it is closing. We humans can get confused trying to remember that kind of detail.

But there are other ambiguities to consider. For example, when a browser receives a file, it may know that it's receiving an HTML file, but it won't know which version of HTML is used (it matters).

That's why **the first thing you need in any HTML file** is a tag to tell you what type of HTML file it is:

```
1. <!DOCTYPE html>
```

In other words, the first thing the browser sees is the declaration "This is an HTML5 file, in case you were wondering". It may seem tedious to put this at the top of every file, but believe me, it used to be worse. You probably noticed that it doesn't say "!DOCTYPE HTML5" but just "html".

HTML5 can do this because all the previous versions were much more long winded. For example, at the top of an HTML 4.01 page, you might have something like this:

```
1. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
```

We do not need to go into the details of why and what that means, just be grateful that HTML5 did away with it.

Everything in HTML

It may seem redundant, but the next bit tells the computer where the actual HTML code begins, using an <html> tag:

```
1. <html>
```

Nearly every HTML document has two parts. The 'body' is the main content of the page, containing text, images, tables and so on. The 'head' comes before the 'body' (on top?). It is where you put information about the document that does not really go in the body, AKA 'meta-' information. Things like what kind of character set the page is using, where the browser can find style tips, and what the title of the page is (which might be different from the title the user reads) all go in the <head>. If you have been paying attention, you should be able to create a very basic html file, in the right form, without any content. Hint, for the head of the document you would write:

```
1. <head>
2.
3. </head>
```

You may recall the paragraph tag <p> that we used in the example above. Try inserting a paragraph into the body of your new document. You should end up with something that looks like this:

```
1. <!DOCTYPE html>
2. <html>
3.    <head>
4.    </head>
5.    <body>
6.      <p>
7.        As my English teacher used to say, 'One sentence
   does not a paragraph make'!
8.      </p>
9.    </body>
10. </html>
```