

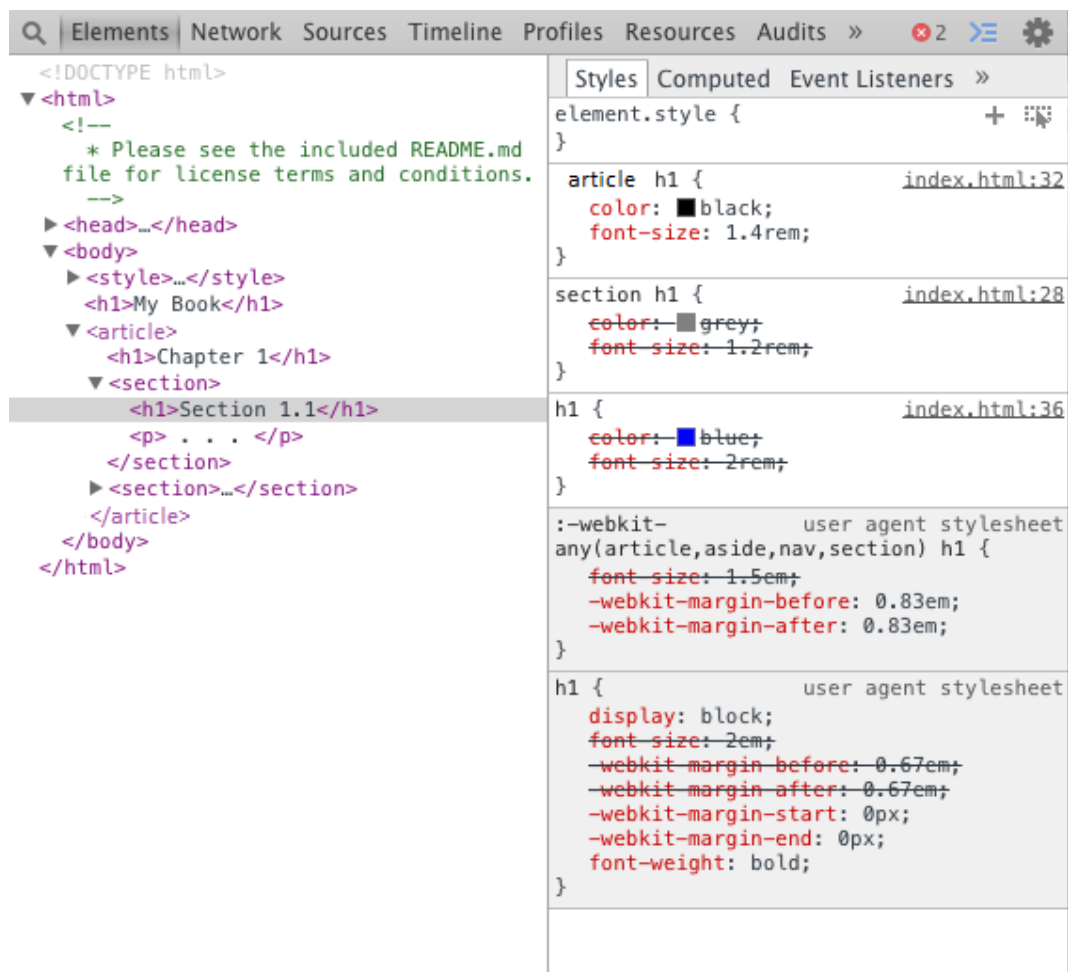


[Course](#) > [Modul...](#) > [4.4 De...](#) > [Debug...](#)

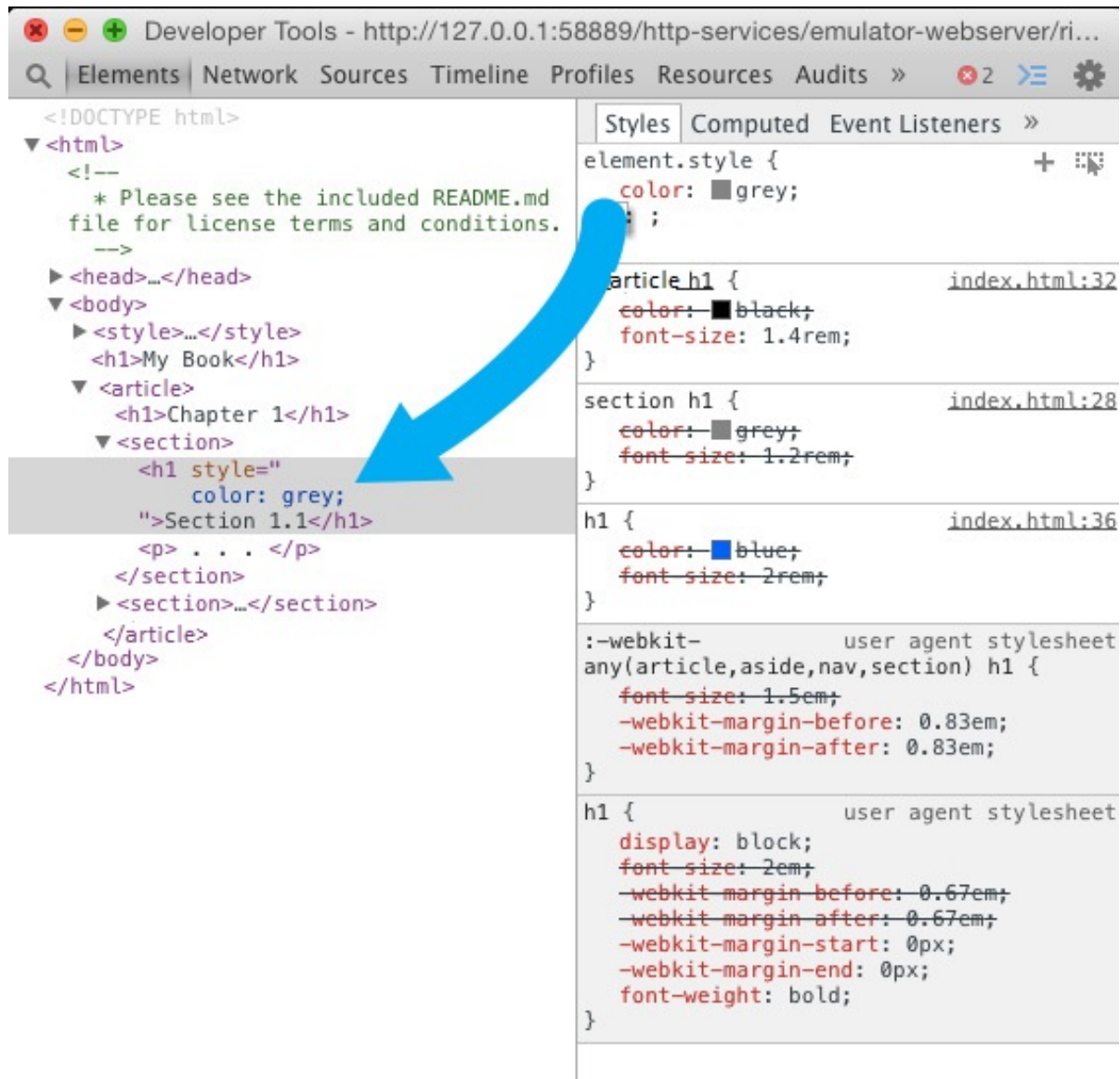
Debugging CSS precedence

Right panel of the debugger

This lesson is using developers tools provided by the Chrome browser. In the right panel of the debugger there are several tabs, such as "Styles" and "Computed". Both of these are helpful in sorting out where a particular style setting is coming from. We saw in a previous section that we can add or change CSS settings in the "Styles" panel, however, there is much more information there.



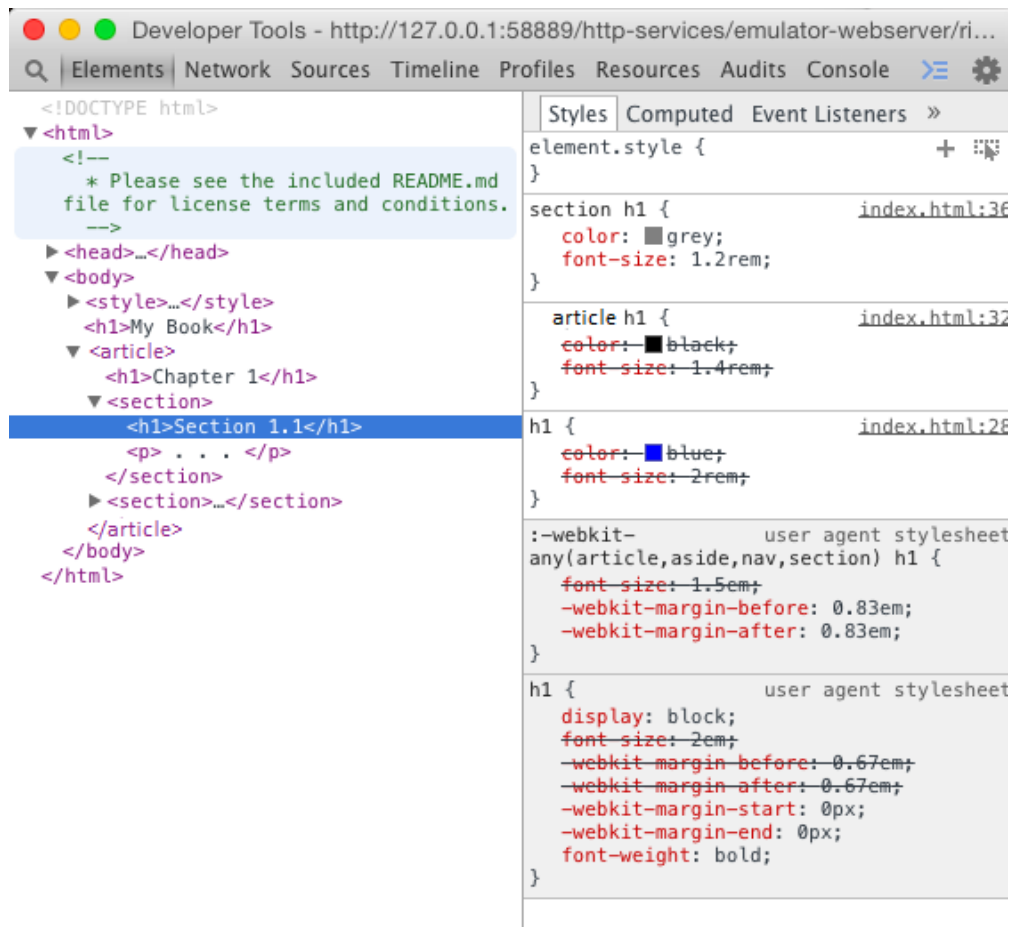
There is a sequence of the panels under "Style" that helps understand just where a particular CSS rule is coming from. Starting at the top, we have rules that apply specifically to the currently active element. In fact changes in this top panel are mirrored as settings in the style attribute of the element:



Under that there are more panels which show where CSS properties for other elements come from. Under the top panel, which corresponds to inline style settings, we find properties for this element that came from the rules with the selector "article h1". This may seem odd at first because the h1 element that we're examining is inside a section, so you'd think that the "section h1" selector would take precedence.

If we keep going down, we find the overridden rule that applies to all h1 elements, and then there are the two grayed-out section with the label "user agent stylesheet". These are basically the defaults, the values that the browser will use if nothing else is specified. Any rule you provide should override the corresponding rule in the user agent stylesheet.

Back to our quandary, why does "article h1" take precedence over "section h1"? Let's take a look at the first version we tried, before rearranging, which did what we wanted:



Here we see just the opposite of what we saw before, now "section h1" takes precedence over "article h1". What's going on?

Our intuition in this case is deceiving us. We think of section as being more specific than article, but that's just because we've organized it that way. In fact, we could decide that a section consists of multiple articles, then we'd want the opposite behavior.

As far as precedence calculations are concerned, though, the computer sees "tag-type tag-type", which is more specific than just one "tag-type", but no more specific than any other "tag-type tag-type" combination. Since "article h1" and "section h1" are of equal precedence, the tie is broken by whichever rule came last. When we rearrange the rules, we change which of the two comes last, thus causing the change in the section headers.

We could fix it by just making sure things are in the right order (which is important) but a more robust solution might be to make use of the fact that the way we're using `<section>` in fact is more specific than `<article>`. We can make this explicit by

changing the selector to "article section h1", so that now the smaller, lighter color will be used only on a section that is inside an article, which is really what we want:

HTML	CSS	Result
		EDIT ON CODEPEN
		LIVE
1 ▾ h1{		
2 color:blue;		
3 font-size: 2rem;		
4 }		
5 ▾ article h1{		
6 color: black;		
7 font-size: 1.4rem;		
8 }		
9 ▾ section h1{		
10 color:grey;		
11 font-size:1.2rem;		

My Book

Chapter 1

Section 1.1

...

Section 1.2

© All Rights Reserved