

fixed ~~TR~~

Why, then, don't we simply nest <colgroup></colgroup> elements the way we can nest lists? If we're trying to achieve recursion in structuring columns, why not set things up so that the higher-level element (<colgroup></colgroup>) recurs?

Beats me. The differences between <colgroup></colgroup> and col are pedantic at best and merely add to the confusion. But we are stuck with this confusion.

To use the span="" attribute: span="1" is implicit unless otherwise specified, and <col /> is the same as <col span="1" />. I have shied away from documenting the innumerable table formatting attributes through this chapter, but I will note that you can set the alignment of the contents of cells with cellalign and cellhalign (note the oddball spellings).

horizontal and vertical, surely?

an

Actually, in that case, now we see a way in which <colgroup></colgroup> and col differ: You can set alignment for a column group and override it individually with <col>. We could nonetheless imagine doing this with nested <colgroup></colgroup>s, but we do not live in that imaginary dreamworld.

fondly

To recast an example from above:

```

<table>
<colgroup>
<col span="1" width="35" />
<!-- Explicitly specifying a single column 35 pixels across (above) -->
</colgroup>
<colgroup span="3" halign="left">
<col />
<col cellhalign="center" />
<col />
<!-- Centre the middle column only -->
</colgroup>
<colgroup span="3">
<col span="3" />
<!-- Redundant, but legal -->
</colgroup>

```

These column groupings are quite the bother. And, at time of writing, the Big Two authoring programs, GoLive and Dreamweaver, cannot insert them automatically for you. So you're stuck typing them in yourself. An appetizing prospect, isn't it?

scope and headers

For users of adaptive technology, "associating" headers with table cells is necessary to interpret the table. A great deal of effort has gone into defining HTML structures for this purpose.

stop. ~~TR~~

gather. Quite a bit of research, including questions posed directly to Microsoft, failed to provide a definitive answer.)

Further, Microsoft has its own SMIL-like markup language for time-based media, the Synchronized Accessible Media Interchange or SAMI. Like SMIL, SAMI resembles HTML and is not particularly difficult to learn. Yet there are no authoring programs for the format; it's nonstandard; and it works only in Windows Media Player, and even then imperfectly. (A Microsoft Web page tell us simply: "Windows Media Player supports a subset of the full SAMI specification." It's Microsoft's own file format and a Microsoft player supports only "a subset" of it?)

SAMI has been rendered obsolete by SMIL and its stepchild, HTML+Time. Unless you are creating accessible media for an intranet or some other installation that you absolutely know uses Windows Media Player and you are entirely sure you will never, ever have a need to migrate accessibility files anywhere else online, you should do no authoring at all in SAMI. Use SMIL instead.

Interface issues

There's also a wee problem in user interfaces. While this is not technically your problem as a developer or designer, it is a necessary detail in understanding the practical obstacles standing in the way of accessible multimedia.

A deaf or hard-of-hearing person has no particular difficulty using the visual controls of media players. A blind or visually-impaired or mobility-impaired person definitely does have difficulty given that it is normal for players to emulate VCR-style control buttons that you have to click with a mouse. There are two classes of player: Standalone (running as its own application) and embedded in a browser. It is quite often impossible to use the Tab key to move from a surrounding Web page to the player application and within the regions of the player. Keyboard equivalents are incomplete and insufficient with all players save for Windows Media on the Windows platform per se, though all players, even Windows Media, suffer from the problem of standalone use vs. embedding in HTML pages. (How do you traverse the boundary?)

- Screen-reader users are particularly ill-served. They're already dealing with layer upon layer of abstraction:
 - The computer hardware. (With screen readers, extensive and unusual keystrokes are the norm.)
 - The computer operating system.
 - The screen reader (and/or Braille display, for that matter).
 - The application software, like a browser or player, with possibly numerous open windows.
 - The content within the software.
 - The accessibility features of that content.

only

post

with

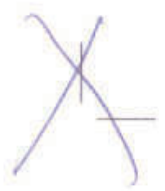
except

and

this is a paragraph, not a bullet

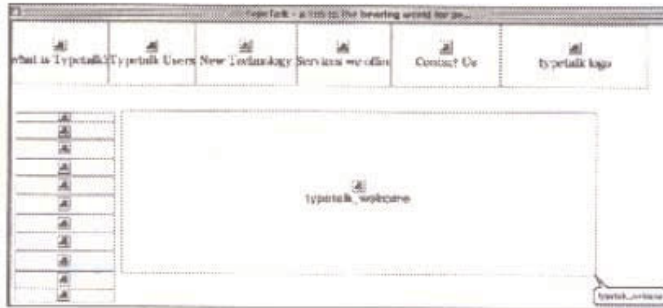
No more!

no



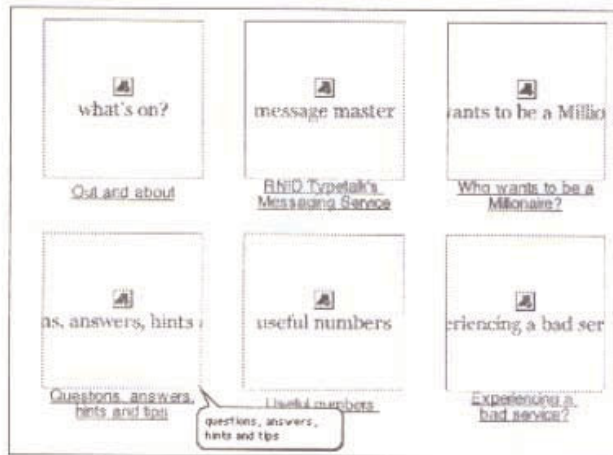


✓ (organ)



The whole point of this book is to make all Websites accessible, within certain minor limits. It is nonetheless true that any disability-related site has to offer unimpeachable accessibility. In this example, the Website of the Typetalk relay service in the U.K. (www.Typetalk.org) shows a litany of inaccessibilities. Note the unhelpful alt texts based on filenames, like typetalk_welcome. (In this example, the browser used to capture the image, Macintosh Internet Explorer, errantly pops up a tooltip containing the alt text.)

help ✓

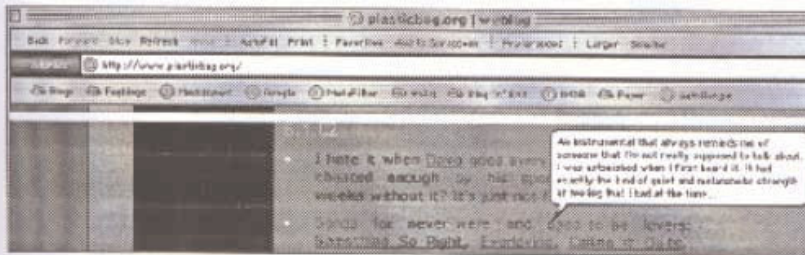


In mid-page, we find another example of images fully explained by subsequent or adjacent text; alt="" is sufficient. The fact, by the way, that a relay service is used by deaf people and not the blind is beside the point, and is blown out of the water anyway once you realize deaf-blind people use relay services, too.

Signa?

argument

by the fact that



There is no predetermined length limit to title text; even quite lengthy paragraphs can and will be displayed, as in this example. The convention I suggest is to keep to 1 K of text or less.

The next step up the ladder

!!! - you've locked to variables - #

You can add another attribute on the img element to improve accessibility: title.

Although it is not widely known, in fact the title attribute can be applied to virtually anything in HTML. Further, the purpose and function of title are quite broad (you might say vague): The actual World Wide Web Consortium definition reads "This attribute offers advisory information about the element for which it is set." In other words, anything goes.

NO!
1 wordspad
Axed



like, and keep watching for that firm's work; if you're going to emulate anybody, emulate only one company, not a mishmash of styles from this approaches captioner and that.

organization's

no

style, yes

no

level 3 heading?

[Low-budget access

After gaining all this knowledge, you may wish to try your hand. You can do captioning and audio description yourself, but do not underestimate the difficulties involved.

with

There's more than one way to publish on the Web, and sites of every description, from a budget of zero on up, can be made accessible to varying degrees. Think of the Priority 1 through 3 guidelines from the Web Accessibility Initiative, or this book's Basic, Intermediate, and Advanced accessibility advice.

is y

Out in meatspace, for example, there's a whole underground movement of home subtitling of Japanese anime videos. Fans go so far as to write their own software to do it. No one particularly cares how good or bad the subtitles are. They're better than a Japanese soundtrack you cannot understand.

no

If your Website has just a few videoclips, which themselves aren't exactly of Stanley Kubrick quality, does it particularly matter that amateurs are doing the captioning and description?

I would certainly endorse this kind of homegrown captioning and description, with reservations. The reservations have little to do with the probably low quality of the captioning (for small-time applications, that is not altogether important) and more to do with the enormous effort required. And audio description is another matter entirely.

level 2 heading?

[CAPTIONING AND TRANSCRIPTION

Captioning starts with text. Transcribing the video accurately is the first problem. You are unlikely to have experience in transcription. Neither is anyone else in your office, for that matter. Among neophytes, there is a tendency to be at once too literal in transcription ("Um, yes. Um, I think, um -") and too free ("No!!! [laughs hysterically!!] DIDN'T SAY THAT!!!").

nearby,

small caps OK here?

The best way to learn how to transcribe is to watch captioned television. Failing that, reading existing transcripts and comparing them to the original audio is a good way to learn. (And where are you going to find those?) The problem with this advice is the fact that a great many rules of written English are literally invisible to most of us: At least at the level of grammar and punctuation, well-written English doesn't call attention to itself, and the mechanisms of use are hidden. It actually takes a lot of work to make a transcript, or anything else, read effortlessly.

Handwritten signature or mark