

Re: Best solution to tedious form validation?

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2007-10/msg01973.html>

- *From:* "otrWalter@xxxxxxxx" <otrWalter@xxxxxxxx>
 - *Date:* Tue, 30 Oct 2007 08:32:13 -0700
-

Well, let's invent HTML attributes? ... use a real attribute. A classname comes to mind.

```
<input type="text" name="firstname" class="foo alpha required" value="">
```

It can more reliably be read using javascript, is valid, and is the place where you're supposed to put this kind of stuff.

Yes, I read that in the spec. I didn't agree with it then, and still don't.

My JS class was written 10 years ago, pre-dates everything about HTML validation, CLASS, etc.

The beauty of HTML (and CSS) is its ability to ignore attributes that it doesn't know about. That **is** in the specs, therefore I decided to "invent HTML attributes". It made the most logical sense (at the time). Designers (who knew HTML) found it easy to understand and code. It was easy to implement, and it failed gracefully.

Now, I don't know about CLASSNAME being "more reliably be read using Javascript" than "invented" attributes. Any attribute added to any HTML Tag is accessible via JS otherwise my method would not work.

From a human readable, non-programmer POV, which do you think is more apparent ...

```
<input type="text"  
id="first_name"  
name="first_name"  
datatype="alpha"  
required="true"  
value="">
```

.... OR ...

Re: Best solution to tedious form validation?

```
<input type="text"  
id="first_name"  
name="first_name"  
class="foo alpha required"  
value="">
```

If someone really didn't know about the new CSS "elements", they would be confused about what they mean, especially if they couldn't find those elements in the assigned CSS file. Yes, they would learn, after their initial confusion, what they mean. But this is just nit-picking.

But, on the other hand, anyone reading the first example would know something was a-miss about the tag, that these "invented" attributes are not "real" so they must have to do with something else on the client-side. Again, petty point.

Understand, just because someone is a web designer (and I know *many* who fall into this category) doesn't mean they know how to code in JS or PHP or Perl. They know enough HTML to get a page built and CSS to make it look right. My library was designed for people like that: load a few JS files, add some attributes which anyone can understand, and you now have instant client-side validation. No fuss, no muss, light weight and clean.

Besides, 100% xHTML validation can ham-string any good design. 100% xHTML validation is a honorable goal and a lofty target, but it is not realistic in the real world. It's the difference between theory and practice, between science and engineering.

Also, why is blind faith adherence to validation such a good thing? If the WC3 was all knowing, then they would have more attributes to handle just this situation, (even multi-level SELECT lists, which I begged for 15 years ago!). But, instead, they gave us a spec that allowed for a much more flexible format. Add an attribute if you like, but we (WC3) will ignore it, use it for your own purposes, don't expect us (WC3) to do anything with it.

So, yes, sir, I'm breaking validation. Yes, sir, I am among the faithful, but not a blind follower. And Yes sir! I have a highly flexible, readable, and *very* understandable (to anyone doing HTML) client-side validation system. Which was my goal, my point, my target.

I can live with my web pages being 99% xHTML validated as long I can get things done in a clean manner. And besides, to my (weird) logical mind, adding elements to CLASS that have nothing to do with CSS is "just not logical". But adding non-validatable (is that a word?) attributes to a tag in order to "extend the language/features" does make sense. Afterall, that is what I have, additional attributes to the tag defining the data type it should accept. Too bad that was (is) not in the xHTML spec.

Re: Best solution to tedious form validation?

Re: Best solution to tedious form validation?

I can only hope that the new spec with xFORMS will allow such definitions. If not, I'll (slightly) modify my library to process xFORMS, add my attributes to the FORM Objects and get on going.

Thanks for your point. I'm sure many will/do follow it. It just smacks of "after-thought" to me.

walter

.