CaCourses.com
Computer-Based Multimedia and ICT Courses

Validation

Now that we have started to write code, it is very important to make sure that the code we are writing is correct. This means that as you write code, you are observing the 6 rles of XHTML 1.0. For this purpose, the W3C has given us the XHTML Validator.

How to Self-Validate

What I mean by self-validate is using the W3C's HTML code validator what they have done at the W3C is set up an online validation service where you can upload your page and check it against the rules of your DTD. Sounds too good to be true right? It gets even better! If there is something wrong with your page, it will not only give you an error message, but it will tell you what line it's on, what character it is, and what rule you have broken. Before this, Web designers would spend hours, sometimes days, trying to figure out where they went wrong.

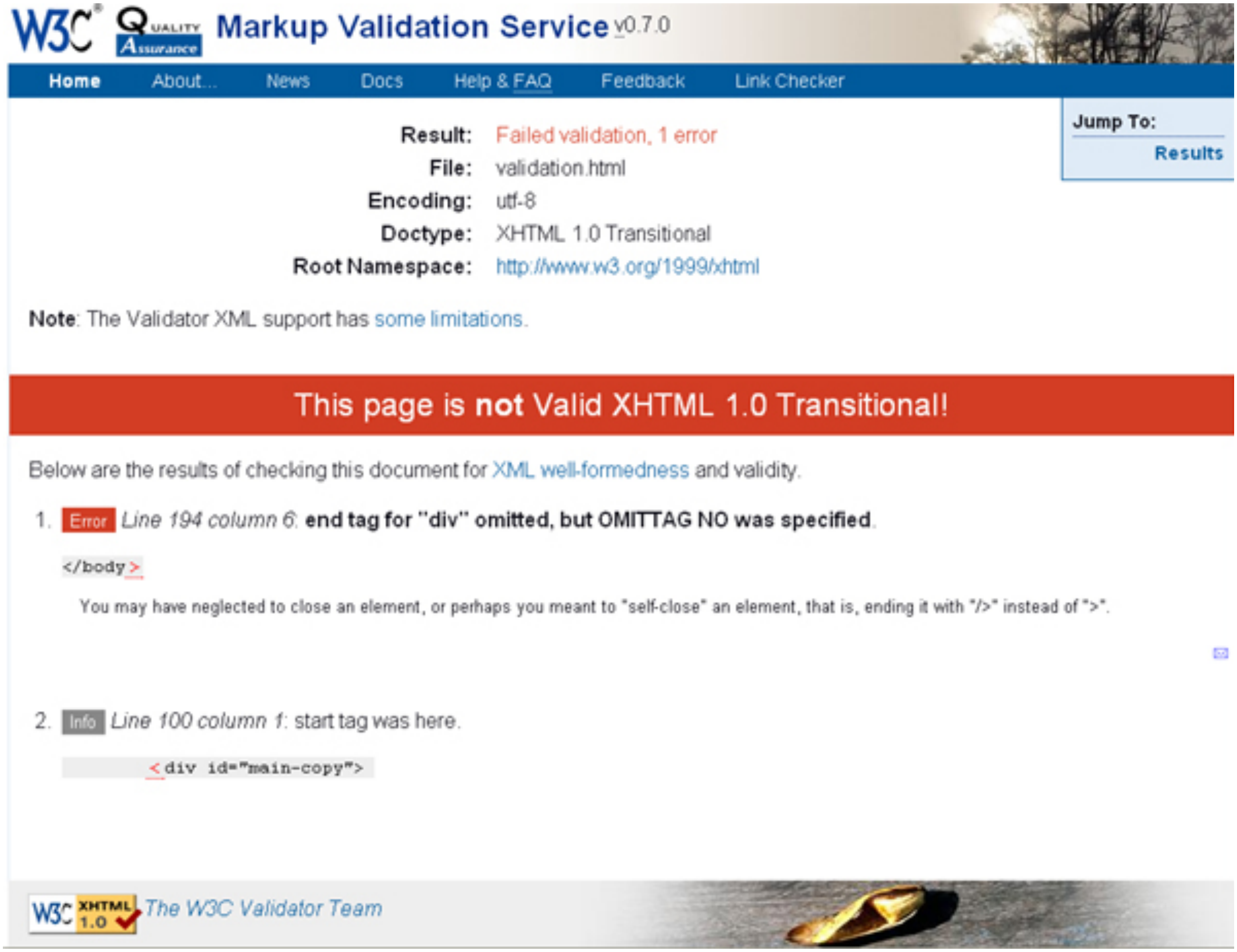
Let's go try it out:

http://validator.w3.org/

You should add this to your bookmarks as we will be using this almost every class from now on.

Understanding The Validator

Even though the W3C has tried to make this easy for us, it still takes some time to find our mistakes. If you page has errors, the validator will let you know with a big red message. A sample can be seen on the next page.



Result: Failed validation, 1 error
File: validation.html
Encoding: utf-8
Doctype: XHTML 1.0 Transitional
Root Namespace: http://www.w3.org/1999/xhtml

Note: The Validator XML support has some limitations.

This page is **not** Valid XHTML 1.0 Transitional!

Below are the results of checking this document for XML well-formedness and validity.

1. Error *Line 194 column 6*: **end tag for "div" omitted, but OMITTAG NO was specified**.

```
</body>
```

You may have neglected to close an element, or perhaps you meant to "self-close" an element, that is, ending it with "/>" instead of ">".

2. Info *Line 100 column 1*: start tag was here.

```
        <div id="main-copy">
```

The Validator Is A Machine!

Because the validator is a machine, it does not use logic to explain the problem to you in simple terms. Instead, it returns an error when your code does not do what is expected. The validator explains where it has found a problem and gives you the line number and character number of where to find the error.

The ironic thing in the above case is that the problem is not with the </body> tag. Sometimes the validator is expecting to see something and when it see something else it will return that something else as an error.

This is why it is important to read the bolded error message and not just look at the tag. The error message above says that the end tag for "div" omitted which means you have not closed your div tag.

This causes an error when the validator gets to the end of the document and was expecting to see the ending div tag but didn't. This is why the error comes back with the last tag in the document.

Just remember, that the problem will usually be ***somewhere around*** the tag shown by the validator.

Cascading Errors

One other thing to beware of, is that errors are usually cascading. This means that one error can generate another error, which can lead to many others. What this means is that you may have forgotten to close a tag, and the validator finds 30 errors.

My advice is fix the first error, and then re-validate. You'll notice that there will be significantly less than the first time just by fixing one error. Keep at this until there are no errors.

Finally, if you are ever in doubt about where your errors could be coming from, go back to the six rules of XHTML:

1) DTD
2) Nesting
3) Tag Case
4) Attribute Values
5) Quotes
6) Termination