

HTML

HyperText Markup Language

HTML TEMPLATE

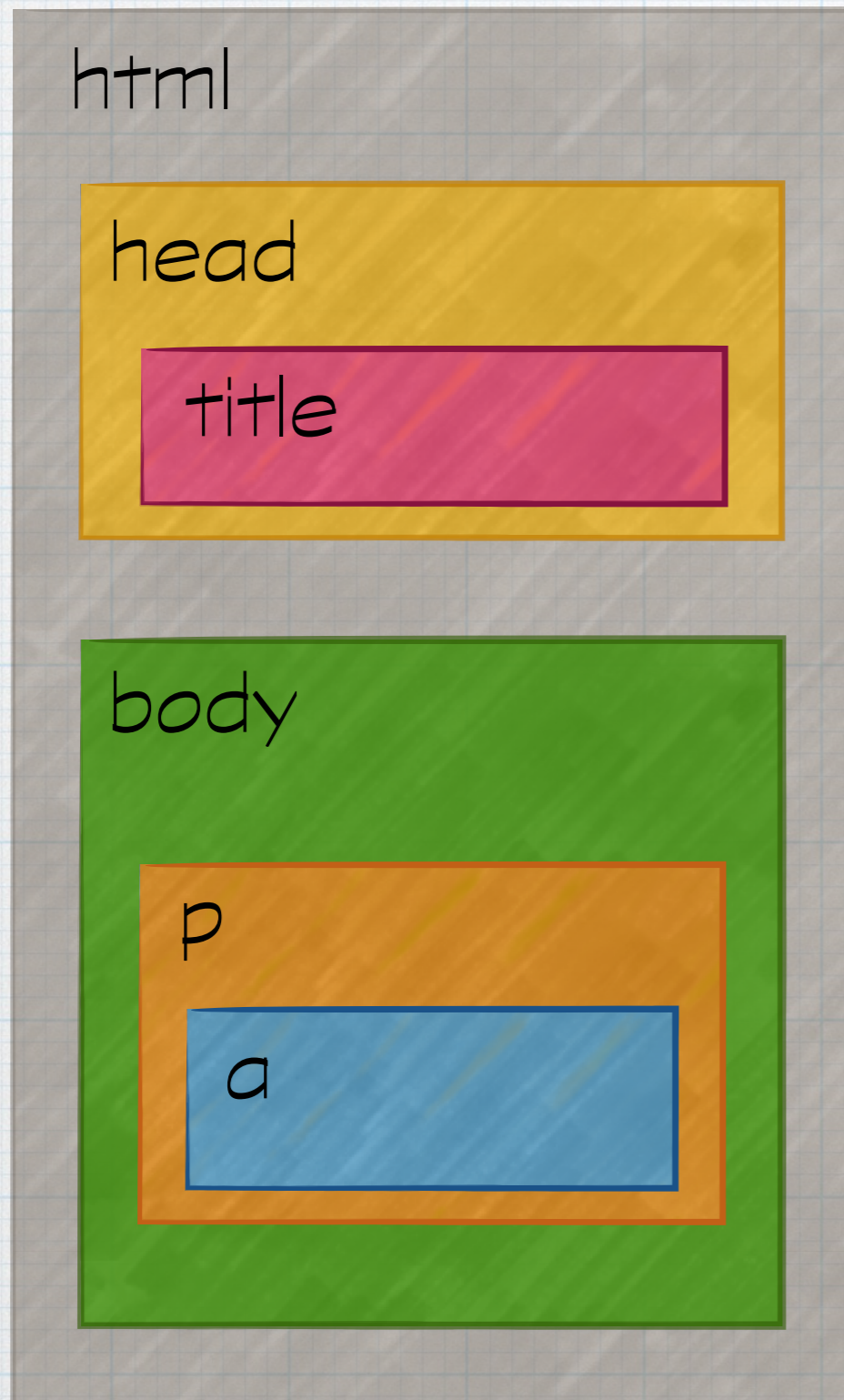
```
<!DOCTYPE html>
<html lang=en>

<head>
  <meta charset=utf-8>
  <title>My Webpage</title>
</head>

<body>
  <p>I'm at the <a href=http://www.ualberta.ca>
  University of Alberta</a>.</p>
</body>

</html>
```

HTML STRUCTURE



HTML structure is a series of nested boxes or containers. The two major containers are `<head>` and `<body>`.

The names of containers are called “tags.”

`<head>` contains metadata for the page, such as the page’s `<title>` (and other things we’ll see as we go along).

`<body>` contains all the visible content of the page, including text paragraphs (`<p>`), anchor links (`<a>`), graphics (``), etc.

HTML TEMPLATE

```
<!DOCTYPE html>  
<html lang=en>  
  
<head>  
  <meta charset=utf-8>  
  <title>My Webpage</title>  
</head>  
  
<body>  
  <p>I'm at the <a href=http://www.ualberta.ca>  
  University of Alberta</a>.</p>  
</body>  
  
</html>
```

This is the basic HTML structure we saw on the last page.

HTML TEMPLATE

```
<!DOCTYPE html>
<html lang=en>

<head>
  <meta charset=utf-8>
  <title>My Webpage</title>
</head>

<body>
  <p>I'm at the <a href=http://www.ualberta.ca>
  University of Alberta</a>.</p>
</body>

</html>
```

<body> contains everything that is visible on the page, including paragraphs, images, anchor links, etc. Here, it contains one paragraph and one anchor.

HTML TEMPLATE

```
<!DOCTYPE html>
<html lang=en>

<head>
  <meta charset=utf-8>
  <title>My Webpage</title>
</head>

<body>
  <p>I'm at the <a href=http://www.ualberta.ca>
  University of Alberta</a>.</p>
</body>

</html>
```

`<head>` contains metadata about the page. Here, that's a meta tag and the page's title. Later, we'll put other things in here too.

HTML TEMPLATE

```
<!DOCTYPE html>
<html lang=en>

<head>
  <meta charset=utf-8>
  <title>My Webpage</title>
</head>

<body>
  <p>I'm at the <a href=http://www.ualberta.ca>
  University of Alberta</a>.</p>
</body>

</html>
```

These are **attributes**. Attributes function like “adjectives” to the “nouns” that are the tags. The “grammar” of HTML defines which attributes are allowed on which tags.

HTML TEMPLATE

```
<!DOCTYPE html>  
<html lang=en>  
  
<head>  
  <meta charset=utf-8>  
  <title>My Webpage</title>  
</head>  
  
<body>  
  <p>I'm at the <a href=http://www.ualberta.ca>  
  University of Alberta</a>.</p>  
</body>  
  
</html>
```

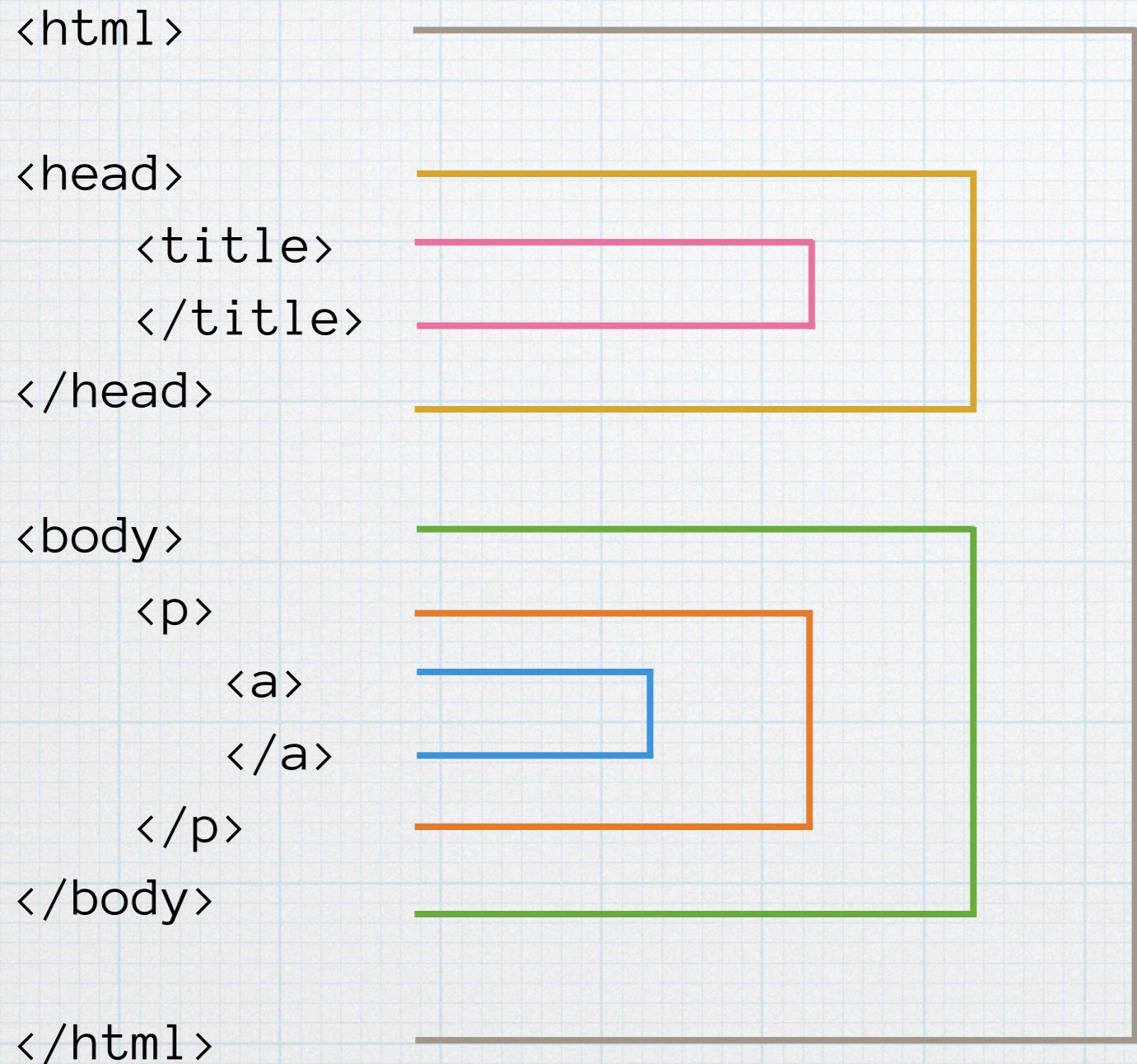
Notice that the attributes are separated from the tag name by a space, and that the attribute takes the form of **attribute=value** with no spaces.

TWO TAG TYPES

Markup languages have two types of tags: 1) the traditional “paired” tags and 2) so-called “empty” tags, which either define a property or else are “replaced” by the browser.

paired	empty
<code><html> </html></code>	<code><meta charset=utf-8></code>
<code><body> </body></code>	<code><hr></code>
<code><p> </p></code>	<code>
</code>
<code><a> </code>	<code></code>

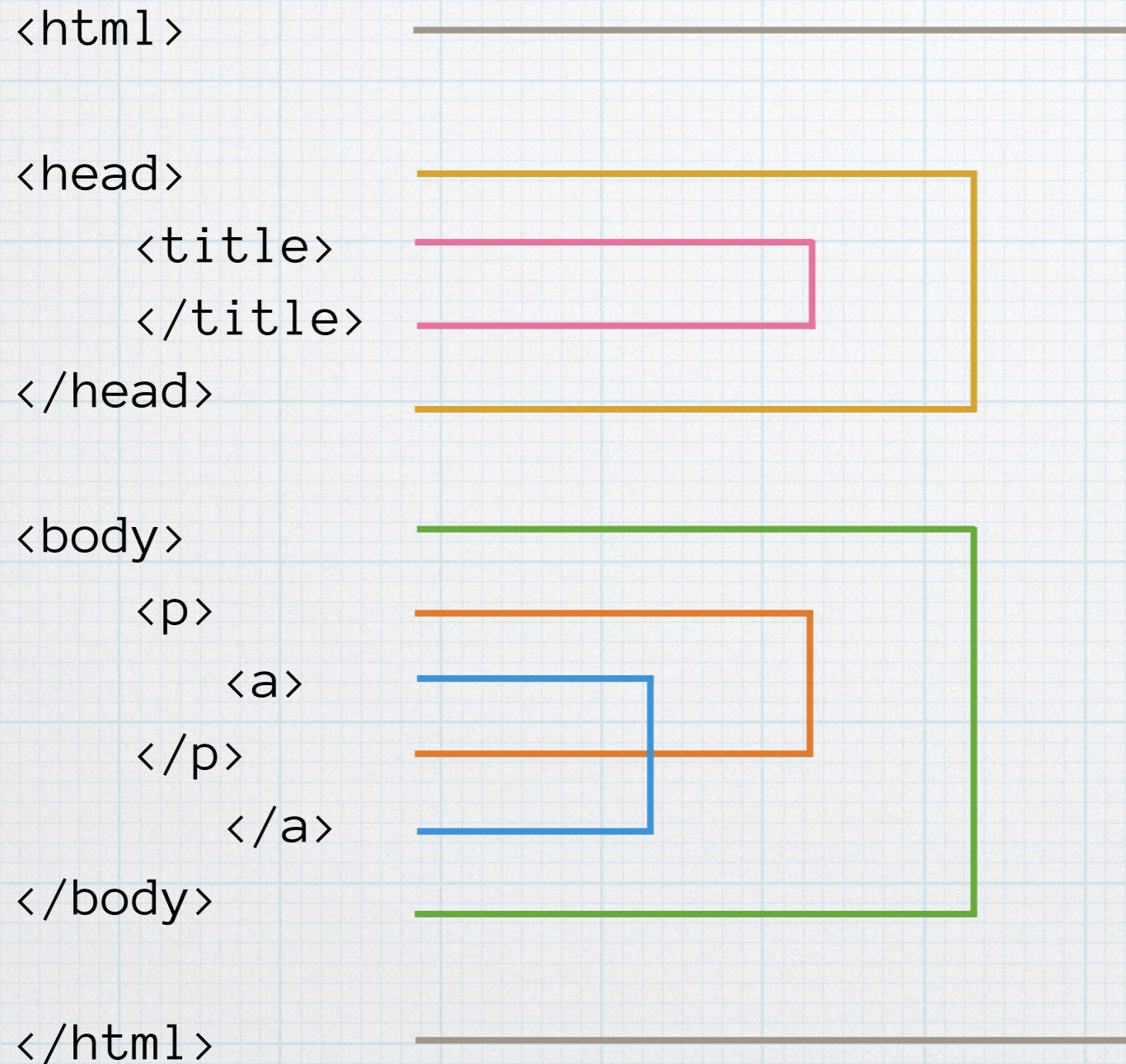
HTML TEMPLATE



Notice the structure: the tags must be closed in the reverse order that they were opened. Last opened, first closed.

This is called *well-formed* HTML. There's no guarantee that the tags and attributes are all legal, but being well-formed is one important part of writing proper HTML.

MALFORMED HTML

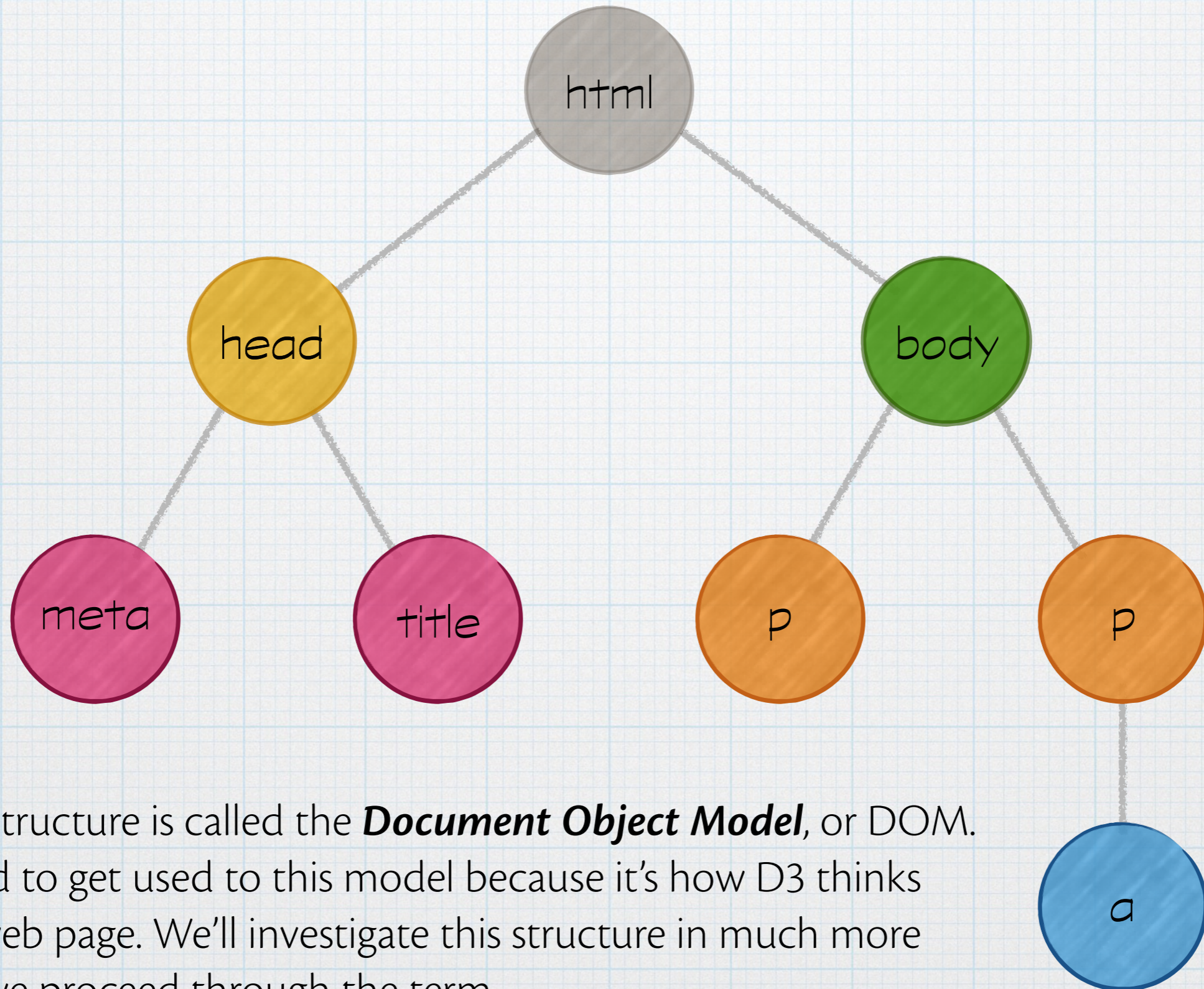


If we reverse the order of tags, we no longer have well-formed HTML.

There are no rules about how browsers are supposed to handle malformed HTML. Every browser can handle this however it wants.

Therefore, to ensure users see what we want them to see, we should write well-formed HTML.

HTML AS A TREE



This tree structure is called the **Document Object Model**, or DOM. We'll need to get used to this model because it's how D3 thinks about a web page. We'll investigate this structure in much more detail as we proceed through the term.

HTML AS A LANGUAGE

Languages have two parts:

1) a set of **words**: a vocabulary or a lexicon

form p em
ul
html head li input
title href span
body strong
div svg
id meta select
section
footer class table
article

These are some of the legal words allowed in HTML. We can't make up our own words. Even if we did, the browser wouldn't understand them anyway.

These words and their meanings have been defined by the World Wide Web Consortium (W3C).

Tags are like nouns. Attributes are like adjectives.

HTML AS A LANGUAGE

Languages have two parts:

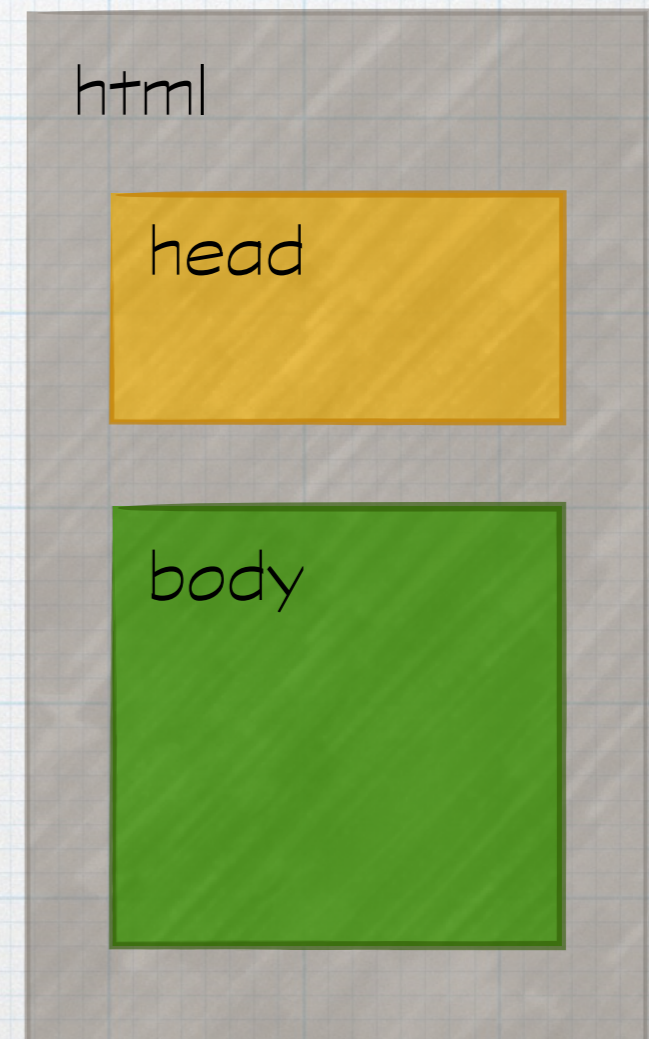
- 1) a set of **words**: a vocabulary or a lexicon
- 2) a **grammar**, or some rules, that govern how to construct meaningful expressions.

Some grammatical rules for HTML include the idea that an `<html>` container is made up of a `<head>` and a `<body>`.

Or that only `<a>` tags can take `href` attributes.

Or that we can't put a `<p>` tag inside `<head>`. Only `<body>` can contain `<p>`.

Or that any HTML element can take an `id` or a `class` attribute (these are two of what we call "global attributes," by the way).



GRAMMATICAL RULES

The legal vocabulary and all the grammatical rules are defined in a document that is called either a "Document Type Definition" (a DTD) or else a "Schema." Although there are technical differences between the two, those terms are often used interchangeably.

In this class, we won't go so far as to write our own DTDs or schemas.

```
<!DOCTYPE html>  
<html lang=en>
```

```
<head>  
</head>
```

```
<body>  
</body>
```

```
</html>
```

The **<!DOCTYPE>** tag at the top of a document identifies the particular DTD whose vocabulary and grammatical rules will govern this document.

This one says to use a DTD called **html** and that is all we need to specify to any validation service that this document is written in HTML version 5.

OLD DOCTYPES

Older doctype definitions were very complex, but you'll still see them. Here are a few:

XHTML version 1.1:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://  
www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

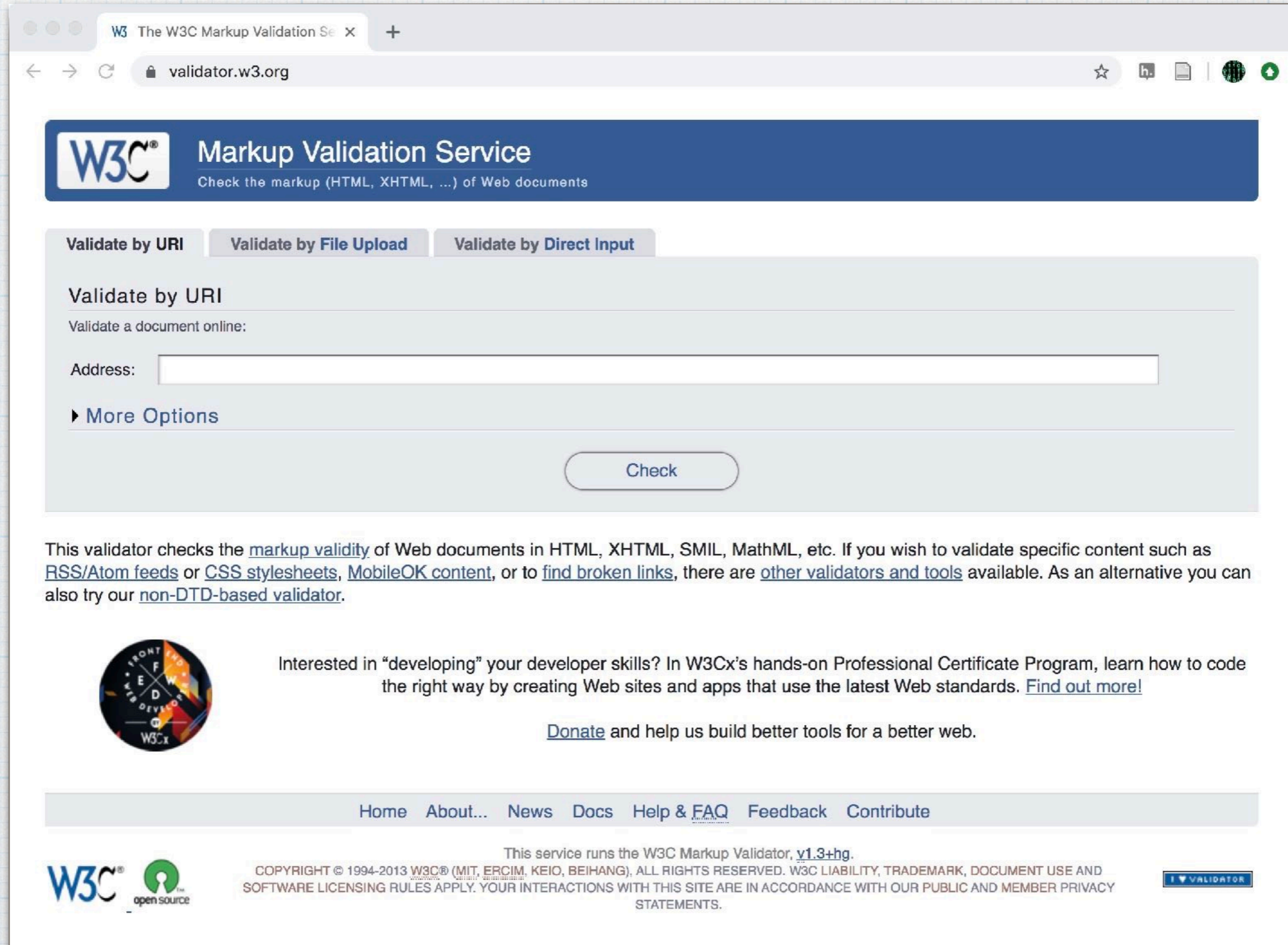
XHTML version 1.0 "Strict":

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://  
www.w3.org/TR/xhtml11/DTD/xhtml11-strict.dtd">
```

HTML version 4.01 Transitional:

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


VALIDATING HTML



The screenshot shows a web browser window with the address bar displaying "validator.w3.org". The page header features the W3C logo and the text "Markup Validation Service" with the subtitle "Check the markup (HTML, XHTML, ...) of Web documents". Below the header, there are three tabs: "Validate by URI" (selected), "Validate by File Upload", and "Validate by Direct Input". The "Validate by URI" section contains a text input field labeled "Address:" and a "Check" button. Below this, there is a "More Options" link. A paragraph of text explains the validator's capabilities and provides links to other tools. A circular logo for "FRONT END DEV W3Cx" is shown next to a promotional message for a Professional Certificate Program. A navigation menu at the bottom includes links for Home, About..., News, Docs, Help & FAQ, Feedback, and Contribute. The footer contains the W3C logo, an "open source" badge, copyright information, and a "I ♥ VALIDATOR" button.

W3C® Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI | Validate by File Upload | Validate by Direct Input

Validate by URI


Validate a document online:

Address:

▶ More Options

Check

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).



Interested in “developing” your developer skills? In W3Cx’s hands-on Professional Certificate Program, learn how to code the right way by creating Web sites and apps that use the latest Web standards. [Find out more!](#)

[Donate](#) and help us build better tools for a better web.

Home About... News Docs Help & FAQ Feedback Contribute

W3C® open source

This service runs the W3C Markup Validator, [v1.3+hg](#).
COPYRIGHT © 1994-2013 W3C® (MIT, ERCIM, KEIO, BEIHANG), ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.

I ♥ VALIDATOR

Use the HTML Validator at <https://validator.w3.org/> to validate your pages.

XML EXAMPLE: POETRY

```
<poem author="Dickinson, Emily">
  <stanza>
    <line n="1">I heard a Fly buzz - when I died -</line>
    <line n="2">The Stillness in the Room</line>
    <line n="3">Was like the Stillness in the Air -</line>
    <line n="4">Between the Heaves of Storm -</line>
  </stanza>

  <stanza>
    <line n="5">The Eyes around - had wrung them dry -</line>
    <line n="6">And Breaths were gathering firm</line>
    <line n="7">For that last Onset - when the King</line>
    <line n="8">Be witnessed - in the Room -</line>
  </stanza>
</poem>
```

HTML is a poor choice to mark up literature like this. Here, tags like **<poem>** and **<stanza>** and **<line>** and attributes like **author** are not legal HTML. This is an example of Extensible Markup Language, or XML.

The point of XML is that we can invent our own tags and our own grammar. Here, a **<poem>** contains **<stanza>** containers, which in turn contain **<line>** tags.

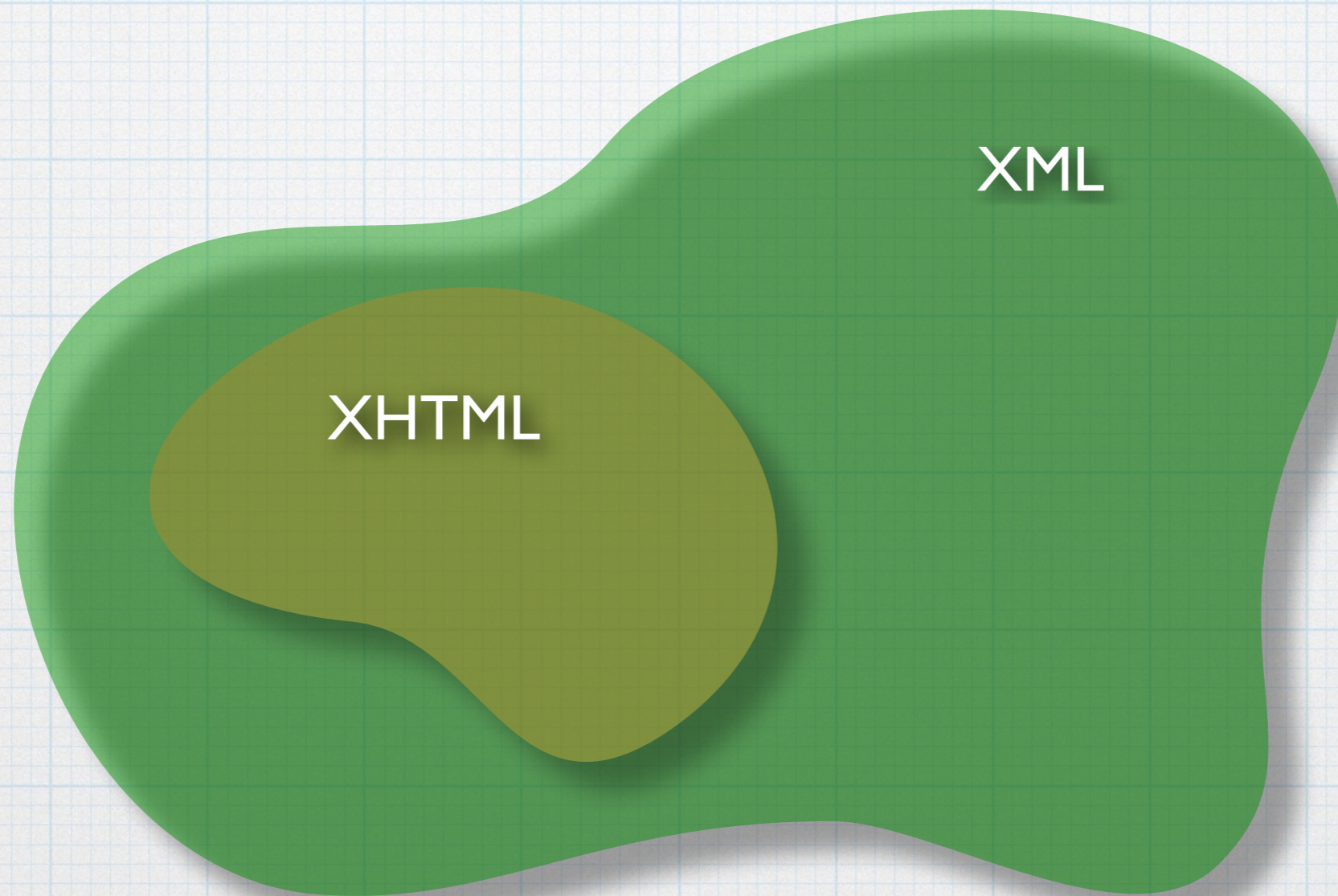
XML EXAMPLE: MARC RECORD

```
<record>
  <datafield tag="245" ind1="1" ind2="0">
    <subfield code="a">Traffic :</subfield>
    <subfield code="b">why we drive the way we do (and what it says about us) /</subfield>
    <subfield code="c">Tom Vanderbilt.</subfield>
  </datafield>
  <datafield tag="250" ind1=" " ind2=" ">
    <subfield code="a">1st ed.</subfield>
  </datafield>
  <datafield tag="260" ind1=" " ind2=" ">
    <subfield code="a">New York :</subfield>
    <subfield code="b">Alfred A. Knopf,</subfield>
    <subfield code="c">2008.</subfield>
  </datafield>
</record>
```

Here's a simplified MarcXML record that records bibliographic data about a book. These data containers have precise meanings and HTML would be a poor choice to capture the complexity of this information.

HTML AND XML

The last version of HTML before version 5 obeyed all the rules of XML. It was even called XHTML to emphasize that relationship.



XML AND XHTML RULES

XML and XHTML follow some precise rules:

1) all tags must always be in lowercase.

2) all attribute values must always be in quotation marks.

```
tag="245"
```

3) all attributes must always take a value:

```
selected="selected"
```

3) all empty tags must be "self-closing" by including a terminating / character:

```
<meta charset="utf-8" />
```

```
<br />
```

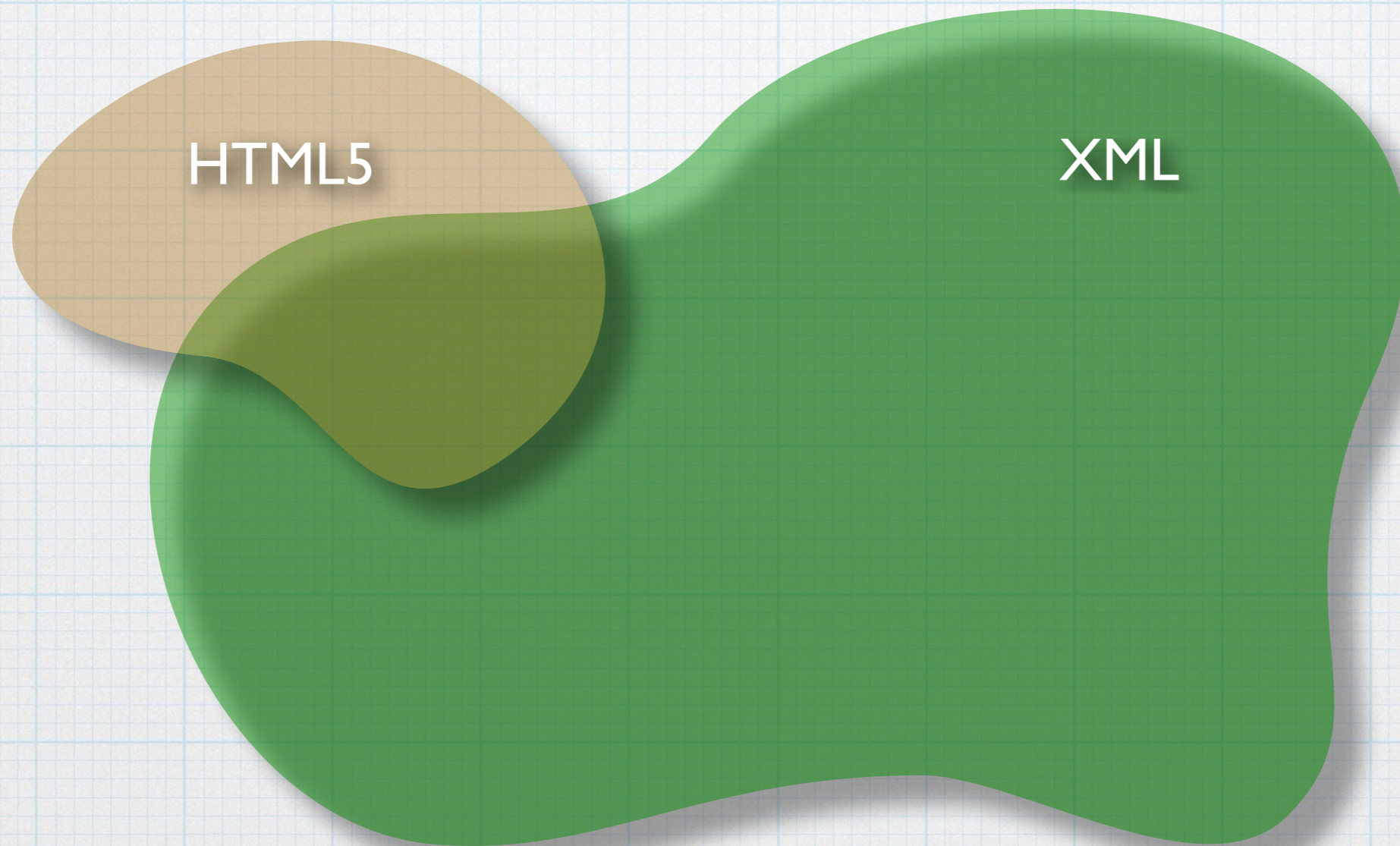
```
<hr />
```

```

```

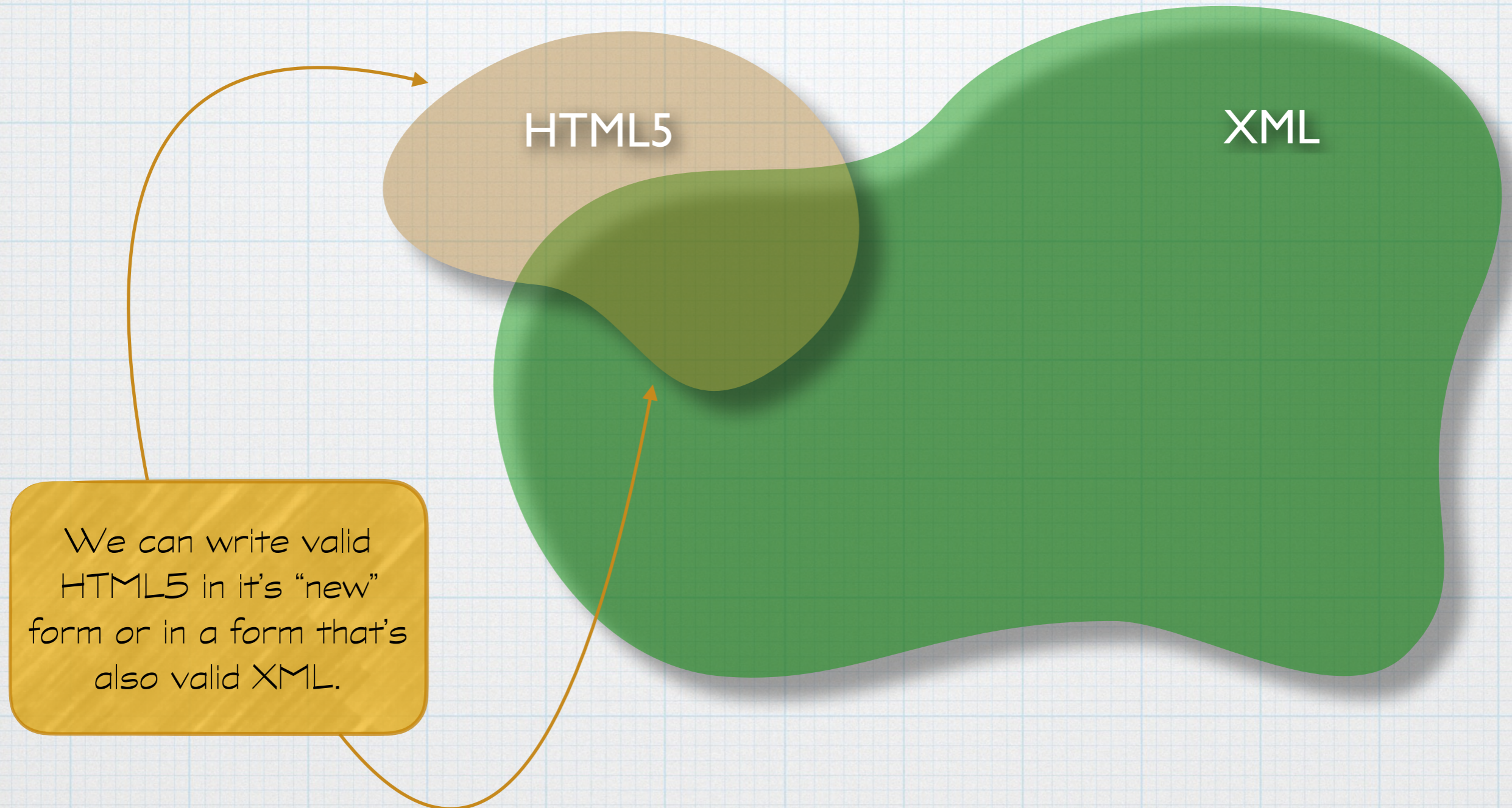
HTML AND XML

The W3C decided, in order to simplify the language, that HTML5 could break several of the old XML rules. HTML5 is no longer strictly XML — which really annoys the XML community and even broke a lot of XML tools that people use to parse and build websites.



HTML AND XML

HTML5 validates either in its new form or in an older, XML-compliant form.



HTML OBEYING XML RULES

```
<!DOCTYPE html>  
<html lang="en">
```

```
<head>  
  <meta charset="utf-8" />  
  <title>My Webpage</title>  
</head>
```

```
<body>  
  <p>I'm at the <a href="http://www.ualberta.ca">  
  University of Alberta</a>.</p>  
</body>
```

```
</html>
```

Notice the quotes on attribute values and the closing / on empty tags.

QUICK REVIEW

- * HTML is “HyperText Markup Language.” We’re on version 5.
- * To put tags into a document is to “mark it up” and we call the result “markup.” (Note: you’re probably going to irritate some programmers if you call this “coding.”)
- * We can create markup files using any plain text editor.
- * The “Document-Type Declaration” (DTD) or “Schema” specifies both the allowable vocabulary and the grammatical rules of the language in question.
- * The World Wide Web Consortium (W3C) creates all the HTML DTDs.
- * Extensible Markup Language (XML) is a framework within which we can create our own new languages. In XML, creating the DTD or Schema that defines a new language is our responsibility.
- * To be considered “correct,” any markup language document must:
 - * be well-formed (all tags must nest properly)
 - * be valid (must use proper vocabulary in properly grammatical ways)