XML

XML stands for “e**X**tensible **M**arkup **L**anguage” – a language designed to store and describe information on the web. Unlike an HTML document, an XML document says pretty much nothing about how its information should be displayed in a browser. Rather, its tags and attributes provide metadata about what kind of information it contains, along with the information itself.

What is it good for?

XML has two primary uses for journalists. First, journalists looking to build news applications will find it a handy format for storing information and transmitting information, because it is relatively easy to read and can be edited by a range of software programs. Second, and perhaps more importantly, reading XML can be a way to find useful information that’s hidden “just beneath the surface” of the web.

What does it look like?

A simple XML document might look something like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<elementList>
<element1>This is text data in the document.</element1>
<element2>This is some other data in the document.</element2>
<element3 someAttribute="aValue" />
</elementList>
```

There are a couple of things going here. The very first line is called the “document type (or ‘doc-type’) declaration”; it’s letting the browser know that the rest of the document should be interpreted as XML (as opposed to HTML, JavaScript, or any number of other web languages).

Starting with the line

```
<elementList>
```

we are into the substance of the document itself. Although XML is very flexible, there are only 2 grammatical structures in any XML document:

1. Element tags, which can be either paired or self-closed, and are surrounded by “carrots” (e.g. <>)

2. Attribute tags, which can exist *only* inside of element tags, and have their values surrounded by double quotes

How do I use it?

Element tags can be recognized by the fact that they are always enclosed by “carrots” (e.g. <>) and can have zero or more attributes. Once an element tag has been declared or “opened”, it can be “closed” in one of 2 ways. Either it must be closed with a matching tag of the same name, but preceded by a forward slash, as with:

```
<element2>This is some other data in the document.</element2>
```

or it can be “self-closed” by including the forward slash before the ending carrot:

```
<element3 someAttribute="aValue" />
```

Plain text between paired tags does not need quotation marks.

Attribute tags, by contrast, do not need to be closed. They are simply declared *inside* an element tag by writing the attribute name, followed by an equals sign and its value surrounded by double quotes:

```
<element3 someAttribute="aValue" />
```

Note that there is no space on either side of the equals sign, just as there is no space between the carrots and slashes of an element tag.

An example

So why use attributes for some things and elements for others? XML documents indicate the relationship among pieces of information by organizing them into a hierarchy, which, for readability are indicated by indentations in the code.

An important difference between elements and attributes is that elements can contain other elements (this is referred to as “nesting” one element inside another). If one element contains another, it is said to be a “parent” of that element (making the contained tag its “child”). This relationship allows us to cluster pieces of information together in a useful way. XML is especially useful for expressing lists of information, such as articles or tweets:

```
<?xml version="1.0" encoding="UTF-8"?>
<articleList>
```

```
    <anArticle>
        <articleLink
url="http://www.guardian.co.uk/media/2011/jul/17/hacking-scandal-pr-disaster-for-murd
ochs"/>
        <articleSummary>News Corporation and the Murdochs have shown rarely-seen
incompetence in their handling of the phone-hacking crisis....</articleSummary>
        <articleDate>Sun, 17 Jul 2011 18:00:02 GMT</articleDate>
    <anArticle>


   <anArticle>
        <articleLink
url="http://www.guardian.co.uk/commentisfree/2011/jul/13/rupert-murdoch-back-brink"/>
        <articleSummary>The task of clearing out the muck from News International's
stables may be insurmountable...</articleSummary>
        <articleDate>Wed, 13 Jul 2011 17:59:00 GMT</articleDate>
    <anArticle>
</articleList>
```

Thanks to the descriptive nature of the XML element tags, we don't have to guess what type of information is contained where. If the author expected others to use the information (by making it accessible through an API or RSS feed), there will generally be documentation describing what information each tag contains. Otherwise, you may have to do some guessing.

Where will I find it?

Nowadays, XML is considered something of a legacy data format. The most common place you'll encounter it is when looking at web feeds of government or older databases, for example. Although you may be able to "see" these in a browser window, very often there is more information in the raw XML. That's why learning to read XML is an essential skill for digital journalists.

If you suspect that you're looking at something in your browser window that's actually XML, give it a right-click in and select "View Page Source". If it looks like it's made up of elements and attributes, it's probably XML.

Issues & FAQs

I did that. It's all jammed up on one line. Now what?

FireFox to the rescue (as so often). Copy and paste the whole thing into a new NotePad (NOT WordPad) or equivalent text editor. Save it with a .xml extension, and then open it up in FireFox at it will look much better. What's more, it will holler at you if something's wrong with it (grammatically) and give you an idea of where the issue is.

Tools & Tutorials

XML editing programs: Notepad++ (free), Dreamweaver ($$$), BBEdit (Mac), SublimeText($$)