

XML

Week 4 Session 2

What is XML?

- ▶ XML is a markup language similar to HTML. It stands for Extensible Markup Language and is a W3C recommended specification as a general purpose markup language.
- ▶ This means, unlike other markup languages, XML is not predefined so you must define your own tags.
- ▶ The primary purpose of the language is the sharing of data across different systems, such as the Internet.

XML

- ▶ XML stands for eXtensible Markup Language.
- ▶ XML was designed to describe data.
- ▶ XML is a software- and hardware-independent tool for carrying information.
- ▶ <http://www.w3schools.com/xml/default.asp>

Advantage

- ▶ **XML Separates Data from HTML**
- ▶ **XML Simplifies Data Sharing**
- ▶ **XML Simplifies Data Transport**
- ▶ **XML Simplifies Platform Changes**
- ▶ **XML Makes Your Data More Available**

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="course_list.css"?>
<CourseList>
  <Course>
    <Title>Introduction to Databases</Title>
    <Description>Introduction to Databases
    Description</Description>
    <Id>ISCG5423</Id>
    <prerequisite>non</prerequisite>
    <Credits>15</Credits>
    <Level>5</Level>
  </Course>
  <Course>
    <Title>Hardware Fundamentals</Title>
    <Description>Description Hardware
    Fundamentals</Description>
    <Id>ISCG5400</Id>
    <prerequisite>non</prerequisite>
    <Credits>15</Credits>
    <Level>5</Level>
  </Course>
</CourseList>
```

Other Example

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="product_list.css"?>
<ProductList>
  <Product>
    <ProductName>milk</ProductName>
    <Image>milk.jpg</Image>
    <Price>5</Price>
  </Product>
  <Product>
    <ProductName>agria potatoes</ProductName>
    <Image>agria_potatoes.jpg</Image>
    <Price>20</Price>
  </Product>
</ProductList>
```

Formed XML Document - rules

- ▶ XML documents must have a root element
- ▶ XML elements must have a closing tag
- ▶ XML tags are case sensitive
- ▶ XML elements must be properly nested
- ▶ XML attribute values must be quoted

XML Validator

http://www.w3schools.com/xml/xml_validator.asp

XML Document Type Definition

The DTD defines the structure and the legal elements and attributes of an XML document.

It is a ruleset for validating an XML document. It provides means of ensuring the contents of XML is readable by all document contact points.

DTD is usually declared in a separate file and linked to the XML file in the DOCTYPE definition:

```
<!DOCTYPE note SYSTEM "Note.dtd">
```

XML DTD `<!DOCTYPE note SYSTEM "Note.dtd">`

File Contents may look something like this:

```
<!DOCTYPE note  
[  
  <!ELEMENT note (to,from,heading,body)>  
  <!ELEMENT to (#PCDATA)>  
  <!ELEMENT from (#PCDATA)>  
  <!ELEMENT heading (#PCDATA)>  
  <!ELEMENT body (#PCDATA)>  

```

Root element is "note"

Required elements inside root

Defines the contents of each specified element to be #PCDATA which is Parseable Character Data

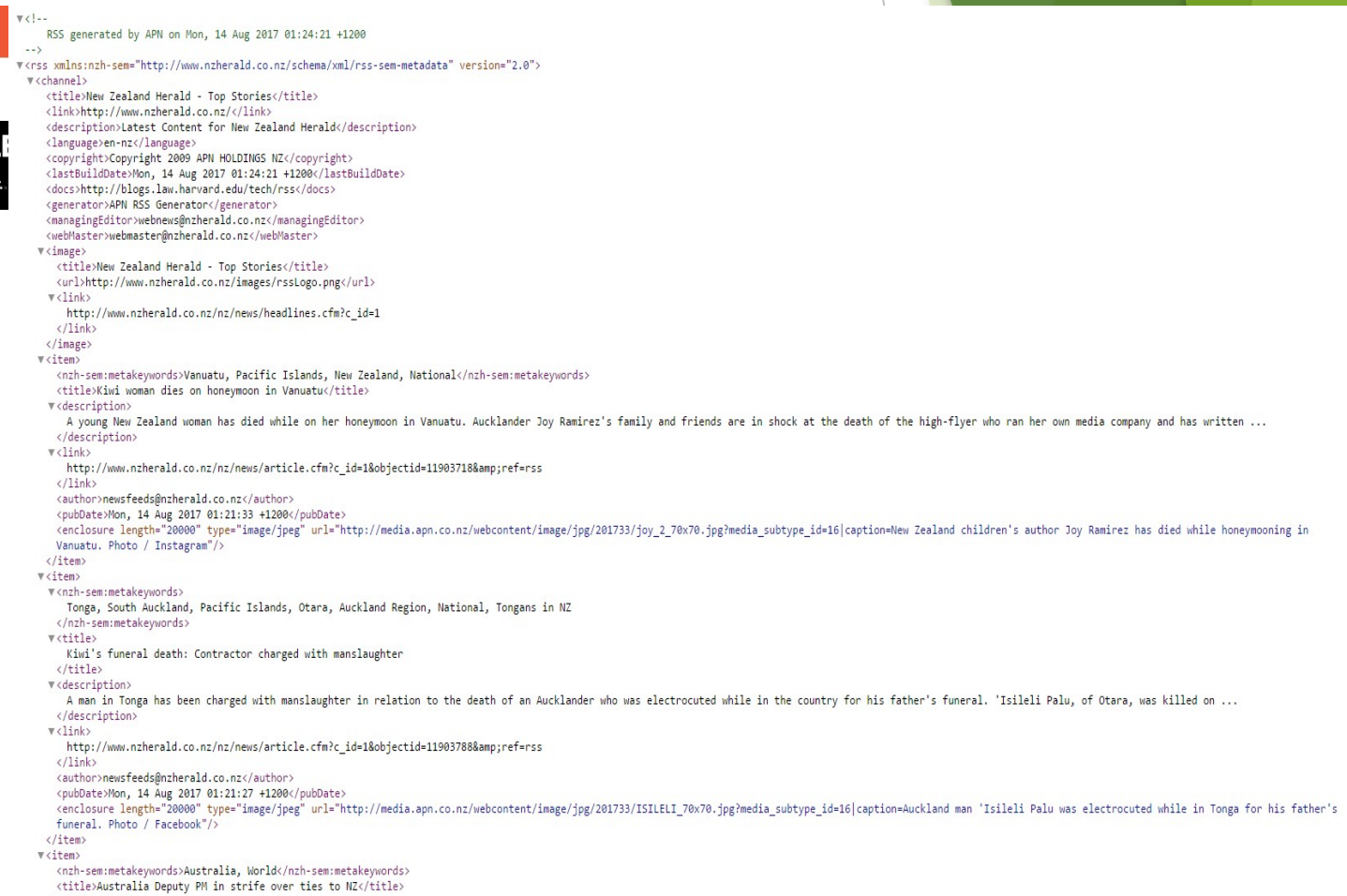
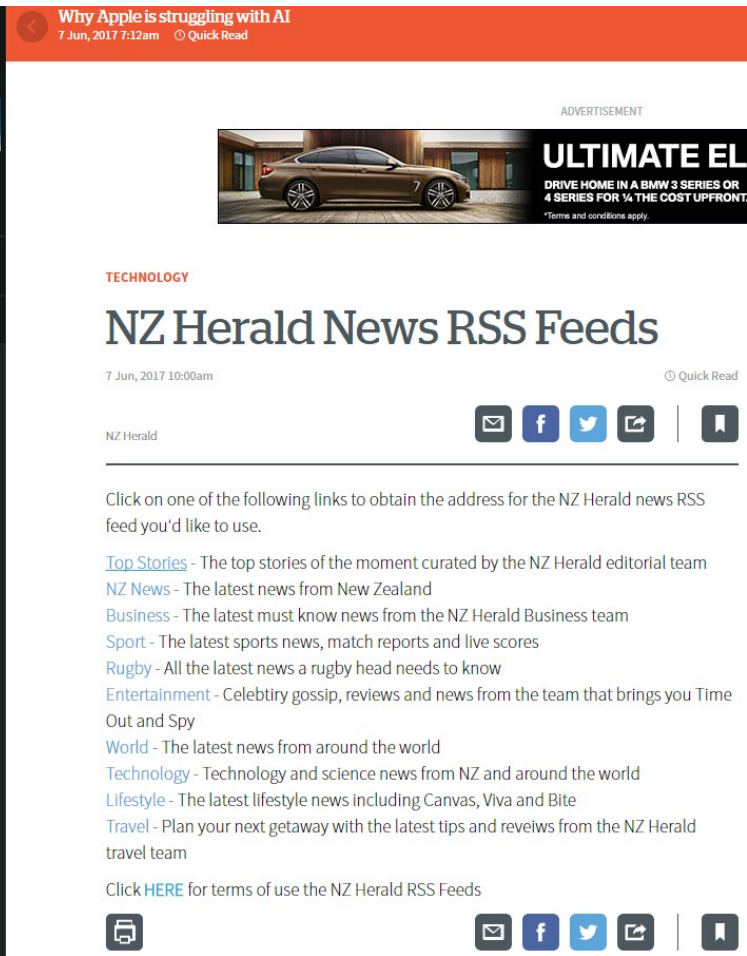
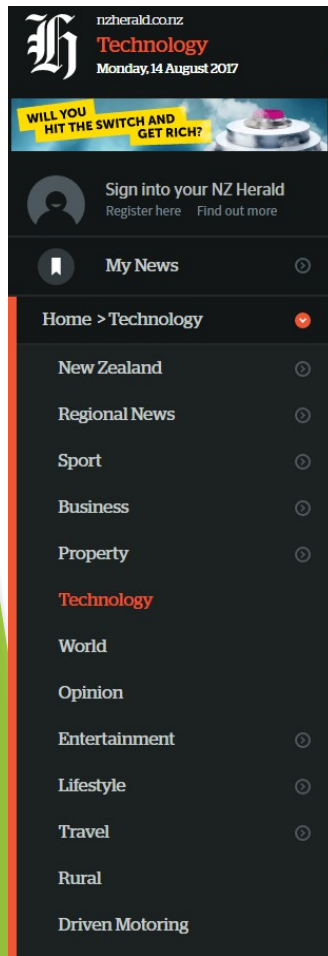
XML with CSS display

milk.jpg		
cornflake	cornflake.jpg	10
soda	soda.jpg	2
bread	bread.jpg	8
butter	butter.jpg	2
agria potatoes	agria_potatoes.jpg	20

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/css" href="product_
3 <ProductList>
4   <Product>
5     <ProductName>milk</ProductName>
6     <Image>milk.jpg</Image>
7     <Price>5</Price>
8   </Product>
9   <Product>
10    <ProductName>cornflake</ProductName>
11    <Image>cornflake.jpg</Image>
12    <Price>10</Price>
13  </Product>
14  <Product>
15    <ProductName>soda</ProductName>
16    <Image>soda.jpg</Image>
17    <Price>2</Price>
18  </Product>
19  <Product>
20    <ProductName>bread</ProductName>
21    <Image>bread.jpg</Image>
22    <Price>8</Price>
23  </Product>
24  <Product>
25    <ProductName>butter</ProductName>
26    <Image>butter.jpg</Image>
27    <Price>2</Price>
28  </Product>
29  <Product>
30    <ProductName>agria potatoes</ProductNam
31    <Image>agria_potatoes.jpg</Image>
32    <Price>20</Price>
33  </Product>
34 </ProductList>
35
```

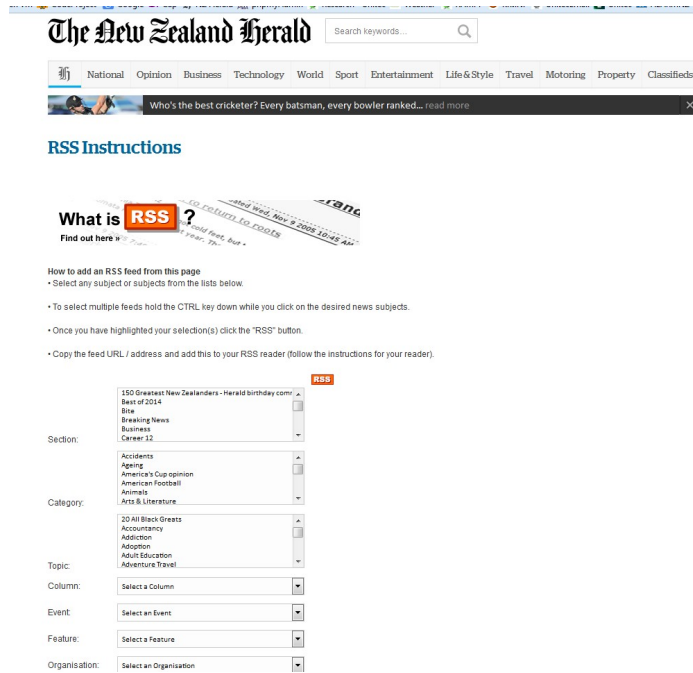
Example -NZ Herald – RSS feed

- ▶ <http://www.nzherald.co.nz/rss/>
- ▶ http://rss.nzherald.co.nz/rss/xml/nzhrsscid_000000001.xml



RSS (Rich Site Summary) - User interface of a feed reader

- ▶ <http://www.nzherald.co.nz/rss/>
- ▶ http://rss.nzherald.co.nz/rss/xml/nzhrsscid_0000000003.xml



- ▶ RSS feeds enable publishers to syndicate data automatically.
- ▶ A standard XML file format ensures compatibility with many different machines/programs.
- ▶ RSS feeds also benefit for timely updates from favourite websites or to aggregate data from many sites.

The XMLHttpRequest Object

- ▶ The XMLHttpRequest object is used to exchange data with a server behind the scenes.
- ▶ The XMLHttpRequest object is **a developer's dream**, because you can:
- ▶ Update a web page without reloading the page
- ▶ Request data from a server after the page has loaded
- ▶ Receive data from a server after the page has loaded
- ▶ Send data to a server in the background

Access Across Domains

- ▶ For security reasons, modern browsers do not allow access across domains.
- ▶ This means that both the web page and the XML file it tries to load, must be located on the same server.
- ▶ The examples on W3Schools all open XML files located on the W3Schools domain.
- ▶ If you want to use the example above on one of your own web pages, the XML files you load must be located on your own server.

Parse an XML String

- Browsers use the DOMParser object.

```
txt="<bookstore><book>";  
txt=txt+"<title>Everyday Italian</title>";  
txt=txt+"<author>Giada De Laurentiis</author>";  
txt=txt+"<year>2005</year>";  
txt=txt+"</book></bookstore>";
```

```
parser=new DOMParser();  
xmlDoc=parser.parseFromString(txt,"text/xml");
```


Create an XMLHttpRequest Object

- ▶ All modern browsers (Firefox, Chrome, Safari, and Opera) have a built-in XMLHttpRequest object.
- ▶ Syntax for creating an XMLHttpRequest object:

```
xmlhttp=new XMLHttpRequest();
```

Parse an XML Document

- ▶ The following code fragment parses an XML document into an XML DOM object:

```
xmlhttp=new XMLHttpRequest();  
xmlhttp.open("GET","books.xml",false);  
xmlhttp.send();  
xmlDoc=xmlhttp.responseXML;
```

XML DOM

The XML DOM

- ▶ The XML DOM defines a standard way for accessing and manipulating XML documents.
- ▶ The XML DOM views an XML document as a tree-structure.
- ▶ All elements can be accessed through the DOM tree. Their content (text and attributes) can be modified or deleted, and new elements can be created. The elements, their text, and their attributes are all known as nodes.

The HTML DOM

- ▶ The HTML DOM defines a standard way for accessing and manipulating HTML documents.
- ▶ All HTML elements can be accessed through the HTML DOM.

XML and HTML display

Note: If you are testing this file on

Shop

display all products

milk	\$5		<input type="text" value="0"/>
cornflake	\$10		<input type="text" value="0"/>
soda	\$2		<input type="text" value="0"/>
bread	\$8		<input type="text" value="0"/>
butter	\$2		<input type="text" value="0"/>
agri potatoes	\$20		<input type="text" value="0"/>

enter

```
7 <script>
8 function product(productName, price, image) {
9     this.productName = productName;
10    this.price = price;
11    this.image = image;
12 }
13
14 var products = []; //Array declaration - empty, and it will be load from XML file
15 // XML -The XMLHttpRequest object is used to exchange data with a server behind the scenes.
16
17 if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
18     xmlhttp=new XMLHttpRequest();
19 }
20 else { // code for IE6, IE5
21     xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
22 }
23 xmlhttp.open("GET","product_list.xml",false);
24 xmlhttp.send();
25 xmlDoc=xmlhttp.responseXML;
26 var x=xmlDoc.getElementsByTagName("Product");
27
28 var pName='';
29 var pPrice=0;
30 var pImage='';
31 for (i=0; i<x.length; i++) {
32     pName=x[i].getElementsByTagName("ProductName")[0].childNodes[0].nodeValue;
33     pPrice=Number(x[i].getElementsByTagName("Price")[0].childNodes[0].nodeValue);
34     pImage=x[i].getElementsByTagName("Image")[0].childNodes[0].nodeValue;
35     var prod =new product(pName, pPrice, pImage);
36     products[i]=prod;
37 }
38
39
40
41 function show_products() {
42     var displaytext='';
43     var table = document.getElementById('dispProduct');
44     for(var i=0; i < products.length; i++) {
45
46         var row = table.insertRow(i);
47         var cell1 = row.insertCell(0);
48         var cell2 = row.insertCell(1);
49         var cell3 = row.insertCell(2);
```

XMLHttpRequest cannot load file.

Cross origin requests are only supported for HTTP

Test on Firefox, Deploy to Host

- ▶ Firefox no longer allows Cross Origin Requests from files served from your hard drive
- ▶ Your web hosting site will allow requests to files in folders as configured by the manifest file

Run a Local Server

- ▶ Run a server on your computer, like Apache or Python
- ▶ Python isn't a server, but it will run a simple server

<http://stackoverflow.com/questions/20041656/xmlhttprequest-cannot-load-file-cross-origin-requests-are-only-supported-for-http>

- ▶ <https://addons.mozilla.org/en-US/firefox/addon/cors-everywhere/>
- ▶ <https://chrome.google.com/webstore/detail/allow-cors-access-control/lhobafahddgcelffkeicbaginigejlf>

XMLHttpRequest Example

```
<script>
if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome,
      Opera, Safari
      xmlhttp=new XMLHttpRequest();
    }
else
    { // code for IE6, IE5
      xmlhttp=new
      ActiveXObject("Microsoft.XMLHTTP")
      ;
    }
xmlhttp.open("GET","course_list.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
```

```
document.write("<table><tr><th>Course</th><th>Title</th></tr>");
var x=xmlDoc.getElementsByTagName("Course");
for (i=0;i<x.length;i++)
{
    document.write("<tr><td>");
    document.write(x[i].getElementsByTagName("Id")[0].childNodes[0].nodeValue);
    document.write("</td><td>");
    document.write(x[i].getElementsByTagName("Title")[0].childNodes[0].nodeValue);
    document.write("</td></tr>");
}
document.write("</table>");
</script>
```

What is different on Chrome:

```
document.write(x[i].getElementsByTagName("Title")  
[0].childNodes[0].nodeValue);
```

Work on server and with **single quote**.

```
document.write(x[i].getElementsByTagName('Title')  
[0].childNodes[0].nodeValue);
```

References

<https://www.w3schools.com/>