


COMP4021  
Internet Computing

# XML

David Rossiter and Gibson Lam

# A Simple XML File

- An XML file is used for storing data
- You can easily make up your own XML
- Here is a very simple XML document:



```
<?xml version="1.0"?>  
<message id="1">Good!</message>
```

- The first line is optional, so you could just do this:

```
<message id="1">Good!</message>
```

# Improving the Document

- Here's an improved XML document:

```
<?xml version="1.0"?>
```

```
<messages>
```

有两行或以上message的时候就需要用messages扩起来。相当于给树加一个 "root"

```
<message id="1">How are you?</message>
```

```
<message id="2">I am fine!</message>
```

```
</messages>
```

# Displaying XML in a Browser

- If you simply load an XML file into a browser, the browser will show the XML **browser will take XML file as text**
- For example, this is what you will see if an XML file is loaded into Chrome:

**no style for XML!**



This XML file does not appear to have any style information associated with it. The document tree is shown below.

If the XML code has some "branches", you can expand or

▼ `<messages>` collapse the branches in the browser  
    `<message id="1">How are you?</message>`  
    `<message id="2">I am fine!</message>`  
    `</messages>`

# Collapsing / Expanding Branches

- In the browser display, if you want to you can collapse or expand some of the XML branches:



```
<messages>
  <message id="1">How are you?</message>
  <message id="2">I am fine!</message>
</messages>
```



*Collapsing / expanding  
the structure*

```
▶ <messages>...</messages>
```

- It's sometimes useful to collapse part of the XML if the whole thing is big

We can use CSS style sheet to give proper styles to pure XML.

Otherwise, in an html file, we can use jQuery to grab an .xml file and process it (convert xml code to html code and append/add the code to proper places in the html file (by using selectors)

## XML By Itself Doesn't Have Any Style

- You will see the following message when you load 'pure' XML files into a browser:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

---

- The browser displays the XML in a very simple way, because it doesn't know how to display it in any other way

# Making Good XML

- XML must be properly written (called *well formed*):
  1. There can only be one root element
  2. Opening and tags must match
  3. XML tags and attributes are case-sensitive
  4. XML attribute values must use quotes
  5. XML tags must be properly nested
- In the following slides we will talk about these

# 1) There Must Be One Root Element

- If you use more than one root element, the XML file will result in an error:

```
<?xml version="1.0"?>  
<message id="1">How are you?</message>  
<message id="2">I am fine!</message>
```

*Bad!*  
*There are*  
*two root*  
*elements*



*Load this XML*  
*into a browser*

**This page contains the following errors:**

error on line 3 at column 1: Extra content at the end of the document



## 2 & 3) Opening and Closing Tags Must Match

- The case of both opening and closing tags must match i.e. the following example won't work:

tags and attributes are case sensitive

```
<?xml version="1.0"?>  
<message id="1">How are you?</Message >
```

*Bad! Different tag name*



*Load this XML  
into a browser*

**This page contains the following errors:**

error on line 2 at column 45: Opening and ending tag mismatch: message line 0 and Message

# XML Closing Tags

- If there's nothing between the opening and closing tags you can use this quick way to close tags:

```
<timestamp value="14385683" />
```

You can simply do this if there is nothing in the middle of the opening and closing tag.

- The above is equivalent to:

```
<timestamp value="14385683"></timestamp>
```

- Both are OK

## 4) Use Quotes for Attributes

- If you don't use quotes (either using " or ') around an attribute then you will get an XML error:

```
<?xml version="1.0"?>  
<message id=1>How are you?</message>
```

*Bad! There's no speech  
marks around the "1"*



**This page contains the following errors:**

error on line 2 at column 13: AttValue: " or ' expected

## 5) Tags Must Be Properly Nested

- Tags must be properly nested, for example:

```
<?xml version="1.0"?>
<messages>
  <question>
    How are you?
  <answer>
    I am fine!
  </question>
</answer>
</messages>
```

*The tag sequence  
is wrong!*



*Load this XML into a browser*

**This page contains the following errors:**

error on line 7 at column 18: Opening and ending tag mismatch: answer line 0 and question

# Using Style Sheets With XML

- So how can we display XML in a nicer way?
- We can use a style sheet e.g. add this to the XML file:

```
<?xml-stylesheet href="04_xml_with_style.css"?>
```

*An external stylesheet*

- The style sheet is then used by the browser to display the XML – see next slide

```
messages {
  font-size:20px;
  margin:1rem;
  background-color:gray
}
message {
  display:block;
  margin-bottom:0.5rem;
  background-color:aliceblue;
}
```

04\_xml\_with\_style.css

## A Style Sheet Example

- Here we use a style sheet together with the XML file to make the content appear nicely

```
<?xml version="1.0"?>
<?xml-stylesheet href="04_xml_with_style.css"?>
<messages>
  <message id="1">How are you?</message>
  <message id="2">I am fine!</message>
</messages>
```

04\_xml\_with\_style.xml

How are you?

I am fine!

*Browser display*

# The Style Sheet

```
messages {  
  font-size:20px;  
  margin:1rem; i.e. 20px  
  background-color:gray  
}
```

- Fairly large text size
- **rem** means the **font size of the root element** of the document

- In other words, there will be a margin all around the messages area which has a size of 20px

```
message {  
  display:block;  
  margin-bottom:0.5rem; i.e. 10px  
  background-color:aliceblue;  
}
```

- Add a margin under each message so there's a gap between messages

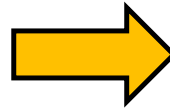
- Displays an element as a **block** element (like `<p>`) which means it **starts on a new line and takes up the whole width**

# Loading XML into a Web Page

## Example of loading XML into a Web Page

Get the XML

*After  
clicking*



## Example of loading XML into a Web Page

Get the XML

How are you?

I am fine!

- More commonly, XML is loaded into the DOM
- Then the XML (which you can't see yet) is converted into something which you can see i.e. some HTML
- On the following slides we show a jQuery example



```
<?xml version="1.0"?>
<messages>
  <message id="1">How are you?</message>
  <message id="2">I am fine!</message>
</messages>
```

- XML is loaded from the server

- XML content is converted into HTML

td means a box in a table cell

```
<tr><td>How are you?</td></tr>
<tr><td>I am fine!</td></tr>
```

messages in XML are converted to rows inside a table

- Browser displays the HTML

How are you?

I am fine!

## The Basic Idea

```
<!DOCTYPE html>
```

```
<html> <head>
```

```
<script
```

```
  src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
```

```
</script>
```

```
<style>
```

```
td {
```

```
  border:1px solid black;
```

```
  padding:6px;
```

```
}
```

```
</style>
```

How are you?

I am fine!

- When the XML content is displayed in the web page, it is displayed in a box (td) in a table – this style instruction means it has nice spacing and a border line around each message

## Example of Loading XML into a Web Page 1/3

```
<script>
```

```
function processXML() {
```

A jquery function - ajax: grab some data from the web server

```
$.ajax({
```

```
url: '02_xml_with_more_structure.xml',
```

```
type: "GET",
```

```
dataType: "xml",
```

If grab the data successfully

```
success: function (result) {
```

```
$(result).find('message').each(function () {
```

```
$("#show_xml_content").append("<tr><td>" + $(this).text() +  
"</td></tr>");
```

```
    } )
```

```
}
```

```
});
```

```
} </script> </head>
```


- When the request for the XML file is successful, find the thing called *show\_sml\_content* in the web page

and convert each message into a table cell (td) which goes inside a table row (tr)

## Example of Loading XML into a Web Page 2/3

```
<body>
```


```
<h1>Example of loading XML into a Web Page</h1>
```

- Click here to trigger the action
- 

```
<button type="button" onclick="processXML()">Get the XML</button>
```

```
<br><br>
```

```
<table id="show_xml_content"></table>
```

- The XML content (the messages) are put in here
- 

```
</body>
```

```
</html>
```

Example of Loading  
XML into a Web Page 3/3