

Getting Data from the Web with R

Part 3: Basics of XML and HTML

Gaston Sanchez

April-May 2014

Content licensed under [CC BY-NC-SA 4.0](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<movies>
  <movie mins="126" lang="eng">
    <title>Good Will Hunting</title>
    <director>
      <first_name>Gus</first_name>
      <last_name>Van Sant</last_name>
    </director>
    <year>1998</year>
    <genre>drama</genre>
  </movie>
  <movie mins="106" lang="spa">
    <title>Y tu mama tambien</title>
    <director>
      <first_name>Alfonso</first_name>
      <last_name>Cuaron</last_name>
    </director>
    <year>2001</year>
    <genre>drama</genre>
  </movie>
</movies>
```

Readme

License:

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
<http://creativecommons.org/licenses/by-nc-sa/4.0/>

You are free to:

- Share** — copy and redistribute the material
- Adapt** — rebuild and transform the material

Under the following conditions:

- Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made.
- NonCommercial** — You may not use this work for commercial purposes.
- Share Alike** — If you remix, transform, or build upon this work, you must distribute your contributions under the same license to this one.

Lectures Menu

Slide Decks

1. Introduction
2. Reading files from the Web
3. **Basics of XML and HTML**
4. Parsing XML / HTML documents
5. Handling JSON data
6. HTTP Basics and the RCurl package
7. Getting data via Web Forms
8. Getting data via Web APIs

Basics of XML and HTML

Goal

XML & HTML

The goal of these slides is to give you a **crash introduction to XML and HTML** so you can get a good grasp of those formats for the rest of the lectures

Synopsis

In a nutshell

We'll cover a the following concepts:

- ▶ Importance of XML and HTML
- ▶ Hierarchical Structure
- ▶ Document Object Model (DOM)

Some References

- ▶ XML Files website (<http://www.xmlfiles.com>)
by Jan Egil Refsnes
- ▶ XML in a Nutshell
by Elliotte Rusty Harold; W. Scott Means
- ▶ XML Tutorial (<http://www.w3schools.com/xml/default.asp>)
by w3schools
- ▶ Introduction to Data Technologies
by Paul Murrell
- ▶ XML and Web Technologies for Data Sciences with R
by Deb Nolan and Duncan Temple Lang

XML and HTML

Why you should care about XML and HTML?

- ▶ Large amounts of data and information are stored, shared and distributed using HTML and XML-dialects
- ▶ They are widely adopted and used in many applications
- ▶ Working with data from the Web means dealing with HTML

XML

eXtensible Markup Language

```
1 <?xml version="1.0" encoding="ISO8859-1" ?>
2 <CATALOG>
3   <PLANT>
4     <COMMON>Bloodroot</COMMON>
5     <BOTANICAL>Sanguinaria canadensis</BOTANICAL>
6     <ZONE>4</ZONE>
7     <LIGHT>Mostly Shady</LIGHT>
8     <PRICE>$2.44</PRICE>
9     <AVAILABILITY>031599</AVAILABILITY>
10  </PLANT>
11
12  <PLANT>
13    <COMMON>Columbine</COMMON>
14    <BOTANICAL>Aquilegia canadensis</BOTANICAL>
15    <ZONE>3</ZONE>
16    <LIGHT>Mostly Shady</LIGHT>
17    <PRICE>$9.37</PRICE>
18    <AVAILABILITY>030699</AVAILABILITY>
19  </PLANT>
20
21  <PLANT>
22    <COMMON>Marsh Marigold</COMMON>
23    <BOTANICAL>Caltha palustris</BOTANICAL>
24    <ZONE>4</ZONE>
25    <LIGHT>Mostly Sunny</LIGHT>
26    <PRICE>$6.81</PRICE>
27    <AVAILABILITY>051799</AVAILABILITY>
28  </PLANT>
29
30  <PLANT>
```

Some Definitions

“XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable”

<http://en.wikipedia.org/wiki/XML>

“XML is a data description language used for describing data”

Paul Murrell

Introduction to Data Technologies

Some Definitions

“XML is a very general structure with which we can define any number of new formats to represent arbitrary data”

“XML is a standard for the semantic, hierarchical representation of data”

Deb Nolan & Duncan Temple Lang

XML and Web Technologies for Data Sciences with R

About XML

XML

XML stands for **eXtensible Markup Language**

Broadly speaking ...

XML provides a flexible framework to create formats for describing and representing data

Markups

Markup

A **markup** is a sequence of characters or other symbols inserted at certain places in a document to indicate either:


- ▶ how the content should be displayed when printed or in screen
- ▶ describe the document's structure

Markup Language

A markup language is a system for **annotating** (i.e. *marking*) a document in a way that the content is distinguished from its representation (eg LaTeX, PostScript, HTML, SVG)

Markups

XML Markups

In XML (as well as in HTML) the marks (aka *tags*) are defined using angle brackets: 

```
<mark>Text marked with special tag</mark>
```


Extensible

Extensible?

The concept of *extensibility* means that we can define our own marks, the order in which they occur, and how they should be processed. For example:

- ▶ `<my_mark>`
- ▶ `<awesome>`
- ▶ `<boring>`
- ▶ `<pathetic>`

About XML

XML is NOT

- ▶ a programming language
- ▶ a network transfer protocol
- ▶ a database

XML is

- ▶ more than a markup language
- ▶ a generic language that provides structure and syntax for representing any type of information
- ▶ a meta-language: it allows us to create or define other languages

XML Applications

Some XML dialects

- ▶ **KML** (*Keyhole Markup Language*) for describing geo-spatial information used in Google Earth, Google Maps, Google Sky
- ▶ **SVG** (*Scalable Vector Graphics*) for visual graphical displays of two-dimensional graphics with support for interactivity and animation
- ▶ **PMML** (*Predictive Model Markup Language*) for describing and exchanging models produced by data mining and machine learning algorithms

XML Applications (con't)

Some XML dialects

- ▶ **RSS** (*Rich Site Summary*) feeds for publishing blog entries
- ▶ **SDMX** (*Statistical Data and Metadata Exchange*) for organizing and exchanging statistical information
- ▶ **GML** (*Geography Markup Language*) for representing geographical features
- ▶ **SBML** (*Systems Biology Markup Language*) for describing biological systems

Minimalist Example



XML Example

Ultra Simple XML

```
<movie>  
  Good Will Hunting  
</movie>
```

- ▶ one single element *movie*
- ▶ start-tag: `<movie>`
- ▶ end-tag: `</movie>`
- ▶ content: `Good Will Hunting`

XML Example

Ultra Simple XML

```
<movie mins="126" lang="en">  
  Good Will Hunting  
</movie>
```

- ▶ xml elements can have **attributes**
- ▶ attributes: **mins** (minutes) and **lang** (language)
- ▶ attributes are *attached* to the element's start tag
- ▶ attribute values **must be quoted!**

XML Example

Minimalist XML

```
<movie mins="126" lang="en">  
  <title>Good Will Hunting</title>  
  <director>Gus Van Sant</director>  
  <year>1998</year>  
  <genre>drama</genre>  
</movie>
```

- ▶ an xml element may contain other elements
- ▶ *movie* contains several elements: *title*, *director*, *year*, *genre*

XML Example

Simple XML

```
<movie mins="126" lang="en">
  <title>Good Will Hunting</title>
  <director>
    <first_name>Gus</first_name>
    <last_name>Van Sant</last_name>
  </director>
  <year>1998</year>
  <genre>drama</genre>
</movie>
```

- ▶ Now *director* has two child elements: *first_name* and *last_name*

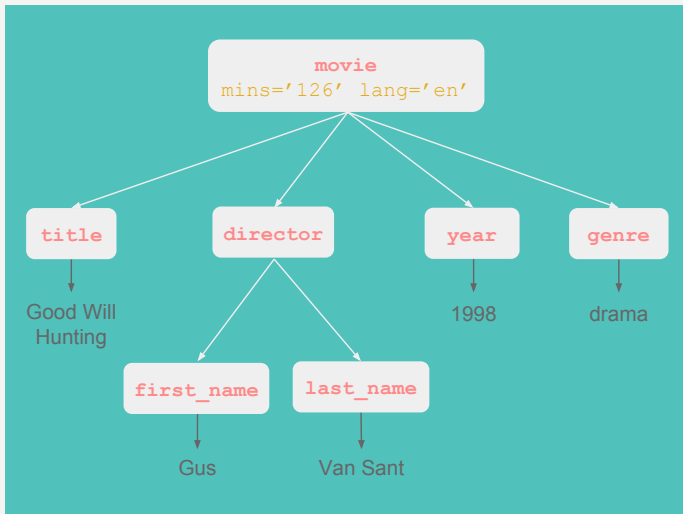
XML Hierarchy Structure

Conceptual XML

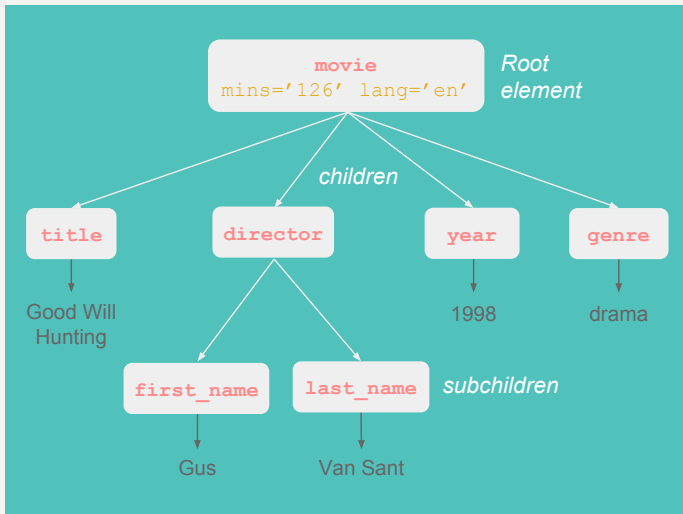
```
<Root>  
  <child_1>...</child_1>  
  <child_2>...</child_2>  
    <subchild>...</subchild>  
  <child_3>...</child_3>  
</Root>
```

- ▶ An XML document can be represented with a **tree structure**
- ▶ An XML document must have **one single Root** element
- ▶ The Root may contain child elements
- ▶ A child element may contain subchild elements

XML Tree Structure



XML Tree Structure (con't)



Well-Formedness

Well-formed XML

We say that an XML document is **well-formed** when it obeys the basic syntax rules of XML. Some of those rules are:

- ▶ one root element containing the rest of elements
- ▶ properly nested elements
- ▶ self-closing tags
- ▶ attributes appear in start-tags of elements
- ▶ attribute values must be quoted
- ▶ element names and attribute names are case sensitive

Well-Formedness

Importance of Well-formed XML

Not well-formed XML documents produce potentially fatal errors or warnings when parsed.

Documents may be well-formed but not valid. Well-formed just guarantees that the document meets the basic XML structure, not that the content is valid.

Additional XML Elements

Some Additional Elements

Example with extra elemets

```
<?xml version="1.0"? encoding="UTF-8" ?>
<![CDATA[ a > 5 & b < 10 ]]>
<?GS print(format = TRUE)>
<!DOCTYPE Movie>
<!-- This is a commet -->
<movie mins="126" lang="en">
  <title>Good Will Hunting</title>
  <director>
    <first_name>Gus</first_name>
    <last_name>Van Sant</last_name>
  </director>
  <year>1998</year>
  <genre>drama</genre>
</movie>
```

Additional Elements

Additional (optional) XML elements

Markup	Description
<code><?xml ></code>	XML Declaration identifies content as an XML document
<code><?PI ></code>	Processing Instruction processing instructions passed to application PI
<code><!DOCTYPE ></code>	Document-type Declaration defines the structure of an XML document
<code><![CDATA[]]></code>	CDATA Character Data anything inside a CDATA is ignored by the parser
<code><!-- --></code>	Comment for writing comments

DTD

Document-Type Declaration

The Document-type Declaration identifies the **type** of the document. The *type* indicates the structure of a **valid** document:

- ▶ what elements are allowed to be present
- ▶ how elements can be combined
- ▶ how elements must be ordered

Basically, the DTD specifies what the format allows to do.

Wrapping Up

About XML

About XML

- ▶ designed to store and transfer data
- ▶ designed to be self-descriptive
- ▶ tags are not predefined and can be extended

Characteristics of XML

XML is

- ▶ a generic language that provides structure and syntax for many markup dialects
- ▶ is a syntax or format for defining markup languages
- ▶ a standard for the semantic, hierarchical representation of data
- ▶ provides a general approach for representing all types of information dialects

XML document example

Simple XML

```
<?xml version="1.0"?>
<!DOCTYPE movies>
<movie mins="126" lang="en">
  <!-- this is a comment -->
  <title>Good Will Hunting</title>
  <director>
    <first_name>Gus</first_name>
    <last_name>Van Sant</last_name>
  </director>
  <year>1998</year>
  <genre>drama</genre>
</movie>
```

XML Tree Structure

Each Node can have:

- ▶ a Name
- ▶ any number of attributes
- ▶ optional content
- ▶ other nested elements

Traversing the tree

There's a **unique** path from the root node to any given node