

Text Markup Languages

- XML (Extensible Markup Language): parent language to HTML, SVG and RSS.
- It is a language markup language for documents containing structured information.
- These are structured as elements, attributes, values, and others.

```
<guid isPermaLink="true">http://processing.org/updates.xml</guid>
```

- `guid`: element.
 - `isPermaLink`: attribute.
 - `true`: attribute value.
 - the URL is the 'content value'
- Hierarchy is achieved through nesting.
 - Processing has native support through: `XMLElement`
 - Also support through: `proXML` and `dom4j`.

Real Simple Syndication (RSS)

- A format for delivering regularly changing web content.
- Consists of static information about the feed, and dynamic items.
- Normally news items or blog entries.
- Each item is inside an <item> element and contains among others a title, a url, and a description.
- Feed readers simply ignore elements which they don't recognize
- RSS formatted files are specified using XML.

Typical feed

- Take a look at the feed to see how the data is organized (`tidy`).

```
<channel>
  <item>
    <title>New Tutorial</title>
    <link>http://www.example.com/news/creative-coding.html</link>
    <description>
      A new tutorial is online. It is about lorem ipsum dolores
    </description>
  </item>
  <item>
    ...
  </item>
</channel>
```

RSS in Processing

- To load an RSS feed (or any other XML file) create a new `XMLElement` and provide a URL in the constructor.

```
// Load RSS feed

String url = "http://processing.org/updates.xml";

XMLElement rss = new XMLElement(this, url);

// Get title of each element

XMLElement[] titleXMLElements = rss.getChildren("channel/item/title");

for (int i = 0; i < titleXMLElements.length; i++) {

    String title = titleXMLElements[i].getContent();

    println(i + ": " + title);

}
```

- **Display!!**
(example)

Tutorials

- <http://btk.tillnagel.com/tutorials/rss-feeds-processing.html>
- <http://workshop.evolutionzone.com/2007/09/24/code-read-rss-feed-in-processing>

JSON – Java Script Object Notation

- a valid JavaScript syntax
- <http://www.json.org>
- Looks much like CSS with more quotes.

```
{ "widget": {  
    "debug": "on",  
    "window": {  
        "title": "Keep it real.",  
        "name": "main_window",  
        "width": 500,  
        "height": 500  
    },  
    "image": {  
        "images/killer.png",  
        "name": "icon1",  
        "hOffset": 250,  
        "vOffset": 250,  
        "alignment": "center"  
    }  
}}
```