# XML - Part 1

## STAT 133

Gaston Sanchez

Department of Statistics, UC–Berkeley

gastonsanchez.com
github.com/gastonstat
Course web: gastonsanchez.com/teaching/stat133

# Data Storage

# Motivation

## Various formats

- We don't usually have control over the format in which data is given to us
- Consider the **IBTrACS Data**

  https://www.ncdc.noaa.gov/ibtracs/index.php?name=wmo-data
- IBTrACS available formats: ATCF, CSV, cXML, HURDAT, netCDF, WMO

# IBTrACS Formats

| Format | Data File Access | | |
|---|---|---|---|
| ATCF | All storms (2 MB) | Storms by basin | Storms by year |
| CSV | All storms (2 MB) | Storms by basin | Storms by year |
| cXML | All storms (2 MB) | Storms by basin | Storms by year |
| HURDAT format | All storms (1 MB) | Storms by basin | Storms by hemisphere NH - SH |
| netCDF | All storms (2 MB) | Storms by basin | Storms by year |
| WMO | All storms (2 MB) | Storms by basin | Storms by year |

# IBTrACS Formats

## HURDAT / NOAA Tape Format / TD9697 format

Data from IBTrACS file:*Basin.WP.ibtracs.v03r06.tape* which includes the corection to report 10-min winds.

```
00001 07/18/2006 M= 5  1 SNBR=   1 BERYL       XING=0 SSS=0
00002 07/18*0000000   0    0*0000000   0    0*3232867  30 1010*3302867  35 1008*
00003 07/19*3382865  35 1006*3452863  35 1005*3522864  40 1004*3592865  50 1003*
00004 07/20*3662868  50 1002*3742868  50 1001*3832870  50 1002*3912875  45 1002*
00005 07/21*3982882  45 1003*4102895  45 1000*4242916  40 1002E4382937  35 1000*
00006 07/22E4552967  35 1002E4723000  35 1003E4853035  30 1004*0000000   0    0*
00007 TS SRC=atcf  S/N=2006200N32287
```

---

## ATCF

Data from IBTrACS file: *Year.1969.ibtracs.v03r06.atcf*

```
NA,0,2006071812,00,0,32.30N, 73.30W, 30.0,1010.0,TS
NA,0,2006071818,00,0,33.00N, 73.30W, 35.0,1008.0,TS
NA,0,2006071900,00,0,33.80N, 73.50W, 35.0,1006.0,TS
NA,0,2006071906,00,0,34.50N, 73.70W, 35.0,1005.0,TS
NA,0,2006071912,00,0,35.20N, 73.60W, 40.0,1004.0,TS
NA,0,2006071918,00,0,35.90N, 73.50W, 50.0,1003.0,TS
NA,0,2006072000,00,0,36.60N, 73.20W, 50.0,1002.0,TS
NA,0,2006072006,00,0,37.40N, 73.20W, 50.0,1001.0,TS
NA,0,2006072012,00,0,38.30N, 73.00W, 50.0,1002.0,TS
NA,0,2006072018,00,0,39.10N, 72.50W, 45.0,1002.0,TS
NA,0,2006072100,00,0,39.80N, 71.80W, 45.0,1003.0,TS
NA,0,2006072106,00,0,41.00N, 70.50W, 45.0,1000.0,TS
NA,0,2006072106,45,0,41.30N, 70.10W, 45.0,1000.0,TS
NA,0,2006072112,00,0,42.40N, 68.40W, 40.0,1002.0,TS
NA,0,2006072118,00,0,43.80N, 66.30W, 35.0,1000.0,ET
NA,0,2006072200,00,0,45.50N, 63.30W, 35.0,1002.0,ET
NA,0,2006072206,00,0,47.20N, 60.00W, 35.0,1003.0,ET
NA,0,2006072212,00,0,48.50N, 56.50W, 30.0,1004.0,ET
```

# IBTrACS Formats

## cXML

The following is from *Storm.1969287N11150.ibtracs_wmo.v03r06.cxml*.

```
<?xml version="1.0" encoding="UTF-8"?>
<cxml  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:noNamespaceSchemaLocation="http://www.bom.gov.au/bmrc/projects/THORPEX/CXML/cxml.1.1.xsd">
  <header>
    <product>IBTrACS</product>
    <generatingApplication>
      <applicationType>IBTrACS Merged Analysis</applicationType>
    </generatingApplication>
    <productionCenter>NCDC</productionCenter>
    <moreInfo>http:/www.ncdc.noaa.gov/wdc/ibtracs/</moreInfo>
    <creationTime>2014-08-21T13:48:31</creationTime>
    <missing>-999</missing>
  </header>
  <data type="analysis">
    <disturbance ID="2006071812_323N_733W">
      <cycloneName>BERYL</cycloneName>
      <fix>
        <validTime>2006-07-18T12:00:00</validTime>
        <latitude  units="deg N"> 32.30</latitude>
        <longitude units="deg E"> -73.30</longitude>
        <cycloneData>
          <minimumPressure>
            <pressure units="hPa">1010.0</pressure>
          </minimumPressure>
          <maximumWind>
            <speed units="kt"> 30.0</speed>
            <averagingPeriod units="min">1</averagingPeriod>
          </maximumWind>
        </cycloneData>
      </fix>
```

# Motivation

## Various formats

- There is no overall best storage format
- The correct choice will depend on:
  - the size and complexity of the data set
  - what the data set will be used for
- Every storage format has its strengths and weaknesses

# Storm Tracker

# CSV data

```
2012296N14283,2012,18, NA, MM,SANDY,2012-10-21 18:00:00, DS, 14.30, -77.40, 25.0, 1006.0,atcf,   4.767,  26.983,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-22 00:00:00, DS, 13.90, -77.80, 25.0, 1005.0,atcf,   4.767,  32.120,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-22 06:00:00, DS, 13.50, -78.20, 25.0, 1003.0,atcf,   4.767,  41.712,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-22 12:00:00, TS, 13.10, -78.60, 30.0, 1002.0,atcf,  17.256,  45.083,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-22 18:00:00, TS, 12.70, -78.70, 35.0, 1000.0,atcf,  32.416,  50.576,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-23 00:00:00, TS, 12.60, -78.40, 40.0,  998.0,atcf,  42.517,  57.193,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-23 06:00:00, TS, 12.90, -78.10, 40.0,  998.0,atcf,  42.517,  57.193,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-23 12:00:00, TS, 13.40, -77.90, 40.0,  995.0,atcf,  42.517,  62.912,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-23 18:00:00, TS, 14.00, -77.60, 45.0,  993.0,atcf,  50.140,  67.740,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-24 00:00:00, TS, 14.70, -77.30, 55.0,  990.0,atcf,  64.208,  71.051,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-24 06:00:00, TS, 15.60, -77.10, 60.0,  987.0,atcf,  69.630,  74.924,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-24 12:00:00, TS, 16.60, -76.90, 65.0,  981.0,atcf,  73.788,  81.333,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-24 18:00:00, TS, 17.70, -76.70, 75.0,  972.0,atcf,  81.938,  87.720,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-24 19:00:00, TS, 17.90, -76.60, 75.0,  971.0,atcf,  81.938,  88.215,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-25 00:00:00, TS, 18.90, -76.40, 85.0,  964.0,atcf,  87.354,  91.835,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-25 05:25:00, TS, 20.00, -76.00,100.0,  954.0,atcf,  92.955,  95.040,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-25 06:00:00, TS, 20.10, -76.00,100.0,  954.0,atcf,  92.955,  95.040,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-25 09:00:00, TS, 20.90, -75.70, 95.0,  960.0,atcf,  91.721,  92.945,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-25 12:00:00, TS, 21.70, -75.50, 95.0,  966.0,atcf,  91.721,  90.872,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-25 18:00:00, TS, 23.30, -75.30, 90.0,  963.0,atcf,  89.071,  92.111,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-26 00:00:00, TS, 24.80, -75.90, 75.0,  965.0,atcf,  81.938,  91.220,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-26 06:00:00, TS, 25.70, -76.40, 70.0,  968.0,atcf,  78.805,  90.148,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-26 12:00:00, TS, 26.40, -76.90, 65.0,  970.0,atcf,  73.788,  88.562,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-26 18:00:00, TS, 27.00, -77.20, 65.0,  971.0,atcf,  73.788,  88.215,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-27 00:00:00, TS, 27.50, -77.10, 60.0,  969.0,atcf,  69.630,  89.887,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-27 06:00:00, TS, 28.10, -76.90, 60.0,  968.0,atcf,  69.630,  90.148,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-27 12:00:00, TS, 28.80, -76.50, 70.0,  956.0,atcf,  78.805,  94.455,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-27 18:00:00, TS, 29.70, -75.60, 70.0,  960.0,atcf,  78.805,  92.945,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-28 00:00:00, TS, 30.50, -74.70, 65.0,  960.0,atcf,  73.788,  92.945,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-28 06:00:00, TS, 31.30, -73.90, 65.0,  959.0,atcf,  73.788,  93.787,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-28 12:00:00, TS, 32.00, -73.00, 65.0,  954.0,atcf,  73.788,  95.040,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-28 18:00:00, TS, 32.80, -72.00, 65.0,  952.0,atcf,  73.788,  95.432,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-29 00:00:00, TS, 33.90, -71.00, 70.0,  950.0,atcf,  78.805,  95.874,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-29 06:00:00, TS, 35.30, -70.50, 80.0,  947.0,atcf,  85.083,  96.939,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-29 12:00:00, TS, 36.90, -71.00, 85.0,  945.0,atcf,  87.354,  97.282,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-29 18:00:00, TS, 38.30, -73.20, 80.0,  940.0,atcf,  85.083,  98.086,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-29 21:00:00, ET, 38.80, -74.00, 75.0,  943.0,atcf,  81.938,  97.660,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-29 23:30:00, ET, 39.40, -74.40, 70.0,  945.0,atcf,  78.805,  97.282,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-30 00:00:00, ET, 39.50, -74.50, 70.0,  946.0,atcf,  78.805,  97.078,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-30 06:00:00, ET, 39.90, -76.20, 55.0,  960.0,atcf,  64.208,  92.945,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-30 12:00:00, ET, 40.10, -77.80, 50.0,  978.0,atcf,  58.394,  84.164,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-30 18:00:00, ET, 40.40, -78.90, 40.0,  986.0,atcf,  42.517,  77.007,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-31 00:00:00, ET, 40.70, -79.80, 35.0,  992.0,atcf,  32.416,  68.884,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-31 06:00:00, ET, 41.10, -80.30, 35.0,  993.0,atcf,  32.416,  67.740,main
2012296N14283,2012,18, NA, MM,SANDY,2012-10-31 12:00:00, ET, 41.50, -80.70, 30.0,  995.0,atcf,  17.256,  62.912,main
```

# Some variables in CSV - IBTrACS

| Variables | Variables |
|---|---|
| Serial_Num | Wind(WMO) |
| Season | Pres(WMO) |
| Num | Center |
| Basin | Wind(WMO) Percentile |
| Sub_basin | Pres(WMO) Percentile |
| Name | Track_type N/A |
| ISO_time | Year |
| Nature | # |
| Latitude | BB |
| Longitude | YYYY-MM-DD HH:MM:SS |

# XML

## 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 following lectures

# Motivation

## Two main limitations of field-delimited files

- In plain text formats there is no information to describe the location of the data values
- There is no recognizable label for each data value within the file
- Serious limitations to store data with hierarchical structure

# XML format

## XML advantages

- XML is a storage format that is still based on plain text
- In XML formats every single value is distinctly labeled
- Moreover, every single value is self-described
- The information is organized in a much more sophisticated manner

# XML and HTML

## Why should you 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

# 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

# Markups

## 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>`
- `<cool>`

# About XML

## XML is NOT

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

# About XML

## 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

# Minimalist Example

# XML Example

## Ultra Simple XML

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

# 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>
```

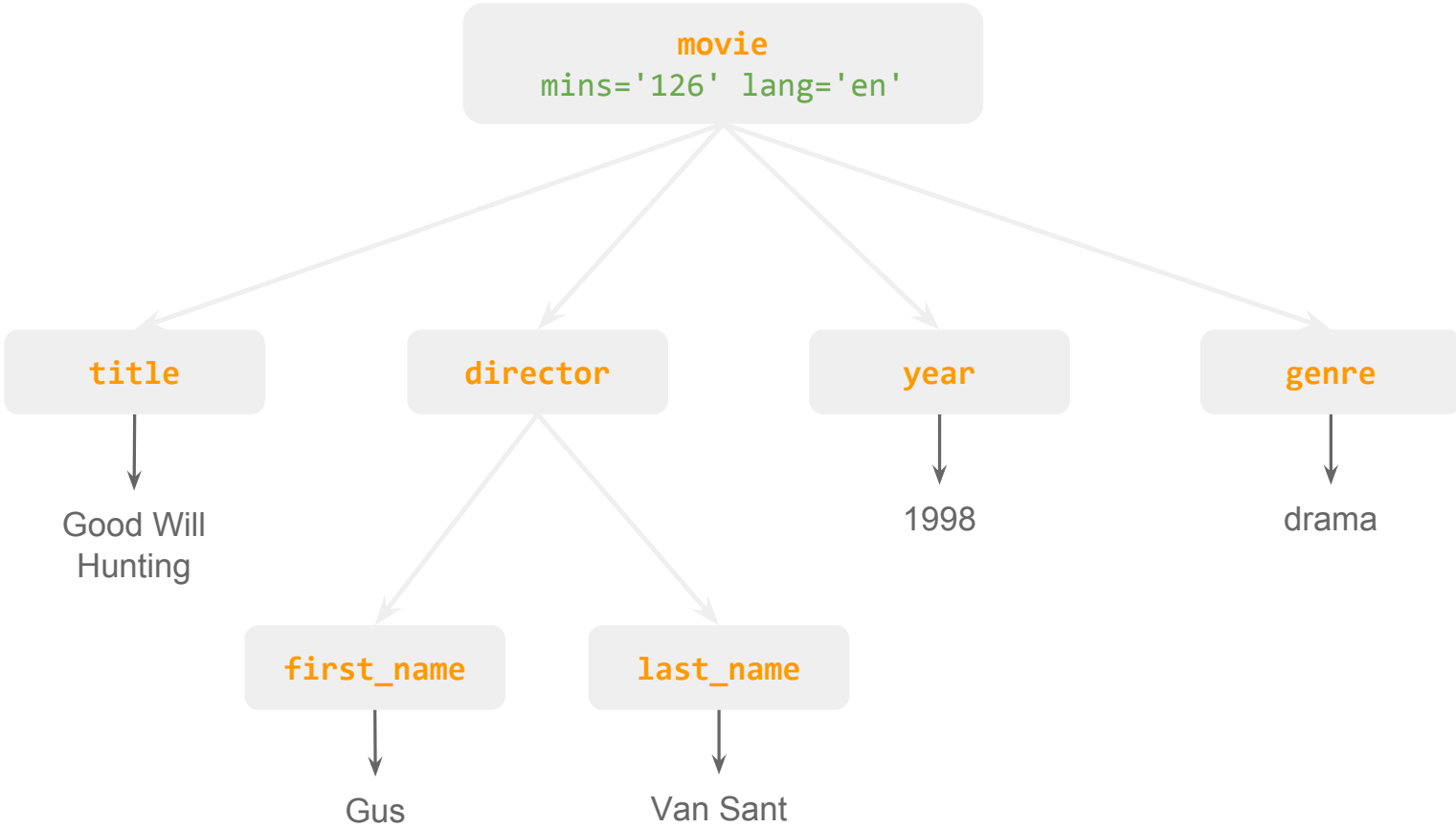
- Now *director* has two child elements: *first_name* and *last_name*

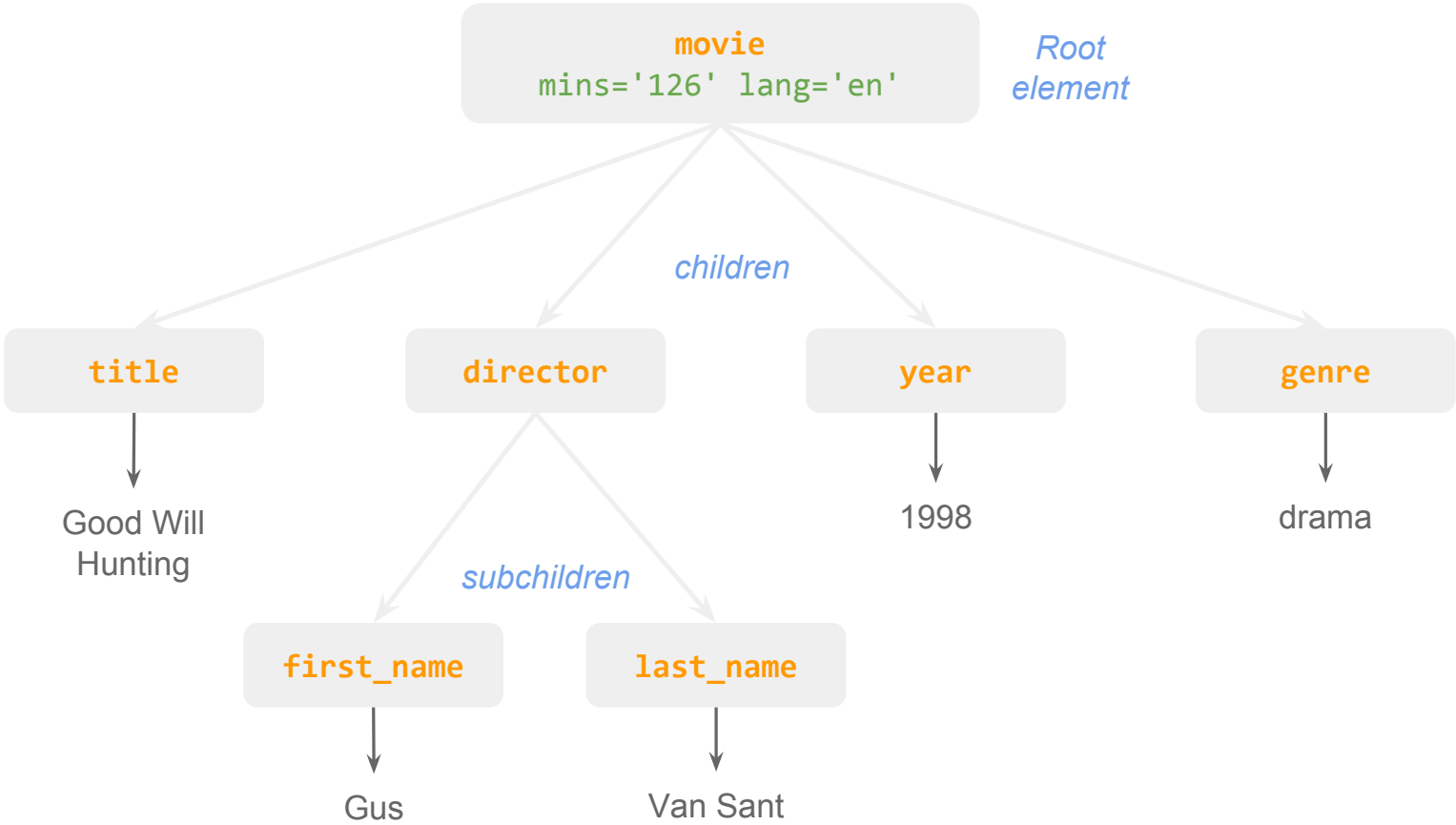
# XML Hierarchy Structure

## 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


movie
mins='126' lang='en'
title
director
year
genre
Good Will Hunting
1998
drama
first_name
last_name
Gus
Van Sant


movie
mins='126' lang='en'
Root element
children
title
director
year
genre
Good Will Hunting
1998
drama
subchildren
first_name
last_name
Gus
Van Sant

# 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

```
<?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 Optional XML Elements

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

## 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

- 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

# 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