HTML - HyperText Markup Language

- Plain text
- Rendered by browser (Firefox, Opera, Camino, Safari, IE)
- Markup is in tags:
  `<p>` for paragraph
  `<img src="myplot.jpg?/>` for bringing in an image.
- Style - Cascading Style Sheets

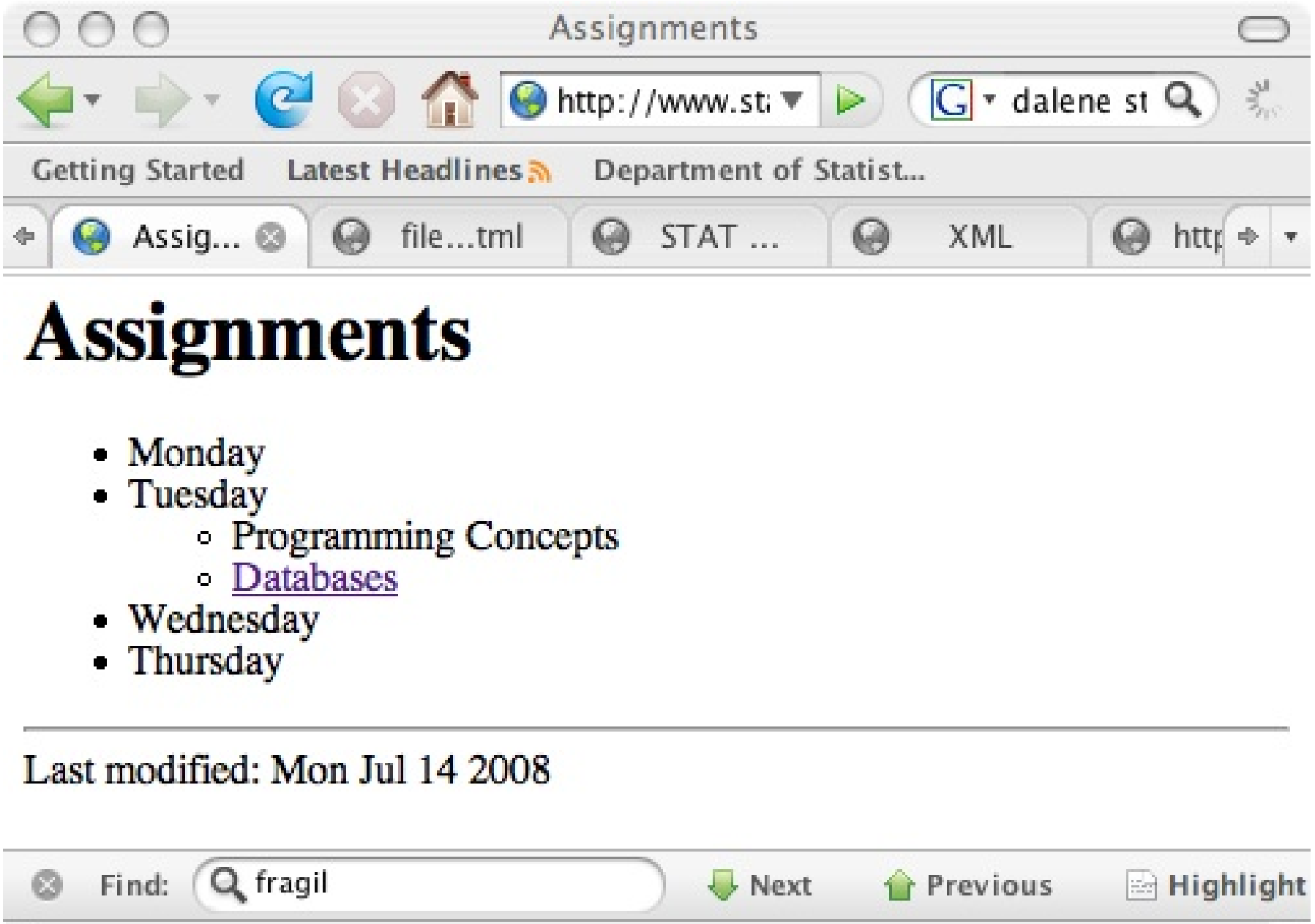
Simple HTML page

http://www.stat.berkeley.edu/users/statcur/Workshop2/Presentations/index.html

```
<html> <head>
<title>Assignments  </title>
<link rel="stylesheet" href="WS.css" />
</head>

<body>
<h1>Assignments </h1>
<ul>
   <li> Monday </li>
   ...  <ul>
           <li> Programming Concepts </li>
           <li> <a href= "RDBMS.pdf">Databases</a> ...
</ul>
<hr>
<!-- Comment -->
Last modified: Mon Jul 14 2008
</body>
</html>
```

Rendered HTML



Data Exchange

ASCII approach:

- Edited with simple text editor
- Natural connection between visual layout and way think about data set
- Simple for applications to read and write

But what about more complex data?

- 999 means missing
- 2.31 is measured in units of mm not inches
- ragged array - think Mannheim

Meta-information

XML is not a markup language

XML is a meta-language facility for defining a markup language. It provides a general framework for supplying meta-information to data.

XML features

- separates content from form
- human readable
- easily generated by machine
- self-describing
- extensible format
- strict parsing rules
- tools shared across discpilines

Climate Science Modeling Language (CSML)

- Layers on top of Geographic Markup Language
- Climate science applications
- Developed by British Atmospheric Data Centre and British Oceanographic Data Centre
- Specialized feature types, e.g.
  - PointFeature - single point measurement, e.g. rain gauge measurement
  - PointSeriesFeature - series of single point measurements, e.g. time series from a tide gauge

```
<gml:featureMember>
  <PointSeriesFeature gml:id="feat02">
    <gml:description>January timeseries of raingauge measurements
    </gml:description>
  <PointSeriesDomain>
    <domainReference>
      <Trajectory srsName="urn:EPSG:geographicCRS:4979">
        <locations>0.1 1.5 25</locations>
        <times frame="#RefSys01"> 1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22 23 24 2 26 27 28 29 30 31
        </times>
      </Trajectory>
    </domainReference>
  </PointSeriesDomain>
  <gml:rangeSet>
    <gml:QuantityList uom="udunits.xml#mm">5 3 10 1 2 8 10 2 5
10 20 21 12 3 5 19 12 23 32 10 8 8 2 0 0 1 5 6 10 17 20
</gml:QuantityList>
</gml:rangeSet>
<parameter xlink:href="#rainfall"/>
</PointSeriesFeature>
</gml:featureMember>
```

Exchange Data

- European Central Bank provides daily exchange rates
- Provides data in several formats including HtML for the iPhone, and 2 XML formats
- XML formats developed by Statistics Data and Metadata Exchange intiative

```
<gesmes:Envelope xmlns:gesmes="http://www.gesmes.org/xml/2002-08-0
      xmlns="http://www.ecb.int/vocabulary/2002-08-01/eurofxref">
  <gesmes:subject>Reference rates</gesmes:subject>
    <gesmes:Sender>
      <gesmes:name>European Central Bank</gesmes:name>
    </gesmes:Sender>
    <Cube>
      <Cube time="2008-04-21">
        <Cube currency="USD" rate="1.5898"/>
        <Cube currency="JPY" rate="164.43"/>
        <Cube currency="BGN" rate="1.9558"/>
        <Cube currency="CZK" rate="25.091"/>
      </Cube>
      <Cube time="2008-04-17">
        <Cube currency="USD" rate="1.5872"/>
        <Cube currency="JPY" rate="162.74"/>
...
</Cube>
</gesmes:Envelope>
```

XML Elements

- Basic unit is an element, aka node or chunk
- Element delimited by tags, `<tag name>`
- Tags open and close the units:

  `<PointSeriesDomain> ..... </PointSeriesDomain>`
- Elements content can be other elements and text content
- Text content is also a node
- Elements must be properly nested
- Elements with no content can collapse the start and end tag

  `<Cube currency="CZK" rate="25.091"/>`

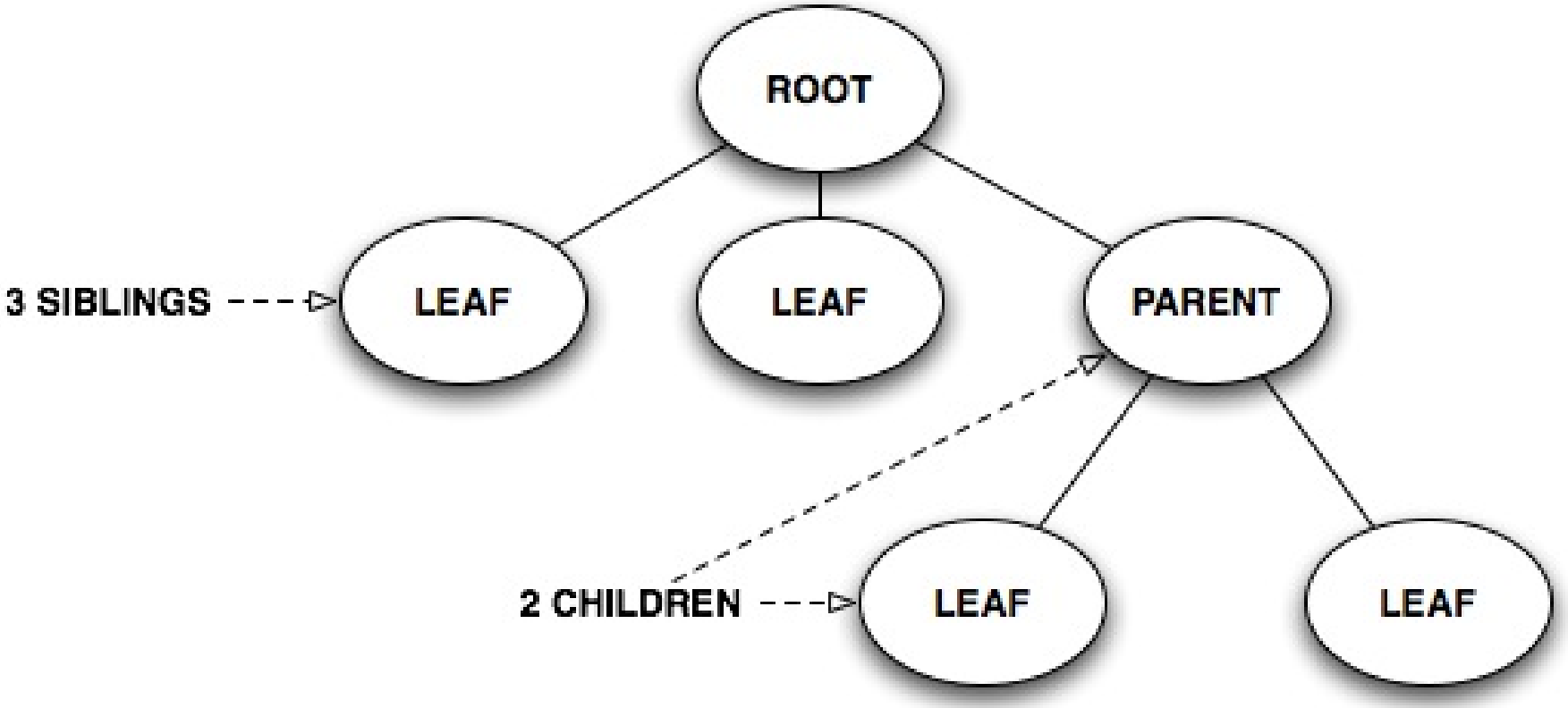
Well-formed XML

- Start and end tag names must match exactly (case-sensitive)
- Elements must nest properly
- Attributes associated with elements appear in name-value pairs,

```
<Cube currency="CZK" rate="25.091"/>
```

- Attribute values must appear in quotes
- Restrictions on tag names and attribute names

XML - Tree



Other Markup

- XML declaration appears outside root element

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

- Processing Instructions, e.g. apply a stylesheet

```
<?xml-stylesheet href="docbook-css-0.3/driver.css"
```

- Comments that are not rendered

```
<!-- This is a comment -->
```

- CDATA - character data that is not processed

```
<![CDATA[
 Good for showing code
]]>
```

- Document-type Declaration - locates schema that describes the application specific grammar

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
```

Generating XML - Example

Data Frame to XHTML Table

```
> chips = read.table("http://www.stat.berkeley.edu/users/nolan/
      stat133/data/Chip.txt", header=TRUE, row.names="Name")
> chips
           Date Transistors Microns Clock speed Data    MIPS
8080       1974         6000    6.00   2.0   MHz    8    0.64
8088       1979        29000    3.00   5.0   MHz   16    0.33
80286      1982       134000    1.50   6.0   MHz   16    1.00
80386      1985       275000    1.50  16.0   MHz   32    5.00
80486      1989      1200000    1.00  25.0   MHz   32   20.00
Pentium    1993      3100000    0.80  60.0   MHz   32  100.00
PentiumII  1997      7500000    0.35 233.0   MHz   32  300.00
PentiumIII 1999      9500000    0.25 450.0   MHz   32  510.00
Pentium4   2000     42000000    0.18   1.5   GHz   32 1700.00
Pentium4x  2004    125000000    0.09   3.6   GHz   32 7000.00
```

```
<table border = "1" cellspacing="2">
  <tr>
    <th/>
    <th>Date</th>
    <th>Transistors</th>
    <th>Microns</th>
    <th>ClockSpeed</th>
    <th>Data</th>
    <th>MIPS</th>
  </tr>
  <tr>
    <th>8080</th>
    <th>1974</th>
    <th>6000</th>
    <th>6</th>
    <th>2</th>
    <th>8</th>
    <th>0.64</th>
  </tr>...
</table>
```

Rendered HTML

	Date	Transistors	Microns	ClockSpeed	Data	MIPS
8080	1974	6000	6	2	8	0.64
8088	1979	29000	3	5	16	0.33
80286	1982	134000	1.5	6	16	1
80386	1985	275000	1.5	16	32	5
80486	1989	1200000	1	25	32	20
Pentium	1993	3100000	0.8	60	32	100
PentiumII	1997	7500000	0.35	233	32	300
PentiumIII	1999	9500000	0.25	450	32	510
Pentium4	2000	42000000	0.18	1500	32	1700

Generating XML -XML Package in R

```
chipHTML = xmlOutputDOM("table",
     attrs = c(border = "1", cellspacing = "3"))

chipHTML$addTag("tr", close = FALSE)
chipHTML$addTag("th")

sapply(names(chips), function(x) chipHTML$addTag("th", x))
     chipHTML$closeTag()

sapply(row.names(chips), function(x) {
   chipHTML$addTag("tr", close = FALSE)
   chipHTML$addTag("td", x)
   sapply(chips[x, ], function(y) chipHTML$addTag("td", y))
   chipHTML$closeTag()
})

chipHTML$closeTag()
```

Demo - Random Walk

- Scalable Vector Graphics
- Animations
- Tutorials:

Demo - Elephant Seal

- Foraging of a tagged female
- Question: Does she travel along a great circle?
- Compare actual path to great circle
- Compare path to simulated random walk on a great circle
- Present results in KML - used by Google Earth
- Tutorial -