

Including meta-data is essential

Tagging

*Lawrence Snyder
University of Washington, Seattle*

Announcements

OED Entry For Byte – Meta-data

byte (balt). *Computers*. [Arbitrary, prob. influenced by bit sb.⁴ and bite sb.] A group of eight consecutive bits operated on as a unit in a computer. **1964** *Blaauw & Brooks* in *IBM Systems Jrnl.* III. 122 An 8-bit unit of information is fundamental to most of the formats [of the System/360]. A consecutive group of *n* such units constitutes a field of length *n*. Fixed-length fields of length one, two, four, and eight are termed bytes, halfwords, words, and double words respectively. **1964** *IBM Jrnl. Res. & Developm.* VIII. 97/1 When a byte of data appears from an I/O device, the CPU is seized, dumped, used and restored. **1967** *P. A. Stark Digital Computer Programming* xix. 351 The normal operations in fixed point are done on four bytes at a time. **1968** *Dataweek* 24 Jan. 1/1 Tape reading and writing is at from 34,160 to 192,000 bytes per second.

<e><hg><hw>byte</hw> <pr><ph>baIt</ph></pr></hg>. <la>Computers</la>.
<etym>Arbitrary, prob. influenced by <xr><x>bit</x></xr> <ps>n.<hm>4</hm></ps>and
<xr><x>bite</x> <ps>n.</ps> </xr></etym> <s4>A group of eight consecutive bits
operated on as a unit in a computer.</s4> <q><q><q>1964</q><a>Blaauw &.
<a>Brooks <bib>in</bib> <w>IBM Systems Jrnl.</w> <lc>III. 122</lc> <qt>An 8-
bit unit of information is fundamental to most of the formats <ed>of the System/
360</ed>.&es.A consecutive group of <i>n</i> such units constitutes a field of
length <i>n</i>.&es.Fixed-length fields of length one, two, four, and eight are
termed bytes, halfwords, words, and double words respectively. </qt></
q><q><q>1964</q> <w>IBM Jrnl. Res. & Developm.</w> <lc>VIII. 97/1</lc>
<qt>When a byte of data appears from an I/O device, the CPU is seized, dumped,
used and restored.</qt></q> <q><q>1967</q> <a>P. A. Stark <w>Digital
Computer Programming</w> <lc>xix. 351</lc> <qt>The normal operations in fixed
point are done on four bytes at a time.</qt></q><q><q>1968</q> <w>Dataweek</w>
<lc>24 Jan. 1/1</lc> <qt>Tape reading and writing is at from 34,160 to 192,000
bytes per second.</qt></q></qp></e>

Meta-Data Describes Data

- Meta-data is data about data ... a description of what the data is
 - Knowing what the data is, as in the OED, allows us to process it better for users
 - Here's an example: Search OED for def of "binary"
 - Without meta-data, get 8,311 hits ... which one is the definition?
 - With meta-data, get each definition in order ... how?
- `<e><hg><hw>binary</hw> ... </hg> ... <e>`

Meta-Data Describes Data

- Meta-data is data about data ... a description of what the data is
 - Knowing what the data is, as in the OED, allows us to process it better for users
 - Here's an example: Search OED for def of "binary"
 - Without meta-data, get 8,311 hits ... which one is the definition?
 - With meta-data, get each definition in order ... how?
- <e><hg><hw>binary</hw> ... </hg> ... <e>

The Principle: We can program computers to better help us if we know what the content is

Give Meta-data Using Tags

- We have seen tags in several forms
 - OED ... custom tags created for dictionary
 - HTML ... the tags “mark up” the content so it can be displayed; it’s a simple use of tagging

```
<html>
  <head>
    <title>All Downhill From Here</title>
  </head>
  <body style="background-color:black;
              face:helvetica; color:white">
    <h1>Downhill Skiing!!! </h1>
    
  </body>
</html>
```

Downhill Skiing!!!



- Recall tag terminology

Meta-data Separation

- Improving on the meta-data of HTML
- Meta-data describes what the data is, but because the tags can be distinguished from the content, it separates itself from the content – that's good
- But HTML combines “identifying content” with “saying how to process it”, i.e. display it
 - Big new idea (not part of HTML)

Separate the content and its tags entirely from the processing – produce a data-only file

The Advantage of Separating

- By separating the content from the processing it is possible to maximize expertise
 - The content expert (you) puts the data together
 - The processing expert (some programmers) write the processing code based on the tags



http://blog.flexcommunity.net/lab/ppv3dGallery_slicedCube/

Photos Rolling Over As Cubes



Just Do It!



Enter The World of XML

- The Extensible Markup Language (XML) is a mark up language in which YOU think up the tags ... it is a self-defining language!
- The usual rules for tags apply
 - Enclose in < and >
 - Start tag `<mynewtag>` and End tag `</mynewtag>`
 - Tags must always be matched or self-terminated
 - Tags can have attributes (think those up, too) of form
`attributename="valueInQuotes"`
 - Use `.xml` as the file extension
 - Always start with "standard text" (shown later)

Example of XML

- Suppose I want to record information about this class; using XML, I might write:

```
<class dept="cse">  
  <catalog qsr="true" credits="5">  
    <num>120</num>  
    <lec len="50" num="3">M, W, F</lec>  
    <lab len="50" num="2"> Tu,Th </lab>  
    <descrip>  
      Must-know computing knowledge for the  
      21st century</descrip>  
  </catalog>  
  <teach>L. Snyder</teach>  
</class>
```

I invented the tags; they make sense to me, and I could write software to process such descriptions

XML Is A Standard

- If you have any digital data, it should be tagged, either by you or by software
- The fact that XML is a standard is powerful ... our example will be “svg”, *scalable vector graphics*
- The idea: specify all 2D graphic figures using XML, and leave the “rendering,” that is, the drawing of the figures to other software
 - Our specifying system will be Inkscape
 - Our rendering system will be processing

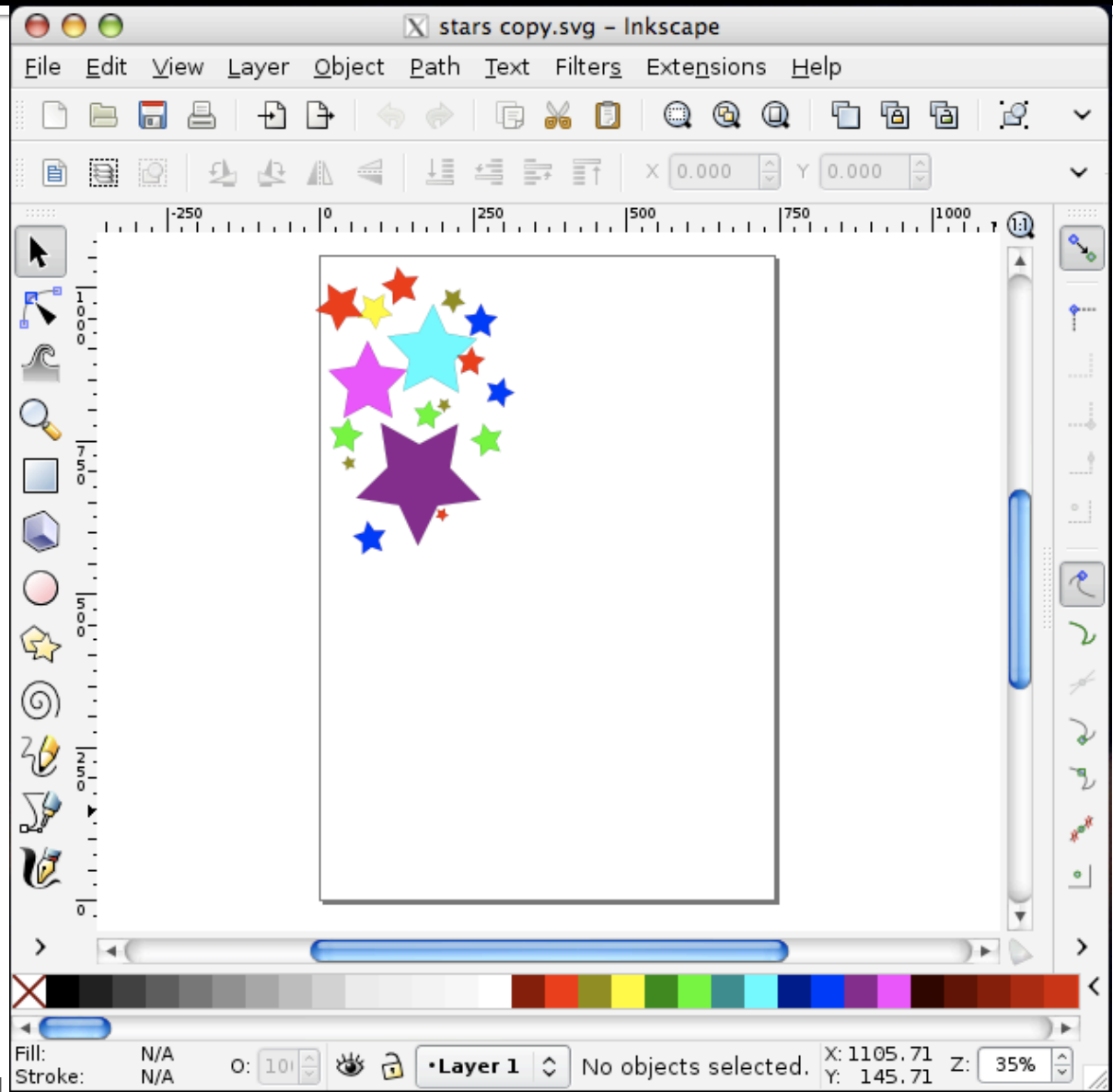
Inkscape

- Inkscape is a lovely, open source system to created svg graphics: <http://inkscape.org/>



Develop Some Graphics

- Install SW and develop some graphics – cake
- Save file as `stars.svg`



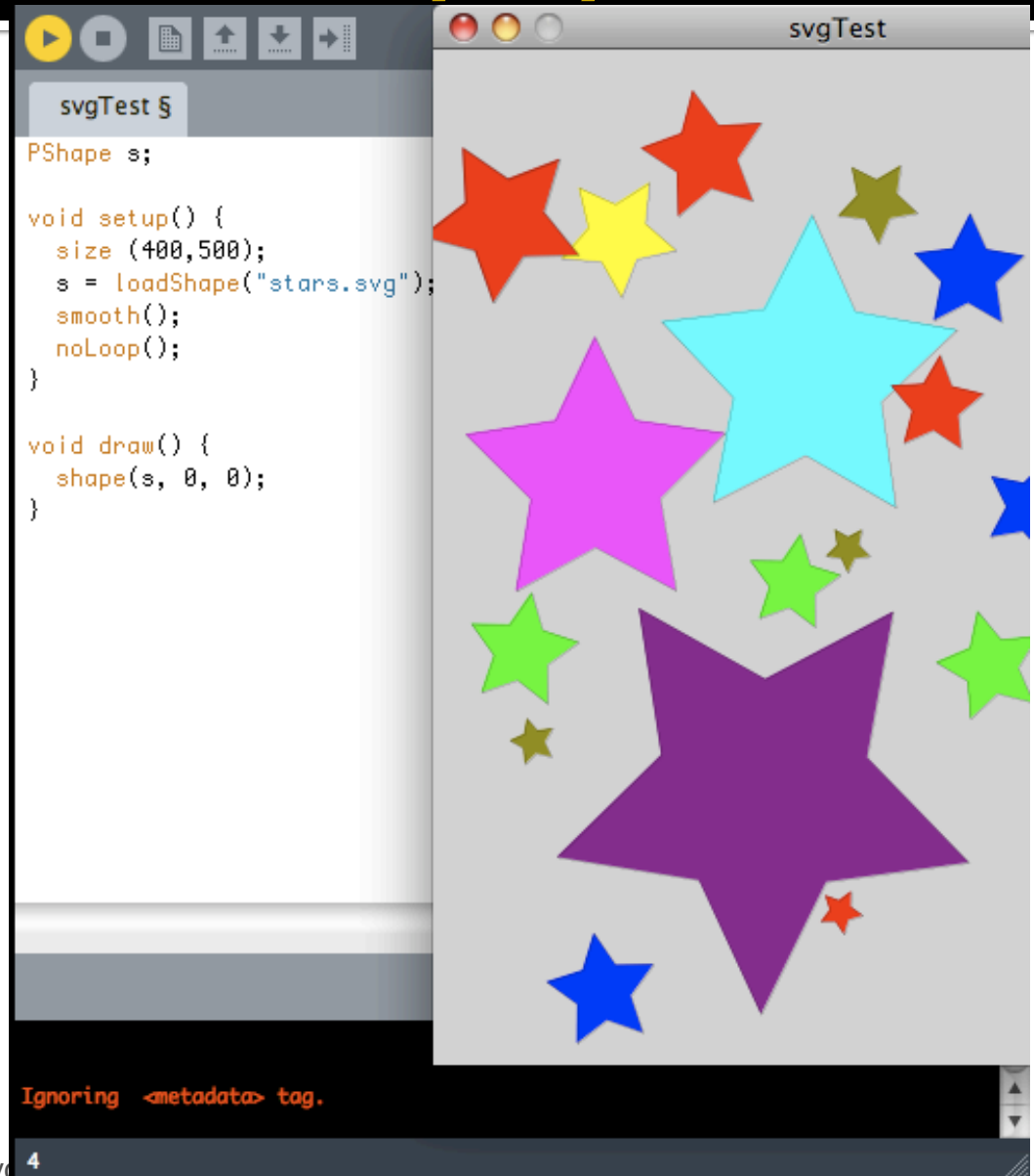
Look At File – Text Editor

- Surprise – it's a (lot) of XML
- We can't understand it, but the SW can ... it's just a lot of attributes and their values

```
<g
  inkscape:label="Layer 1"
  inkscape:groupmode="layer"
  id="layer1">
  <path
    sodipodi:type="star"
    style="fill:#ff00ff;fill-opacity:1;stroke:#000000;stroke-width:0.30000001;stroke-linecap:
    id="path2995"
    sodipodi:sides="5"
    sodipodi:cx="111.42857"
    sodipodi:cy="386.64789"
    sodipodi:r1="106.29186"
    sodipodi:r2="53.145927"
    sodipodi:arg1="0.93804749"
    sodipodi:arg2="1.566366"
    inkscape:flatsided="false"
    inkscape:rounded="0"
    inkscape:randomized="0"
    d="m 174.28572,472.36218 -62.62169,-32.56889 -62.330663,33.12247 11.623681,-6
    inkscape:transform-center-x="0.091857293"
    inkscape:transform-center-y="-6.5198123"
    transform="matrix(0.6312463,0,0,0.65123413,9.4197122,-41.478467)" />
  <path
    sodipodi:type="star"
    style="fill:#800080;fill-opacity:1;stroke:#000000;stroke-width:0.30000001;stroke-lineca
    id="path2997"
    sodipodi:sides="5"
    sodipodi:cx="242.85715"
    sodipodi:cy="623.79077"
    sodipodi:r1="168.95839"
    sodipodi:r2="84.479195"
    sodipodi:arg1="0.3270981"
```

Inkscape Isn't The Only System

- Processing (and lots of graphics systems) understand the SVG primitives and XML let's us structure the graphics



Learning XML

- Since we think up the tags ourselves, it's the easiest language in the world to learn, right?
- Right.
- It's trivial?!
- Not quite ... there is a little technique, and we'll do that now
- Tags can serve in three roles ...

Ways To Use Tags

- **Identity** – tag it so you know what it is

```
<name>George Washington</name>
```

- **Affinity** – all properties of a thing should be collected together

```
<personal>
```

```
  <name>George Washington</name>
```

```
  <height>6' 2"</height>
```

```
  <teeth>Wooden</teeth>
```

```
  <home>Mount Vernon</home>
```

```
</personal>
```

Ways To Use Tags (continued)

- **Collection** – enclose a group of items of the same type in a collective tag

```
<presidents>  
  <prez num="1"><personal><name>George ...  
  <prez num="2"><personal><name>John ...  
  <prez num="3"><personal><name>Thomas ...  
  ...  
  <prez num="44"><personal><name>Barack ...  
</presidents>
```

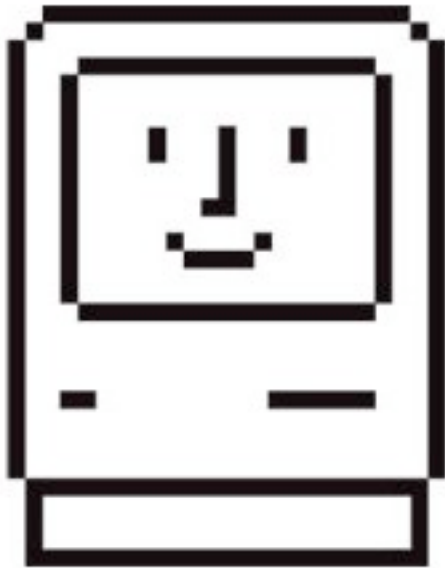
- These uses become intuitive quickly

Example: Classify The Uses

```
- <travels>
  - <visit>
    <sight>Washington State</sight>
    - <action flag="wash.gif">
      The State of Washington is a fun place to visit. We toured Spokane,
      Grand Coulee Dam, Seattle's Space Needle and Mt. Rainier, which
      wasn't rainy at all, but beautiful in the sun!
    </action>
  </visit>
  - <visit>
    <sight>Oregon</sight>
    - <action flag="oregon.jpg">
      South of Washington is Oregon. It is at the end of the old Oregon
      Trail. It is an unusual place. First, the University of Oregon's team is
      called the Ducks. Also, Mt. Bachelor is near the Sisters; with so
      many women around, why is it still a bachelor?
    </action>
  </visit>
  - <visit>
    <sight>California</sight>
    - <action flag="california.png">
      California seems to be a republic, but not a banana republic. More
      like an orange republic. We visited San Francisco, San Quentin, the
      Monterey Bay Aquarium, LA and Hollywood. We didn't see any
      stars, but we were not there in the dark either.
    </action>
  </visit>
</travels>
```

Describe The Mac

- Using XML, give a description of the Mac



Display XML with Browser

- We can see the structure of our XML (and check that it is well formed) by displaying it in Firefox

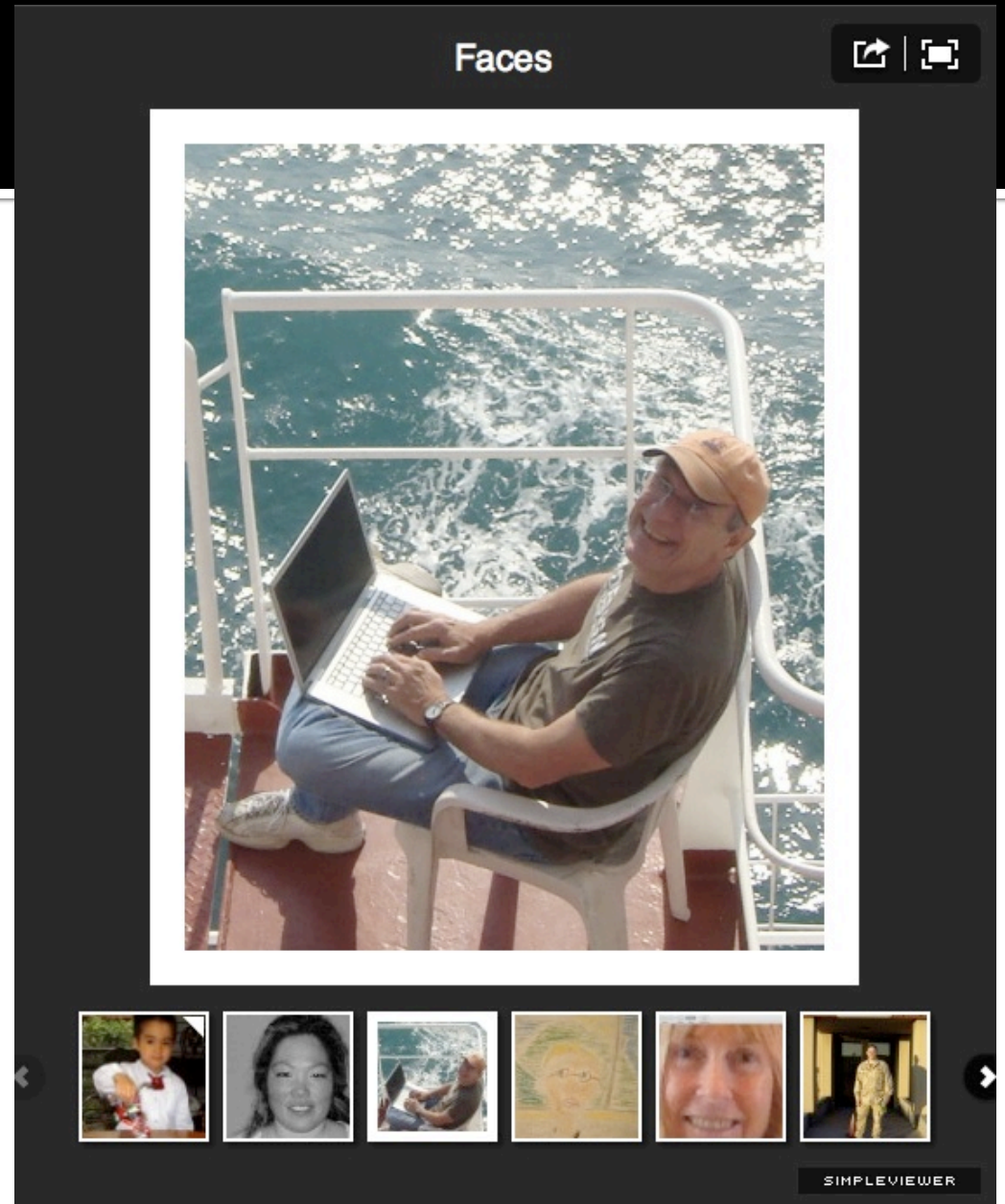
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
-<mac>
-  <chassis>
-    <extern>
      <!-- left side -->
      <line frx=".000" fry=".000" tox=".000" toy=".000"/>
      <dot x=".000" y=".000"/>
      <!-- top side -->
      <line frx=".000" fry=".000" tox=".000" toy=".000"/>
      <dot x=".000" y=".000"/>
      <!-- right side -->
      <line frx=".000" fry=".000" tox=".000" toy=".000"/>
    </extern>
-  <face>
-    <extern>
      <!-- l/r sides -->
      <paralline frx=".000" fry=".000" tox=".000" toy=".000" space=".000"/>
      <!-- t/b sides -->
      <paralline frx=".000" fry=".000" tox=".000" toy=".000" space=".000"/>
    </extern>
  
```

© 2011 Larry Orydel, OOL

Assignment

- Part 1: Set up a picture viewer on your own computer of your own pix



- <http://www.simpleviewer.net/simpleviewer/>

Part II: The Personal Online Diary

■ Build A Diary Using XML

MOD My Online Diary

This is my diary of the Cool Stuff that happens each day



Links

[NPR Hourly News](#)

[My CSP Page](#)

[CS Principles](#)

[Astronomy Picture
of the Day](#)

Sunday, February 27, 2011

Spring Break Is SET!

Rod's uncle is going to loan us his condo for the week of Spring Break. So, it's Hawaii, here we come. Now all I need is airfare, and the prices are crazy! :(



Part III: Improve/Customize Page

- The browser displays page, but the SW given is a little lame and a little limited! Just Fix It!

```
<entry>
  <day>Sunday, February 27, 2011</day>
  <happenin>
    Spring Break Is SET!<bline/><bline/>
    Rod's uncle is going to loan us his condo for the week of Spring Break. So, it's Hawaii, here we
    come. Now all I need is airfare, and the prices are crazy! :( <bline/><bline/>
    <pic name="im/hawaii.jpg" expl="Hawaiian Sunset" width="600"/><bline/><bline/>
  </happenin>
</entry>
<entry>
  <day>Sunday, February 27, 2011</day>
  <happenin>
    <stress>Spring Break Is SET!</stress>
    Rod's uncle is going to loan us his condo for the week of Spring Break. So, it's Hawaii, here we
    come. Now all I need is airfare, and the prices are crazy! :( <bline/><bline/>
    <pic name="im/hawaii.jpg" expl="Hawaiian Sunset" width="600"/><bline/><bline/>
  </happenin>
</entry>
```

Sunday, February 27, 2011

Spring Break Is SET!