Elements and Attributes

Non-Empty Elements

- Remember in XML you can make whatever elements you like
  - Case sensitive
  - Must begin with a slash or an underscore
  - No blank spaces
  - Can’t begin with xml
  - Opening must match closing
  - Every element must be contained within the opening and closing root element

```
<name>Tiger</name>
```

Nesting Elements

- Remember elements can contain text or other elements
- Nesting elements allow you to establish hierarchical relationships with your data
- Each nested element must be completely nested in another

```
1  <?xml version="1.0" ?>
2  <superheroes>
3      <superhero>
4          <name>Superman</name>
5          <alterEgo>Clark Kent</alterEgo>
6      </superhero>
7
8  </superheroes>
```

Attributes

Must begin with a letter or an underscore

No Spaces

Not begin with xml

Can only appear once within an element

Attributes are known as “name-value” pairs

Placed in the opening tag of an element

Generally considered meta data (data about the data)

Element can have multiple attributes as long as they are unique

MUST be in quotes – double or single

Attributes

- If a value contains double quotes – use single quotes to contain the value
  - `comments = 'She yelled, "Help Superman!"'`

```
1  <?xml version="1.0" ?>
2  <superheroes>
3      <superhero>
4          <name>Superman</name>
5          <alterEgo>Clark Kent</alterEgo>
6          <height units="feet">6.25</height>
7      </superhero>
8  </superheroes>
```

Why Use Attributes

- They tend to simplify your markup – you don't need to worry about proper nesting
- `<name nickname='Shiny John'></name>`
- Vs.
- `<name>`
- `<nickname> Shiny John</nickname>`
- `</name>`

Why use Attributes?

- Elements can make your markup bulky and difficult to read
- `<note>`
- `<type>Informational</type>`
- `This is a note.`
- `</note>`
- Vs.
- `<note type="informational">This is a note</note>`

When not to use Attributes?

- Attributes are unordered – so if order matters use elements

- Elements can be more complex
  - Attributes only allow for simple text as a value

Is this Valid?

```
<input checked = true>
```

Attributes must be quoted

Is this Valid?

```
<input checked = 'true'>
```

Attributes must be quoted

Is this Valid?

```
<input checked = “true’>
```

Attributes must be quoted and the quotes must match

Is this Valid?

```
<input size='11px' size = '2em'>
```

No two attributes in a given element can share the same name

Is this Valid?

```
<name>Superman</ alias = “Clark Kent” name>
```

Attributes are attached to the start-tag

Empty Elements

- Do not have content
- The attributes store data for the element
- Has a single element that is self-closing

```
<main_image file="dog.jpg" />
```

OR

```
<main_image file="dog.jpg"></main_image>
```

Comments

- Useful to annotate XML documents
- As with HTML, comments are not parsed by the processor

```
<!- update January 2014 -->
```

- No nesting of comments
- Can span multiple lines
- Hide sections during debugging (commenting out)
- Can contain spaces, text, elements
- Cannot nest a comment INSIDE an element
- Cannot use a – inside a comment

Is this Valid?

```
<!-- Did you know that Batman is -- Bruce Wayne -->
```

Can not have the double-dash inside a comment

Is this Valid?

```
<name <!- this is the superhero's name --> >Superman</ name>
```

Comments cannot be nested inside of element tags

Special Characters

Work the same as in html – character references are used for non standard characters

There are only 5 predefined entities:

- `<` represents "<"
- `>` represents ">"
- `&` represents "&"
- `&apos;` represents '
- `&quot;` represents “

Use within the text value of your XML document

- Any other entities that are used have to be pre-defined in a DTD

Text Characters

- Xml documents consist only of text characters
- Falls into three categories:
  - Parsed character data (PCDATA):
    - All the characters that XML treats as parts of markup tags
  - Character data (CDATA):
    - All that remains when pcdata is removed – this information is NOT parsed
  - White space

CDATA Section

- `<![CDATA[ ……..]]`
- Tells the parser to not parse the information
- Must be contained within the root element
- Can not have nested CDATA
- Special Symbols not required.
- Often used for large blocks of text or if you want to protect specific code from being processed by the XML processor

CDATA Section

```
<![CDATA[
Any information included here will be treated as text.
&amp; will display as &amp;
]]>
```

```
– <superheroes>
  – <superhero>
      <name>Superman</name>
      <alterEgo>Clark Kent</alterEgo>
      <height units="feet">6.25</height>
    </superhero>
    Any information included here will be treated as text. &amp; will display as &amp;
  </superheroes>
```