

XML}

XML

- { What is XML
- { Basic Rules
- { Writing XML
- { Sample
- { Validation



XML



XML

Extensible Markup Language. A flexible language used for storing information for various applications.

XML

XML is a W3C standard.

<http://www.w3.org/XML/>

XML is EVERYWHERE!

- { The internet
- { Mobile devices
- { Desktop applications
- { Television
- { Widgets
- { Electronic Signage



Basic Rules

XML Basic Rules

1. Open a tag close a tag
2. Must always have a root element
3. Open a quote close a quote
4. Elements are lower case

Writing XML



Writing XML

- { Starting the Document
- { Root Node
- { Elements (tags)
- { Information
- { Nesting
- { Attributes



Starting the Document

Starting the Document

```
<?xml version="1.0" encoding="utf-8"?>
```

Starting the Document

```
<?xml version="1.0" encoding="utf-8"?>
```

**This goes at the
top of every
xml document**

Starting the Document

```
<?xml version="1.0" encoding="utf-8"?>
```

**The version tells the browser how to interpret the xml.
A form of validation.**

Starting the Document

```
<?xml version="1.0" encoding="utf-8"?>
```

The encoding tells the browser how to interpret the xml as content.

Starting the Document

```
<?xml version="1.0" encoding="utf-8"?>
```

**utf-8 is an
8bit unicode
text format.
(universal)**



Root Node



Root Node

Every xml document must have one root node. This is basically one element, or tag, that wraps around all other tags.

Root Node

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Single Ladies</name>  
  </song>  
</songs>
```

Root Node

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Sing  
  </song>  
</songs>
```

**The <songs>
element wraps
around all other
elements**



Elements (tags)

XML Elements

The biggest advantage of xml is that the elements are defined by you.

Root Node

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Sing  
  </song>  
</songs>
```

**<songs> was
invented by me.**

Root Node

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Single Ladies</name>  
  </song>  
</songs>
```

**<name> and <song>
were invented
by me too.**



Information

Information

All of the data, aka information, is placed in between the elements.

Root Node

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Single Ladies</name>  
  </song>  
</songs>
```

**The element should
semantically
describe the data**

Root Node

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Single Ladies</name>  
  </song>  
</songs>
```

**Single Ladies
is the ...**

Root Node

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Single Ladies</name>  
  </song>  
</songs>
```

...name...

Root Node

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Single Ladies</name>  
  </song>  
</songs>
```

...of the song.



Nesting



Nesting

Nesting is the idea of placing elements within other elements to group data.

Nesting

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Single Ladies</name>  
  </song>  
</songs>
```

Nesting

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Single Ladies</name>  
  </song>  
</songs>
```

**<song> is nested
inside of <songs>**

Nesting

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song>  
    <name>Single Ladies</name>  
  </song>  
</songs>
```

**<name> is nested
inside of <song>**



Nesting: More Data

Nesting: More Data

```
<?xml version="1.0" encoding="utf-8"?>
<songs>
  <song>
    <name>Single Ladies</name>
  </song>
  <song>
    <name>I Saw the Sign</name>
  </song>
</songs>
```

Nesting: More I

The new <song> element is a sibling of the previous <song> element/

```
<?xml version="1.0" >
<songs>
  <song>
    <name>Single Ladies</name>
  </song>
  <song>
    <name>I Saw the Sign</name>
  </song>
</songs>
```

Nesting: More I

Note the <songs> is the root. There can be only one.

```
<?xml version="1.0" >
<songs>
  <song>
    <name>Single Ladies</name>
  </song>
  <song>
    <name>I Saw the Sign</name>
  </song>
</songs>
```

How can you add another song?

Nesting: More Data

```
<?xml version="1.0" encoding="utf-8"?>
<songs>
  <song>
    <name>Single Ladies</name>
  </song>
  <song>
    <name>I Saw the Sign</name>
  </song>
  <song>
    <name>Ice Ice Baby</name>
  </song>
</songs>
```



Attributes

Attributes

Another way to store data. Typically they store data about the element.

Attributes

```
<?xml version="1.0" encoding="utf-8"?>  
<songs>  
  <song id="1" length="3:15">  
    <name>Single Ladies</name>  
  </song>  
</songs>
```

Attributes

```
<?xml version="1.0" encoding="utf-8"?>
<songs>
  <song id="1" length="3:15">
    <name>Single Ladies</name>
  </song>
</songs>
```

**Attributes are
inside the
element (tag).**

Attributes

```
<?xml version="1.0" encoding="utf-8"?>
<songs>
  <song id="1" length="3:15">
    <name>Single Ladies</name>
  </song>
</songs>
```

**Attributes describe
the element (tag).**

Attributes

```
<?xml version="1.0" encoding="utf-8"?>
<songs>
  <song id="1" length="3:15">
    <name>Single Ladies</name>
  </song>
</songs>
```

Attributes must be lowercase and values must always appear in quotes.



Sample



Sample

```
<?xml version="1.0" encoding="utf-8"?>
<songs>
  <song id="1" length="3:18">
    <name>Single Ladies</name>
    <artist>Beyonce Noles</artist>
  </song>
  <song id="2" length="3:08">
    <name>I saw the Sign</name>
    <artist>Ace of Base</artist>
  </song>
</songs>
```



Validation



Validation

- { Dreamweaver
- { <http://validator.w3.org>
- { Check in your browser

XML: Summary

- { What is XML
- { Basic Rules
- { Writing XML
- { Sample
- { Validation