

**xhtml**

**X-Treme Markup!!!**

# Interview Question:

Why should we use xhtml instead of html?

# OK Answer:

“It’s the latest and greatest.”

# Better Answer:

“It’s XML compliant”

# Best Answer:

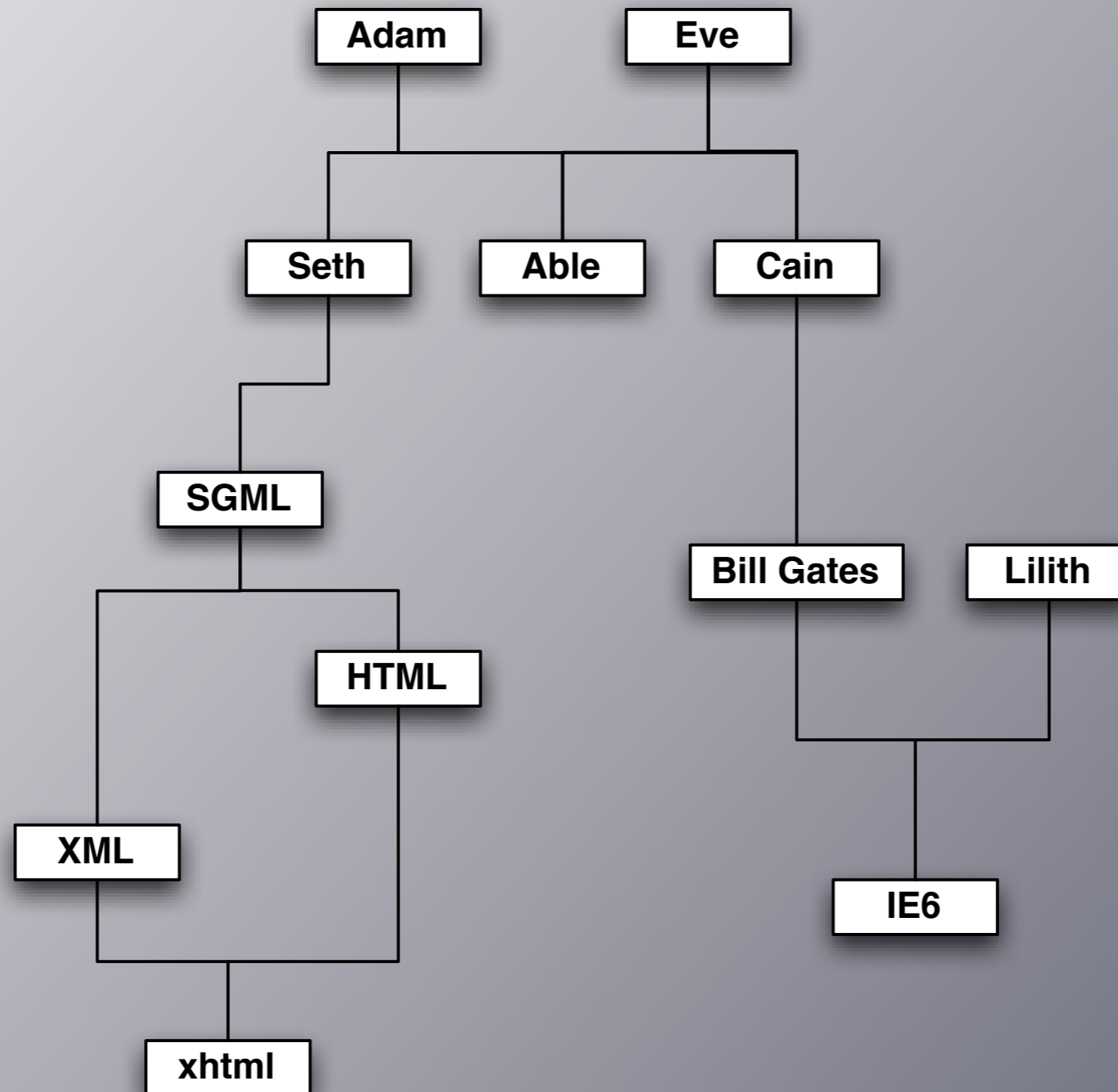
I spent most of 2002 learning this one, and it's going to take the next 30 minutes or so to share it with you.

# Why X?

X is cool and mysterious; an unknown quantity.

Racer **X** -Files  
-Men

# Markup Genealogy



# Why X?

e**X**tensible **M**arkup **L**anguage

# Why X?

xml + html = **xhtml**

# Why X?

**xhtml is XML-compliant HTML**

Ok...

So what?

What's so great about  
XML, anyway?

# XML is...

**Flexible** - brew your own schema!

# XML is...

**eXtensible** - namespaces allow tags from different schemas to exist side-by-side in the same document.

# XML is...

**Self-validating** - schema languages like DTD and XSD let you define your tags, their ordering, what types of data they hold, etc.

# XML is...

**Self-descriptive** - your document describes itself, and consumers know what your data is.

# XML is...

A W3C recommendation

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

# XML Anatomy

```
<?xml version="1.0" encoding="utf-8"?>  
<!DOCTYPE root SYSTEM "somefile.dtd">  
<root>  
    <child>...</child>  
</root>
```

# XML Anatomy

XML Declaration - processing instruction to declare the document as XML

```
<?xml version="1.0" encoding="utf-8"?>  
<!DOCTYPE root SYSTEM "somefile.dtd">  
<root>  
    <child>...</child>  
</root>
```

# (Quick Aside)

Many of you will recognize the following popular XML processing instruction:

**<?php**

# XML Anatomy

Doctype - wait a minute, this looks familiar!

```
<?xml version="1.0" encoding="utf-8"?>  
<!DOCTYPE root SYSTEM "somefile.dtd">  
<root>  
    <child>...</child>  
</root>
```

# XML Anatomy

## Tags

- `<element>data</element>`
- `<element />`
- `<element><child /><element>`

# XML Anatomy

## Attributes

- `<element attribute="value" />`
- `<element attribute='value' />`

xhtml is essentially html  
with terminated tags  
and quoted attributes.

# Ok...

I still don't see why I should care if my html is  
xml-compliant.

# XML gives us...

- A **D**ocument **O**bject **M**odel (DOM)
- JavaScript (or ECMAScript, to be P.C.)
- **C**ascading **S**tyle **S**heets

# Big Flippin' Deal

# XML gives us...

**True** semantic markup!

# Semantic Markup

Microformats are good:

```
<span class="fn">George</span>
```

XML is better:

```
<firstName>George</firstName>
```

# XML gives us...

**XPath** - a super-slick way to find information in a document (way better than DOM or CSS selectors)

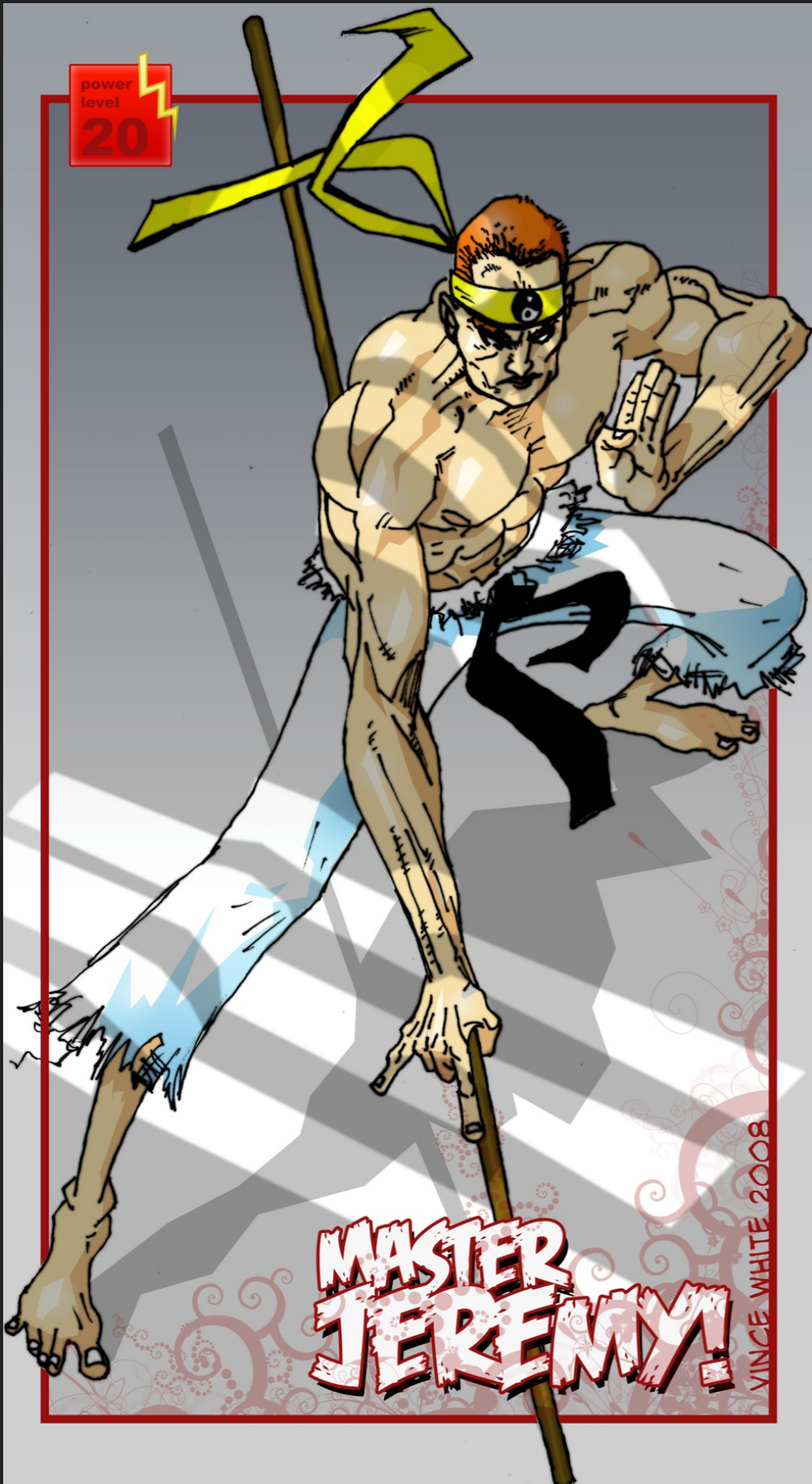
# Basic XPath example

`//person/@id`

```
<people>  
  <person id="1">Wayne</person>  
  <person id="2">Garth</person>  
</people>
```

# Dude...

I can totally do that with perl or grep.



## STATS:

STR: 8      INT: 10  
SPD: 8      CHR: 7  
AGL: 7      XML: 8

## SKILLS:

LONG BLADE: 25      PHP: 75  
SHORT BLADE: 15      REGEX: 65  
LONG STICK: 70      SQL: 70  
SHORT STICK: 65

## AFFILIATIONS:

DOMINION ENTERPRISES  
NORFOLK WSUUG  
WHITE LOTUS CLAN

# Advanced XPath Example

# //fName[text()="Wayne"]/..//address[1]/street

```
<people>
  <person id="1">
    <fName>Wayne</fName>
    <lName>Campbell</lName>
    <phone>802-123-4567</phone>
    <addresses>
      <address>
        <street>200 Gameon St</street>
        <city>Aurora</city>
        <state>IL</state>
      </address>
      <address>
        <street>1916 Asphinctersayswhat Ln</street>
        <city>Aurora</city>
        <state>IL</state>
      </address>
    </addresses>
  </person>
  <person id="2">
    <fName>Garth</fName>
    <lName>Algar</lName>
    <phone>802-987-6543</phone>
    <addresses>
      <address>
        <street>302 Partyon Plaza</street>
        <city>Aurora</city>
        <state>IL</state>
      </address>
    </addresses>
  </person>
</people>
```

# XML Applications

- xhtml
- RSS/Atom
- DocBook
- OpenDocument
- XUL
- SOAP (and other web services)
- SVG
- VoiceXML
- XSL:FO

XML also gives us...

**XSLT**

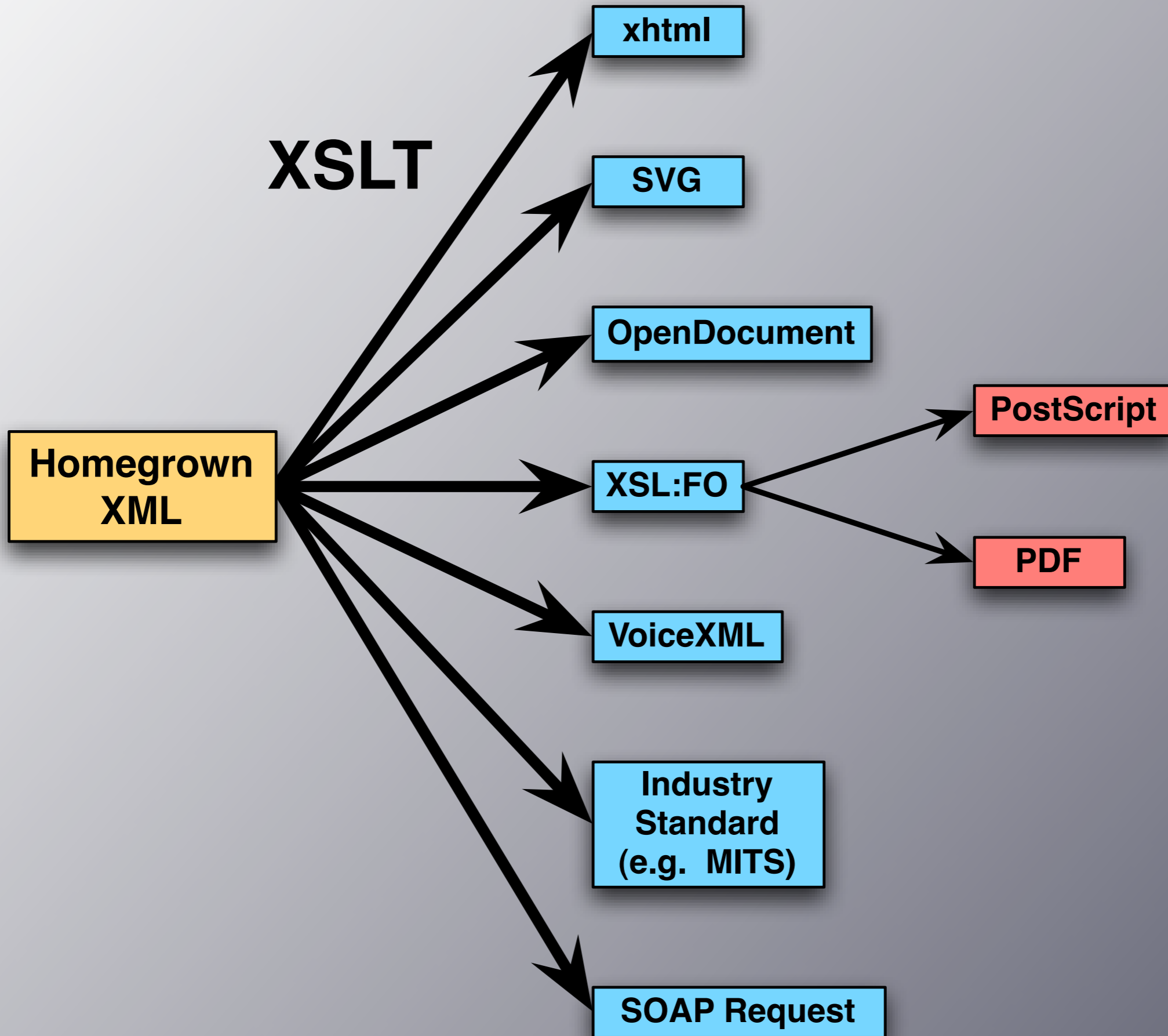
**X**ML **S**tylesheet **L**anguage **T**ransformations

# XSLT

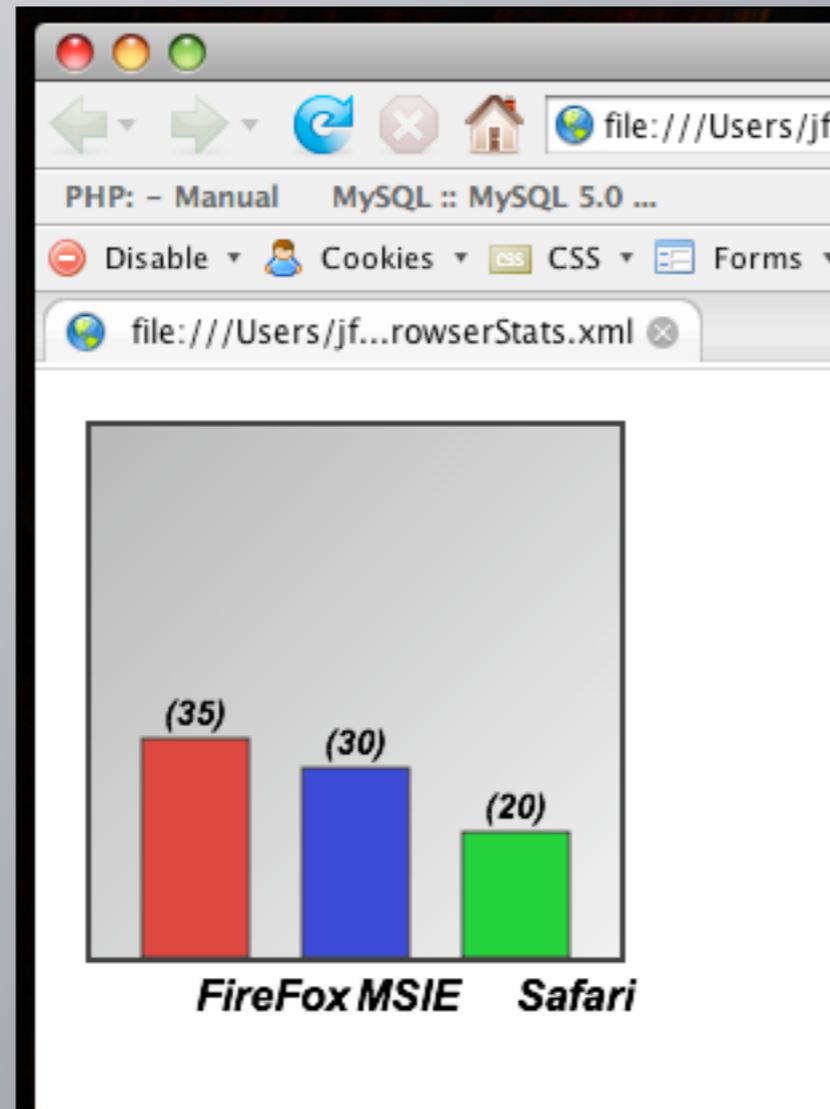
- Style sheets for XML
- Transforms data from one schema to another
- True separation of data from presentation
- Are written in XML

# XSLT

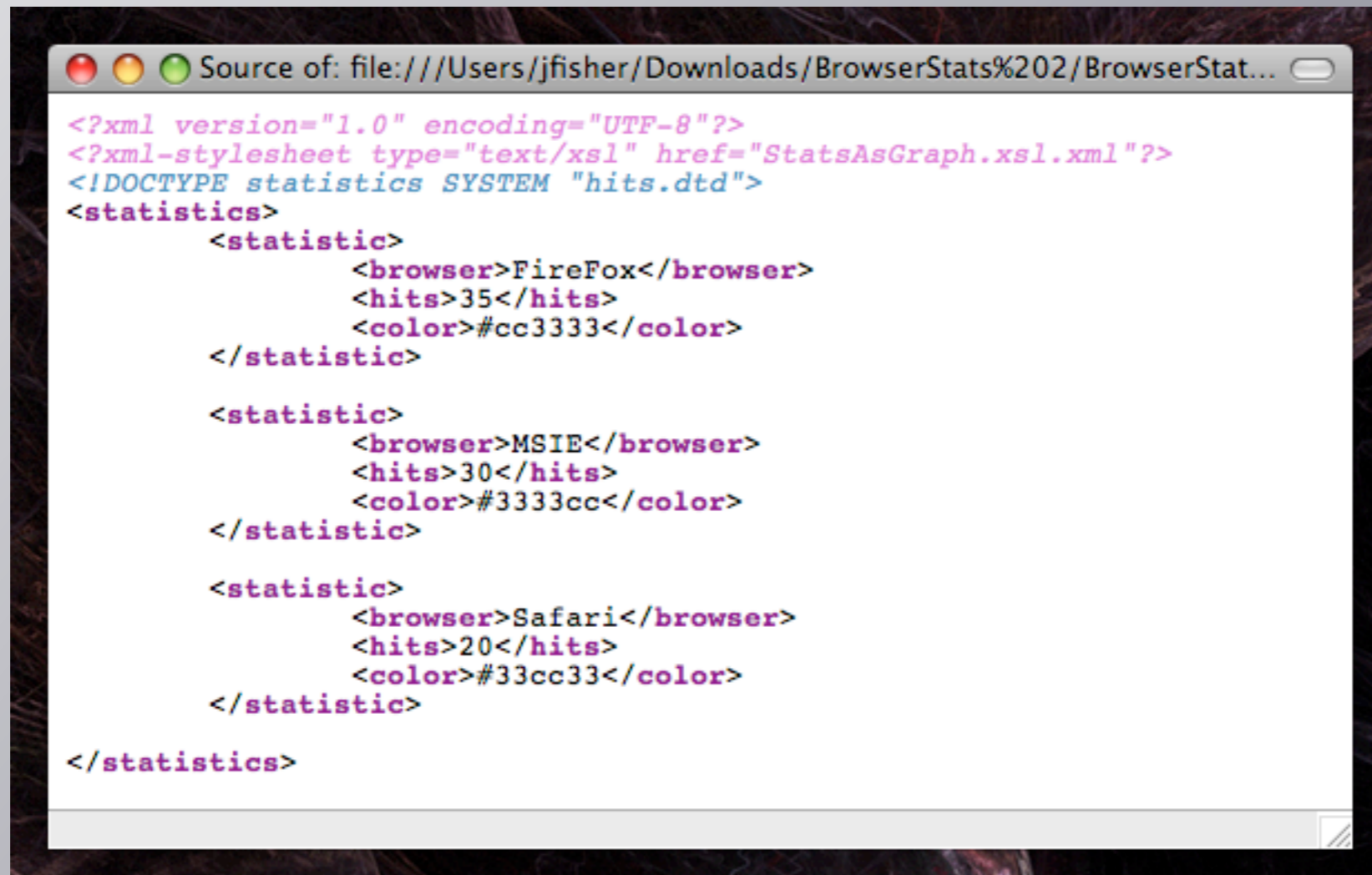
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:variable name="totalHits" select="sum(//hits)" />
  <xsl:template match="/">
    <svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      version="1.1">
      <g>
        <defs>
          <linearGradient id="LightGrey" x1="0%" y1="0%" x2="100%" y2="100%">
            <stop offset="0%" stop-color="#aaaaaa" />
            <stop offset="100%" stop-color="#eeeeee" />
          </linearGradient>
        </defs>
        <rect stroke-width="2" stroke="#333333" fill="url(#LightGrey)">
          <xsl:attribute name="x">
            <xsl:value-of select="$hOffset" />
          </xsl:attribute>
        </rect>
      </g>
    </svg>
  </template>
</xsl:stylesheet>
```



# A Slick Example (Browser)



# A Slick Example (Source)



```
Source of: file:///Users/jfisher/Downloads/BrowserStats%202/BrowserStat...
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="StatsAsGraph.xsl.xml"?>
<!DOCTYPE statistics SYSTEM "hits.dtd">
<statistics>
  <statistic>
    <browser>FireFox</browser>
    <hits>35</hits>
    <color>#cc3333</color>
  </statistic>

  <statistic>
    <browser>MSIE</browser>
    <hits>30</hits>
    <color>#3333cc</color>
  </statistic>

  <statistic>
    <browser>Safari</browser>
    <hits>20</hits>
    <color>#33cc33</color>
  </statistic>
</statistics>
```

**Holy crap! I can take  
my data anywhere!**

# In Answer to My Question

- HTML is a *document* language
- XML is a *data* language
- xhtml, by being an XML application, lets us create *documents* (and news feeds, graphics, sounds, web service requests, etc.) from *data*, taking advantage of all that XML has to offer

**That's all, folks!**