IT452 Advanced Web and Internet Systems

Set 8: XML, XPath, and XSLT

(Chapter 15.1-4,15.8)

Some XSLT examples derived from prior textbook: “Professional Web 2.0 Programming”

Why use XML?

1. Provides a well-defined structure for communication.
2. The client knows the exact format that it will receive.
3. The client can **formally verify** that the received data conforms to the agreed format.
4. The order of data elements doesn’t matter.

```
<book>
  <title>Moby Dick</title>
  <author>Herman Melville</author>
  <year>1851</year>
</book>
```

```
<book>
 <year>1851</year><title>Moby Dick</title>
 <author>Herman Melville</author>
</book>
```

3

You Already Know…

1. What XML is.
   - One root element
   - Tree structure

2. How to send XML from the server.

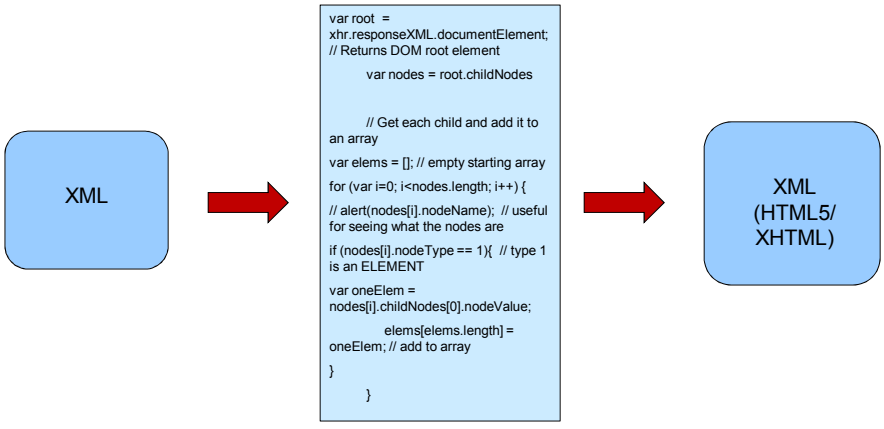
```
print “Content-type: text/xml…”;
print “<book>\n”;
…
```

3. How to read XML with Javascript on the client.

```
var doc = xhr.responseXML;
var children = doc.getElementsByTagName(“person”);
…
```

4

An Observation



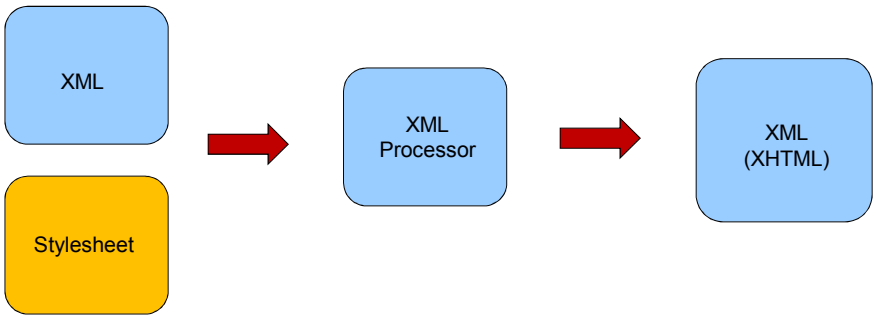
We write javascript simply to transform XML into different (almost) XML.

5

Why is this a bad model?

6

The Stylesheet Approach



Example: http://www.w3schools.com/xsl/tryxslt.asp?xmlfile=cdcatalog&xsltfile=cdcatalog

7

Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<RDF>
  <channel about="http://web2.0thebook.org/channel.rss">
    <title>Planet web2.0thebook</title>
    <title>This is our alternate title </title>
    <link>http://web2.0thebook.org/</link>
    <description>Aggregated content relevant to the upcoming book "Professional Web 2.0
      Programming".</description>
  </channel>
  <item about="http://www.orbeon.com/blog/2006/06/13/firebug-a-must-have-firefox-extension-for-web-
      developers/">
    <title>XForms Everywhere Â» FireBug: A Must-Have Firefox Extension for Web Developers</title>
    <link>http://www.orbeon.com/blog/2006/06/13/firebug-a-must-have-firefox-extension-for-web-
      developers/</link>
    <description>Alessandro Vernet recommends FireBug, â\200\234an absolute godsendâ\200\235, the
      â\200\234greatest web
      developer extension out thereâ\200\235, an â\200\234awesomeâ\200\235,
      â\200\234phenomenalâ\200\235, and â\200\234absolutely, completely
      brilliantâ\200\235 extension.</description>
  </item>
  <item about="http://eric.van-der-vlist.com/blog/2504_Web_2.0_at_XML_Prague.item">
    <title>Web 2.0 at Prague</title>
    <link>http://eric.van-der-vlist.com/blog/2504_Web_2.0_at_XML_Prague.item</link>
    <description>Eric van der Vlist will do a presentation about Web 2.0 at XML Prague
      2006.</description>
  </item>
</RDF>
```

8

XPath

XPath: syntax for selecting parts of an XML

XPath expression “navigate” the XML

/

```
/RDF
```

```
/RDF/channel/title
```

```
channel/title
```

```
channel//title
```

```
item[@title='Jaw surgery']
```

More complex:

```
/RDF/item[@about=current()/@resource]
```

Exercise: XPath (part 1)

```
<booklist listtitle="Science Fiction">
    <book>
        <title>The Naked Sun</title>
        <author>Isaac Asimov</author>
        <isbn>0553293397</isbn>
        <price>30</price> <!-- add by hand to online demo -->
    </book>

    <book>
        <title>Foundation's Triumph</title>
        <author>David Brin</author>
        <isbn>0061056391</isbn>
        <price>20</price> <!-- add by hand to online demo -->

    </book>

    <book>
        <title>Snow Crash</title>
        <author>Neal Stephenson</author>
        <isbn>0553380958</isbn>
    </book>
</booklist>
```

Demo from: http://www.futurelab.ch/xmlkurs/xpath.en.html

Exercise: XPath (part 2)

Write XPath to find:

1. All the book elements

2. The ISBN of all the books

3. Book elements with author Isaac Asimov

4. The title of all books priced > 20

11

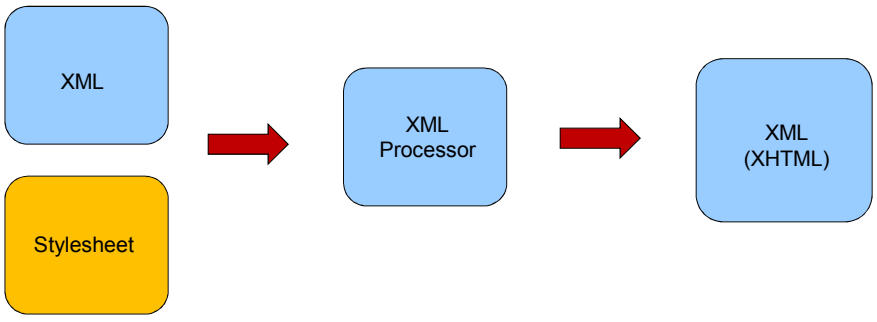
XSLT

- XSL – “Extensible Stylesheet Language”
- Parts
  - XSLT – “XSL Transformations”
  - XPath – how to identify a node in an XML document?

- Not primarily used for style
- Not a replacement for CSS

12

Recall: The Stylesheet Approach



Example: http://www.w3schools.com/xsl/tryxslt.asp?xmlfile=cdcatalog&xsltfile=cdcatalog

Simple XML with Style

Add style here!

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

<RDF>
 <channel about="http://web2.0thebook.org/channel.rss">
  <title>Planet web2.0thebook</title>
  <title>This is our alternate title </title>
  <link>http://web2.0thebook.org/</link>
  <description>Aggregated content relevant to the upcoming book "Professional Web 2.0
   Programming".</description>
 </channel>
 <item about="http://www.orbeon.com/blog/2006/06/13/firebug-a-must-have-firefox-extension-for-web-developers/">
  <title>XForms Everywhere Â» FireBug: A Must-Have Firefox Extension for Web Developers</title>
  <link>http://www.orbeon.com/blog/2006/06/13/firebug-a-must-have-firefox-extension-for-web-developers/</link>
  <description>Alessandro Vernet recommends FireBug, â\200\234an absolute godsendâ\200\235, the â\200\234greatest
     web …</description>
 </item>
 <item about="http://eric.van-der-vlist.com/blog/2504_Web_2.0_at_XML_Prague.item">
  <title>Web 2.0 at Prague</title>
  <link>http://eric.van-der-vlist.com/blog/2504_Web_2.0_at_XML_Prague.item</link>
  <description>Eric van der Vlist will do a presentation about Web 2.0 at XML Prague 2006.</description>
 </item>
 <item about="http://www.orbeon.com/blog/2006/06/10/unicode-in-java-not-so-fast/">
  <title>XForms Everywhere Â» Unicode in Java: not so fast (but XML is better)!</title>
  <link>http://www.orbeon.com/blog/2006/06/10/unicode-in-java-not-so-fast/</link>
 </item>
</RDF>
```


XSLT (Example 0)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  version="1.0" >
  <xsl:template match="/">
    <html>
      <head>
        <title>
          <xsl:value-of select="RDF/channel/title"/>
        </title>
      </head>
      <body>
        <p>This page made with XSLT! </p> <!-- test message -->

        <div class="channel" id="planet">
          <h1> <xsl:value-of select="RDF/channel/title"/> </h1>
          <p>  <xsl:value-of select="RDF/channel/description"/> </p>
          <a href="{RDF/channel/link}">
            <img alt="RSS channel" src="feed-icon-24x24.png"/>
          </a>
        </div>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

XML + XSLT Output (example 0)

```
<html>
<head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Planet web2.0thebook</title>
</head>
<body>

<p>This page made with XSLT! </p>
<div class="channel“ id="planet">
<h1>Planet web2.0thebook</h1>
<p>Aggregated content relevant to the upcoming book "Professional Web 2.0
    Programming".</p>
<a href="http://web2.0thebook.org/">
  <img alt="RSS channel“ src="feed-icon-24x24.png"></a>
</div>
</body>
</html>
```

From http://www.futurelab.ch/xmlkurs/xslt.en.html

XSLT: Example 1

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   version="1.0" >
   <xsl:template match="/">
      <html>
         <head> <title>
               <xsl:value-of select="RDF/channel/title"/>
            </title>
         </head>
         <body>
            <p>This page made with XSLT - and more templates! </p>
            <xsl:apply-templates select="RDF/channel"/>
         </body>
      </html>
   </xsl:template>

   <xsl:template match="channel">
      <div class="channel" id="planet">
         <xsl:apply-templates select="title" />
         <xsl:apply-templates select="description" />
         <a href="{link}">
            <img alt="RSS channel" src="feed-icon-24x24.png"/>
         </a>
      </div>
   </xsl:template>
...
```

17

XSLT: Example 1, continued

```
<xsl:template match="title">
    <h1>
      <xsl:value-of select="."/>
    </h1>
  </xsl:template>


  <xsl:template match="/RDF/channel/description">
    <p class="description">
      <xsl:value-of select="."/>
    </p>
  </xsl:template>
</xsl:stylesheet>
```

18

Exercise: The previous 2 slides were one XSL file. I changed just the "title" and "description" templates of the XSL (below). What is the new output? (the next slide has it started for you)

```
<xsl:template match="title">
    <ul>
      <li> <xsl:value-of select="."/> </li>
      <li> <xsl:value-of select="../link"/> </li>
    </ul>
</xsl:template>

<xsl:template match="description">
    <p> Hello! </p>
</xsl:template>
```

```
<html>
<head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Planet web2.0thebook</title>
</head>
<body>

<p>This page made with XSLT - and more templates! </p>
<div id="planet" class="channel">
```


Status Check

- We have XML documents.
- We have XSLT stylesheets.
- We changed XML docs into **entire XHTML pages. using the browser's built in transformation**

- Now let's ..
  a. Transform XML into a XHTML page **using JavaScript**
  b. Extract part of the transformed page (use to identify what part)
  c. Insert that part into our "regular" page

Example 2: HTML inserts transformed XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Professional Web 2.0 Programming</title>
        <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7/jquery.min.js"></script>
        <script type="text/javascript" src="transform.js"> </script>
    </head>
    <body onload="transform(‘channelNoNS.xml’, ‘2_rss.xsl')">
        <div id="book">
            <h1>A page with some exciting content </h1>

            <p>Web 2.0 offers developers substantial advantages if they design their web
                applications as service providers and service consumers. This change in architecture
                has opened up an incredible number of options for flexible design, creative reuse,
                and easier updates. There is, however a cost: doing this requires rethinking how to
                apply many traditional web development technologies, as well as adding some new
                ideas.</p>
        </div>

    <h2>Below is some content from elsewhere:</h2>

        <div id="planet">
          <h1>This element should be replaced by JS.
               If you see this content, something went wrong with the XSLT!</h1>
        </div>
    </body>
</html>
```


Ex. 2: HTML inserts transformed XML (part A)

```
var xml = null;
var xsl = null;

function transform (xmlFileName, xslFileName) {

  var settings = {
    type: "GET",
    error: function(xhr, status, error) {
                window.alert("Failed to load the input as an XML document!");
            window.alert("Raw text was: " + xhr.responseText);
    }};

  // Get the XML input data
  settings.url = xmlFileName;
  settings.success = function(data) { xml = data; insertXML(); };
  $.ajax(settings);

  // Get the XSLT file
  settings.url = xslFileName;
  settings.success = function(data) { xsl = data; insertXML(); };
  $.ajax(settings);

}
```

Ex 2: HTML inserts transformed XML (part B)

```
function insertXML() {
  // Make sure both AJAX requests have returned.
  if( xml != null && xsl != null ) {

    // Transform the XML via the XSLT
    var processor = new XSLTProcessor();
    processor.importStylesheet(xsl);
    var newDocument = processor.transformToDocument(xml);
    if (newDocument == null) {
      window.alert("Failed to convert the new document!");
      return;
    }

    // Replace part of original document with the new content
    var o = document.getElementById("planet");
    if (o == null) {
      window.alert("Failed to find element for 'o'!");
      return;
    }

    var n = newDocument.getElementById("planet");
    if (n == null) {
      window.alert("Failed to find element for 'n'!");
      return;
    }

    o.parentNode.replaceChild(n, o);
    // Reset the variables to null.
    xml = null;
    xsl = null;
  }
}
```


Example 3: Improved XSL

```
...
...
...
...
          <script type="text/javascript" src="transform.js"> </script>
    </head>
    <body onload="transform(‘channelNoNS.xml’, '3_rss.xsl')">
        <div id="book">
            <h1>A page with some exciting content </h1>
...
...
...
...
```

25

Example 3: Improved XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:template match="/">
        <html>
            <head>
                <title>
                    <xsl:value-of select="RDF/channel/title"/>
                </title>
            </head>
            <body>

                <div class="channel" id="planet">
                    <h1> <xsl:value-of select="RDF/channel/title"/> </h1>
                    <p>  <xsl:value-of select="RDF/channel/description"/></p>
                    <a href="{RDF/channel/link}">
                      <img alt="RSS channel" src="feed-icon-24x24.png"/>
                    </a>

                    <ul>
                        <xsl:apply-templates select="/RDF/item"/>
                    </ul>
                </div>
            </body>
        </html>
    </xsl:template>

    ... (more on next page)
```

26

Example 3: Improved XSL

```
 <!-- deal with each <item> -->
    <xsl:template match="item">
        <li>
            <div class="item">
                <h2>
                    <a href="{./link}">
                        <xsl:value-of select="./title"/>
                    </a>
                </h2>
                <p class="description">
                    <xsl:value-of select="description"/>
                </p>
            </div>
        </li>
    </xsl:template>

</xsl:stylesheet>
```

Example 4: Nicer Looking XSL Output

```
...
...
...
...
        <script type="text/javascript" src="2_transform.js"> </script>
    </head>
    <body onload="transform('channelNoNS.xml', '4_rss.xsl')">
        <div id="book">
            <h1>A page with some exciting content </h1>
...
...
...
...
```


Example 4: Nicer Looking XSL Output

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:template match="/">
        <html>
            <head>
                <title>
                    <xsl:value-of select="RDF/channel/title"/>
                </title>
            </head>
            <body>
                <div class="channel" id="planet">
                    <h1> <xsl:value-of select="RDF/channel/title"/> ... Get it!
                        <a href="{RDF/channel/link}">
                          <img alt="RSS channel" src="feed-icon-24x24.png"/>
                        </a>
                    </h1>
                    <p>  <xsl:value-of select="RDF/channel/description"/></p>

                    <table border="1">
                        <xsl:apply-templates select="/RDF/item"/>
                    </table>
                </div>
            </body>
        </html>
    </xsl:template>
```

29

Example 4: Nicer Looking XSL Output

```
<!-- deal with each <item> -->
    <xsl:template match="item">
            <tr>
                <td> Votes: <xsl:value-of select="@votes" /> </td>
                <td> <a href="{./link}">
                        <xsl:value-of select="./title"/> </a>
                </td>
                <td>
                    <xsl:value-of select="description"/>
                </td>
            </tr>
    </xsl:template>

</xsl:stylesheet>
```

30

Sidebar: XML with namespaces

XML elements and attributes have **namespaces**

<title>…</title>

<bn:title>…</bn:title>

<usna:title>…</usna:title>

Why namespaces?

31

XML with namespaces

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF   xmlns:rdf="http://realPrefixRemovedForThisDemo.org#"
           xmlns="http://purl.org/rss/1.0/"
           xmlns:dc="http://purl.org/dc/elements/1.1/" >

  <channel rdf:about="http://web2.0thebook.org/channel.rss">
    <title>Planet web2.0thebook</title>
    <link>http://web2.0thebook.org/</link>
    <description>Aggregated content relevant to the upcoming book "Professional Web 2.0
      Programming".</description>
  </channel>
…
```

32

XML with namespaces

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://realPrefixRemovedForThisDemo.org#" xmlns="http://purl.org/rss/1.0/"
  xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <channel rdf:about="http://web2.0thebook.org/channel.rss">
    <title>Planet web2.0thebook</title>
    <link>http://web2.0thebook.org/</link>
    <description>Aggregated content relevant to the upcoming book "Professional Web 2.0
      Programming".</description>
  </channel>
  <item
    rdf:about="http://www.orbeon.com/blog/2006/06/13/firebug-a-must-have-firefox-extension-for-web-developers/">
    <title>XForms Everywhere Â» FireBug: A Must-Have Firefox Extension for Web Developers</title>
    <link>http://www.orbeon.com/blog/2006/06/13/firebug-a-must-have-firefox-extension-for-web-developers/</link>
    <description>Alessandro Vernet recommends FireBug, â\200\234an absolute godsendâ\200\235, the
     â\200\234greatest web
      developer extension out thereâ\200\235, an â\200\234awesomeâ\200\235, â\200\234phenomenalâ\200\235, and
     â\200\234absolutely, completely
      brilliantâ\200\235 extension.</description>
    <dc:creator>evlist</dc:creator>
    <dc:date>2006-06-15T05:56:16Z</dc:date>
    <dc:subject>ajax debugger dom firefox javascript tools web2.0thebook webdev</dc:subject>
  </item>
  <item rdf:about="http://eric.van-der-vlist.com/blog/2504_Web_2.0_at_XML_Prague.item">
    <title>Web 2.0 at Prague</title>
    <link>http://eric.van-der-vlist.com/blog/2504_Web_2.0_at_XML_Prague.item</link>
    <description>Eric van der Vlist will do a presentation about Web 2.0 at XML Prague 2006.</description>  ...
```

Which mean the same thing?

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://realPrefixRemovedForThisDemo.org#" xmlns="http://purl.org/rss/1.0/"
  xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <channel rdf:about="http://web2.0thebook.org/channel.rss">
    <title>Planet web2.0thebook</title>
    <link>http://web2.0thebook.org/</link>
  </channel>      ...

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://realPrefixRemovedForThisDemo.org#" xmlns:rss="http://purl.org/rss/1.0/"
  xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <rss:channel rdf:about="http://web2.0thebook.org/channel.rss">
    <rss:title>Planet web2.0thebook</rss:title>
    <rss:link>http://web2.0thebook.org/</rss:link>
  </rss:channel>   ...

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://realPrefixRemovedForThisDemo.org#" xmlns:rss="http://blahblahblah.com/stuff"
  xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <rss:channel rdf:about="http://web2.0thebook.org/channel.rss">
    <rss:title>Planet web2.0thebook</rss:title>
    <rss:link>http://web2.0thebook.org/</rss:link>
  </rss:channel>   ...

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://realPrefixRemovedForThisDemo.org#" xmlns:dog="http://purl.org/rss/1.0/"
  xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <dog:channel rdf:about="http://web2.0thebook.org/channel.rss">
    <dog:title>Planet web2.0thebook</dog:title>
    <dog:link>http://web2.0thebook.org/</dog:link>
  </dog:channel> ...

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://realPrefixRemovedForThisDemo.org#" xmlns:dog="http://blahblahblah.com/stuff"
  xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <dog:channel rdf:about="http://web2.0thebook.org/channel.rss">
    <dog:title>Planet web2.0thebook</dog:title>
    <dog:link>http://web2.0thebook.org/</dog:link>
  </dog:channel> ...
```


XPath with Namespaces

```
<booklist listtitle="Science Fiction“ xmlns:bn=“http://barnes..”>
    <bn:book>
        <title>The Naked Sun</title>
        <author>Isaac Asimov</author>
        <isbn>0553293397</isbn>
        <price>30</price> <!-- add by hand to online demo -->
    </bn:book>
    </booklist>
```

Grab all books:

Moby Dick is a sperm whale who is the main antagonist in Herman Melville's 1851 novel of the same name. Melville based the fictional whale partially on a real albino whale of that period called Mocha Dick. Ishmael describes Moby Dick as having two prominent white areas around "a peculiar snow-white wrinkled forehead, and a high, pyramidical white hump", the rest of his body being of stripes and patches between white and gray. The animal's exact dimensions are never given, but the novel claims that the Moby-Dick was published to mixed reviews, was a commercial failure, and was out of print at the time of the author's death in 1891. Its reputation as a "Great American Novel" was established only in the 20th century, after the centennial of its author's birth.Â Melville began writing Moby-Dick in February 1850, and would eventually take 18 months to write the book, a full year more than he had first anticipated. Moby Dick, novel (1851) by Herman Melville detailing the voyage of the Pequod, a whaling vessel whose captain is intent on finding the white sperm whale Moby Dick. The novel was not well received at first but is now widely regarded as Melvilleâ€™s magnum opus and one of the greatest novels in American literature.Â She received her bachelorâ€™s degree in philosophy and creative writing in 2020 at the University of Iowa. See Article History. Alternative Titles: â€œMoby-Dick; or, The Whaleâ€, â€œThe Whaleâ€. The sole survivor of a lost whaling ship relates the tale of his Captain's self-destructive obsession to hunt the white whale, Moby Dick.Â The sole survivor of a lost whaling ship relates the tale of his Captain's self-destructive obsession to hunt the white whale, Moby Dick. Stars: Henry Thomas, Patrick Stewart, Bruce Spence | See full cast & crew Â».

Moby-Dick is considered one of the greatest novels in the English language and has secured Melville's place among America's greatest writers. Moby dick or the whale. CHAPTER 1: Loomings. Call me Ishmael. Share this Rating. Title: Moby Dick (1956). 7,3/10. Want to share IMDb's rating on your own site?Â This classic story by Herman Melville revolves around Captain Ahab and his obsession with a huge whale, Moby Dick. The whale caused the loss of Ahab's leg years before, leaving Ahab to stomp the boards of his ship on a peg leg. Ahab is so crazed by his desire to kill the whale, that he is prepared to sacrifice everything, including his life, the lives of his crew members, and even his ship to find and destroy his nemesis, Moby Dick. Moby-Dick was published to mixed reviews, was a commercial failure, and was out of print at the time of the author's death in 1891. Its reputation as a "Great American Novel" was established only in the 20th century, after the centennial of its author's birth.Â Melville began writing Moby-Dick in February 1850, and would eventually take 18 months to write the book, a full year more than he had first anticipated.