# XSLT

- Press `Space` to navigate through the slides
- Use `Shift+Space` to go back

# INTRODUCTION TO XSL

- **eXtensible Stylesheet Language** consists of:
    - A language for **transforming XML** documents
    - A language for **specifying formatting properties**
- Based on existing style sheets languages:
    - **Document Style Semantics and Specification Language (DSSSL)**
    - **Cascading Style Sheets (CSS)**
- A style sheet language specifically for XML documents
- Uses an XML syntax
- XSL is a combination of three specifications:
    - **XML Path Language (XPath)**
    - **XSL Transformations (XSLT)**
    - **XSL-FO (Formatting Objects)**

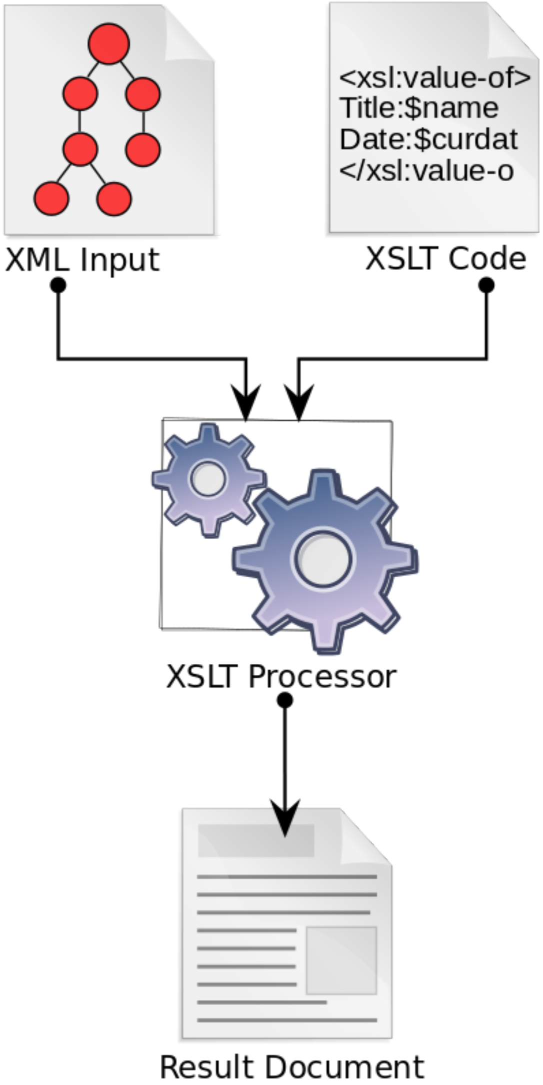
# WHAT IS XSLT?

- **eXtensible Stylesheet Language Transformations**
- **Language** for transforming XML documents into other XML/other documents
    - More generally input can be any document as long as it is represented as tree
- Uses a collection of **templates (rules)** to transform the source document
- **XPath** is used to select the parts of a source document to transform
- Server-side/client-side execution

# WHY USE XSLT?

- Powerful transformation functionality, i.e. **XML conversion**
- An XML document may be completely **re-structured**
    - **XML** to **HTML**
    - **XML** to **TXT**
    - **XML** to **JSON**
    - **XML** to **?**
- A variety of **XSLT processors** are available
- Easy and fast **prototyping** is possible

# INPUTS AND OUTPUTS IN XSLT

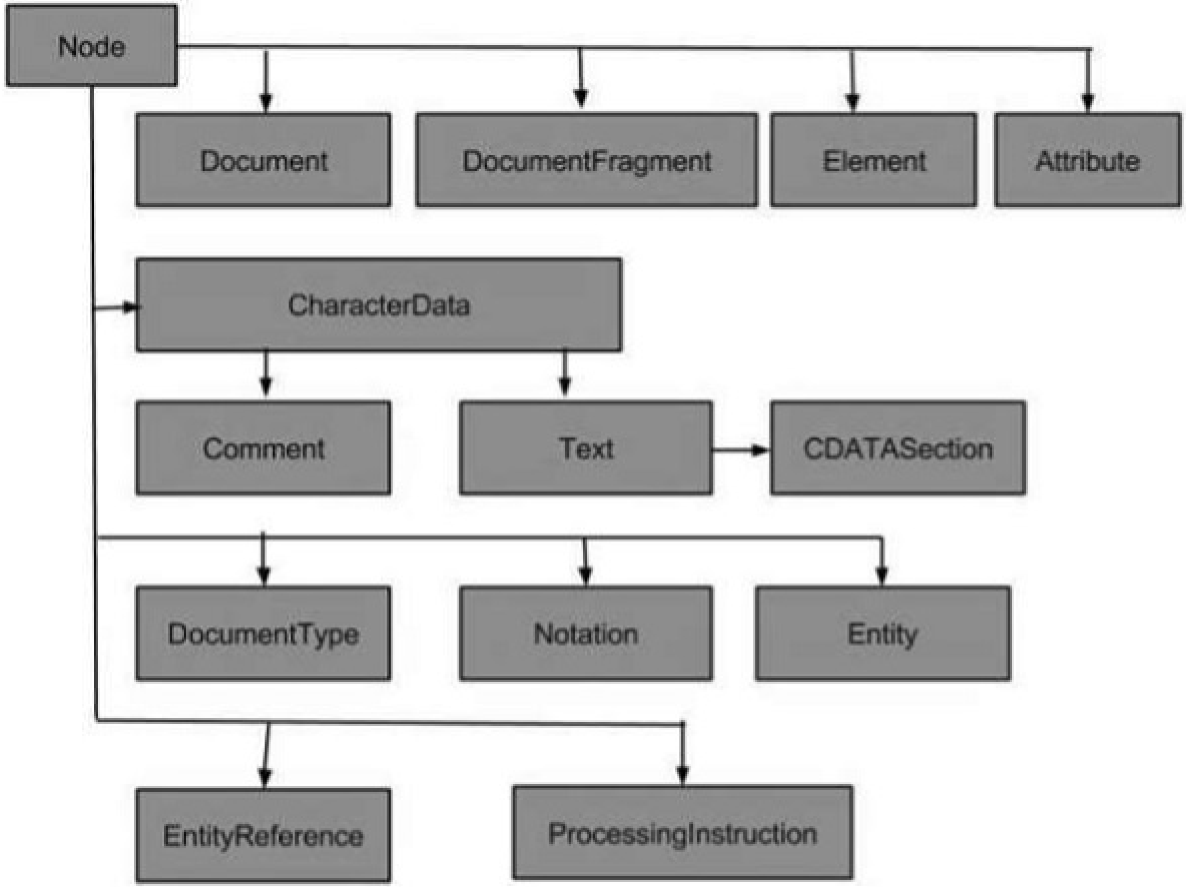

<xsl:value-of>
Title:$name
Date:$curdat
</xsl:value-o
XML Input
XSLT Code
XSLT Processor
Result Document

# NODE TYPES IN XML

- **Document Node:** complete XML document structure is a document node.
- **Element Node:** every XML element is an element node.
    - This is also the only type of node that can have attributes.
- **Attribute Node:** each attribute is considered an attribute node.
    - It contains information about an element node, but is not actually considered to be children of the element.
- **Text Node:** the document texts are considered as text node.
    - It can consist of more information or just white space.

# LESS COMMON NODE TYPES

- **CData Node:** this node contains information that should not be analysed by the parser.
- **Comment Node:** this node includes information about the data, and is usually ignored by the application.
- **Processing Instructions Node:** this node contains information specifically aimed at the application.
- **Document Fragments Node**, **Entities Node**, **Entity reference nodes**, **Notations Node**


Node
Document
DocumentFragment
Element
Attribute
CharacterData
Comment
Text
CDATASection
DocumentType
Notation
Entity
EntityReference
ProcessingInstruction

# XSLT:TEMPLATE

- The `<xsl:template>` element is used to build templates
- Elements in a template body classified as either **data node** or **instruction**
    - `<html>...</html>` and `<body>...</body>` below are **data nodes**
    - `<xsl:value-of select="//message" />` is an **instruction**

```
<xsl:template match="/">
    <html>
        <body>
            <h1><xsl:value-of select="//note" /></h1>
        </body>
    </html>
</xsl:template>
```

# XSLT:TEMPLATE

```
<xsl:template match="/">
    <html>
        <body>
            <h1><xsl:value-of select="//message" /></h1>
        </body>
    </html>
</xsl:template>
```

- The `match` attribute is used to associate a template with an XML element
- The `match` can also be used to define a template for the entire XML document.
- The value of the `match` attribute is an **XPath expression** (i.e. `match="/"` defines the whole document).

# EXAMPLE

In [ ]:

```
# %load examples/test.xml
<?xml version="1.0"?>
<document>
    <message>It worked</message>
</document>
```

In [ ]:

```
# %load examples/test.xsl
<?xml version="1.0"?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
    <html>
        <body>
            <h1><xsl:value-of select="//message" /></h1>
        </body>
    </html>
</xsl:template>
</xsl:transform>
```

# RESULT

In [607]:

```
from IPython.core.display import display, HTML
from bs4 import BeautifulSoup
from io import StringIO
import lxml.etree as ET

dom = ET.parse('test.xml')
xslt = ET.parse('test.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(ET.tostring(newdom, pretty_print=True))
print(result.prettify())

display(HTML(result.prettify()))
```

```
<html>
 <body>
  <h1>
   It worked
  </h1>
 </body>
</html>
```

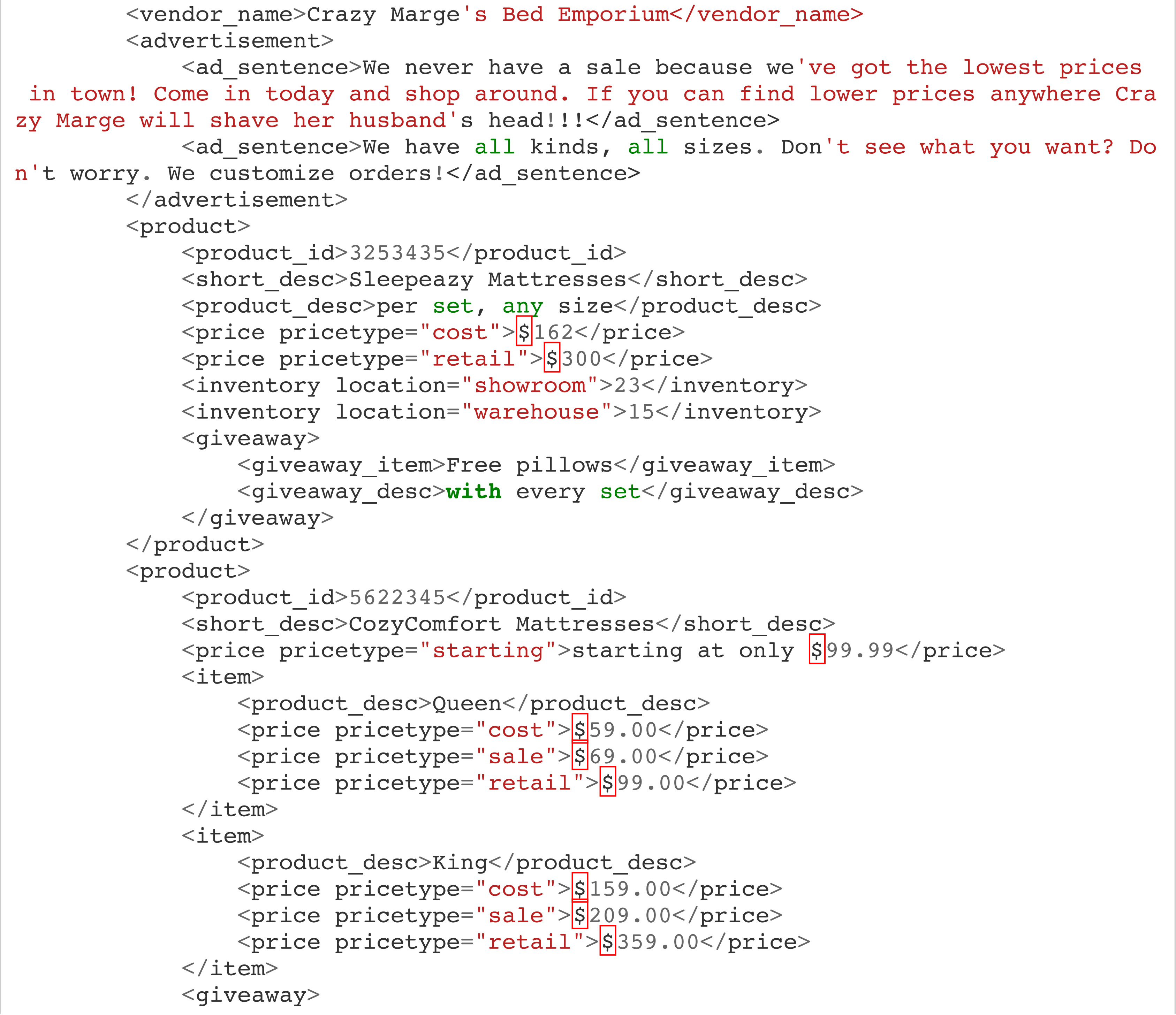
# IT WORKED

# PRODUCT.XML

In [ ]:

```
# %load examples/product.xml
<?xml version="1.0"?>
<products>
  <vendor webvendor="full" id="conners">
    <vendor_name>Conners Chair Company</vendor_name>
    <advertisement>
      <ad_sentence>
 Conners Chair Company presents their annual big three day only chair sale. We're making way for our new stock!
 <b>All current inventory must go!</b>
 Regular prices slashed by up to 60%!
 </ad_sentence>
    </advertisement>
    <product>
      <product_id>QA3452</product_id>
      <short_desc>Queen Anne Chair</short_desc>
      <price pricetype="cost">$85</price>
      <price pricetype="sale">$125</price>
      <price pricetype="retail">$195</price>
      <inventory color="royal blue" location="warehouse">12</inventory>
      <inventory color="royal blue" location="showroom">5</inventory>
      <inventory color="flower print" location="warehouse">16</inventory>
      <inventory color="flower print" location="showroom">3</inventory>
      <inventory color="seafoam green" location="warehouse">20</inventory>
      <inventory color="teal" location="warehouse">14</inventory>
      <inventory color="burgundy" location="warehouse">34</inventory>
      <giveaway>
        <giveaway_item>Matching Ottoman included</giveaway_item>
        <giveaway_desc>while supplies last</giveaway_desc>
      </giveaway>
    </product>
    <product>
      <product_id>RC2342</product_id>
      <short_desc>Early American Rocking Chair</short_desc>
      <product_desc>with brown and tan plaid upholstery</product_desc>
      <price pricetype="cost">$75</price>
      <price pricetype="sale">$62</price>
```

```xml
            <price pricetype="retail">$120</price>
            <inventory location="warehouse">40</inventory>
            <inventory location="showroom">2</inventory>
        </product>
        <product>
            <product_id>BR3452</product_id>
            <short_desc>Bentwood Rocker</short_desc>
            <price pricetype="cost">$125</price>
            <price pricetype="sale">$160</price>
            <price pricetype="retail">$210</price>
            <inventory location="showroom">3</inventory>
        </product>
    </vendor>
    <vendor webvendor="partial" id="wally">
        <vendor_name>Wally's Wonderful World of Furniture</vendor_name>
        <advertisement>
            <ad_sentence>Wally's Wonderful World of Furniture is closing its doors forever. Last chance to get great bargains. Make us an offer. We can't refuse!</ad_sentence>
        </advertisement>
        <suite>
            <product_id>CDRS</product_id>
            <short_desc>Complete Dining Room Set</short_desc>
            <long_desc>This five piece dining site set features swivel chairs with cushions in five exciting colors.</long_desc>
            <price pricetype="cost">$435</price>
            <price pricetype="sale">$699</price>
            <price pricetype="retail">$999</price>
            <product>
                <product_id>WWWdrt</product_id>
                <short_desc>Dining Room Table</short_desc>
                <price pricetype="cost">$105</price>
                <price pricetype="sale">$145</price>
                <price pricetype="retail">$195</price>
                <inventory location="warehouse">132</inventory>
            </product>
            <product>
                <product_id>WWWsc</product_id>
                <short_desc>Swivel Chair</short_desc>
                <price pricetype="cost">$50</price>
                <price pricetype="sale">$45</price>
                <price pricetype="retail">$99</price>
                <inventory location="warehouse">300</inventory>
            </product>
            <product>
                <product_id>WWWhch</product_id>
                <short_desc>Hutch</short_desc>
                <price pricetype="cost">$346</price>
                <price pricetype="sale">$425</price>
                <price pricetype="retail">$600</price>
                <inventory location="warehouse">232</inventory>
            </product>
        </suite>
        <product>
            <product_id>HallBench</product_id>
            <short_desc>Hall Bench</short_desc>
            <price pricetype="cost">$75</price>
            <price pricetype="sale">$62</price>
            <price pricetype="retail">$120</price>
            <inventory location="warehouse">143</inventory>
            <inventory location="showroom">5</inventory>
        </product>
        <product>
            <product_id>SofaLoveSeat</product_id>
            <short_desc>Sofa and Love Seat</short_desc>
            <price color="magnolia print" pricetype="cost">$125</price>
            <price color="nautical print" pricetype="cost">$145</price>
            <price pricetype="sale">$175</price>
            <price pricetype="retail">$250</price>
            <inventory color="magnolia print" location="showroom">3</inventory>
            <inventory color="magnolia print" location="warehouse">36</inventory>
            <inventory color="nautical print" location="warehouse">1</inventory>
            <inventory color="nautical print" location="showroom">432</inventory>
        </product>
    </vendor>
    <vendor webvendor="no" id="marge">
```



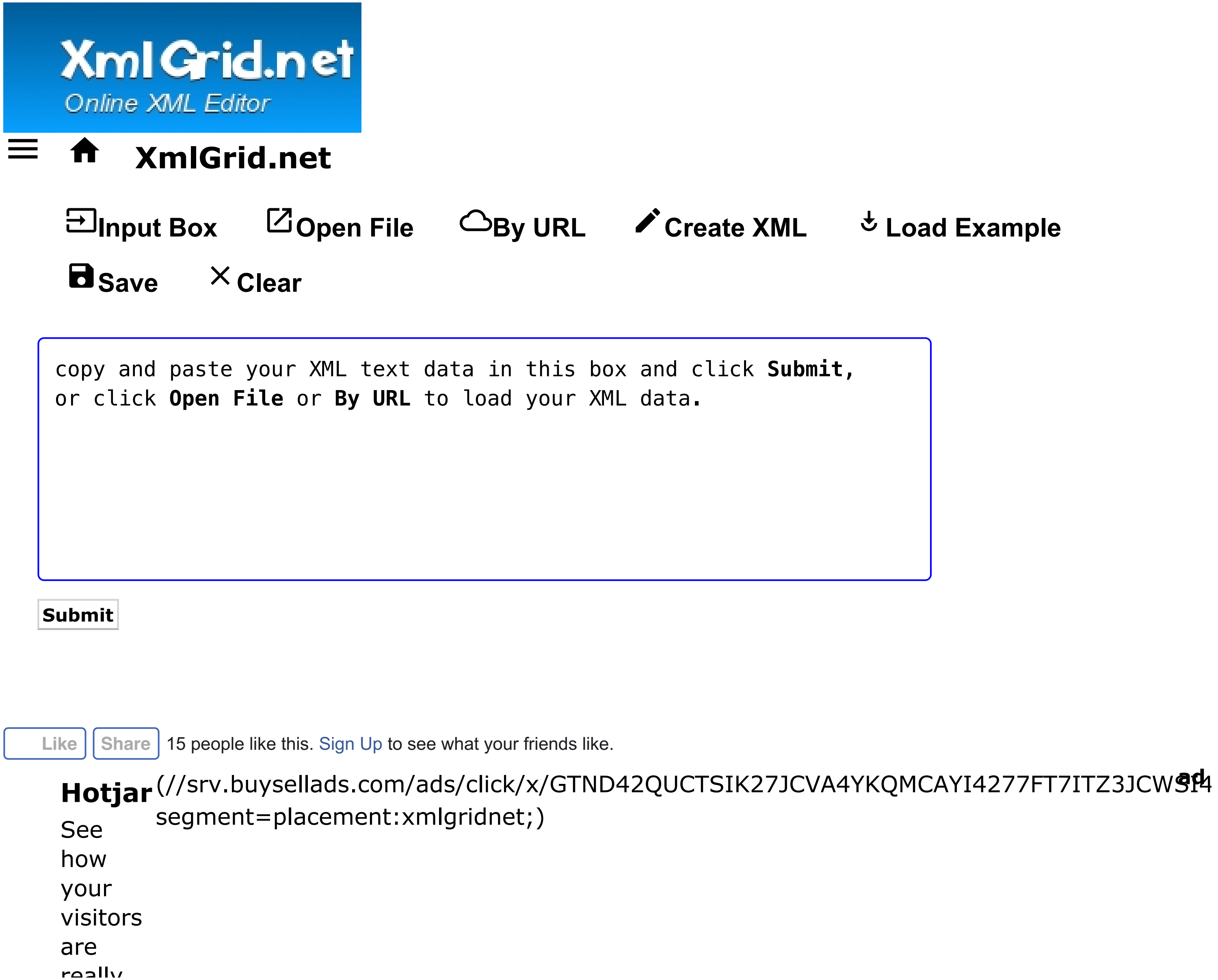
```
        <vendor_name>Crazy Marge's Bed Emporium</vendor_name>
        <advertisement>
            <ad_sentence>We never have a sale because we've got the lowest prices in town! Come in today and shop around. If you can find lower prices anywhere Crazy Marge will shave her husband's head!!!</ad_sentence>
            <ad_sentence>We have all kinds, all sizes. Don't see what you want? Don't worry. We customize orders!</ad_sentence>
        </advertisement>
        <product>
            <product_id>3253435</product_id>
            <short_desc>Sleepeazy Mattresses</short_desc>
            <product_desc>per set, any size</product_desc>
            <price pricetype="cost">$162</price>
            <price pricetype="retail">$300</price>
            <inventory location="showroom">23</inventory>
            <inventory location="warehouse">15</inventory>
            <giveaway>
                <giveaway_item>Free pillows</giveaway_item>
                <giveaway_desc>with every set</giveaway_desc>
            </giveaway>
        </product>
        <product>
            <product_id>5622345</product_id>
            <short_desc>CozyComfort Mattresses</short_desc>
            <price pricetype="starting">starting at only $99.99</price>
            <item>
                <product_desc>Queen</product_desc>
                <price pricetype="cost">$59.00</price>
                <price pricetype="sale">$69.00</price>
                <price pricetype="retail">$99.00</price>
            </item>
            <item>
                <product_desc>King</product_desc>
                <price pricetype="cost">$159.00</price>
                <price pricetype="sale">$209.00</price>
                <price pricetype="retail">$359.00</price>
            </item>
            <giveaway>
```

# PRODUCT.XML

In [3]:

```python
from IPython.display import IFrame
IFrame(src='https://xmlgrid.net', width="100%", height="600px")
```

Out[3]:



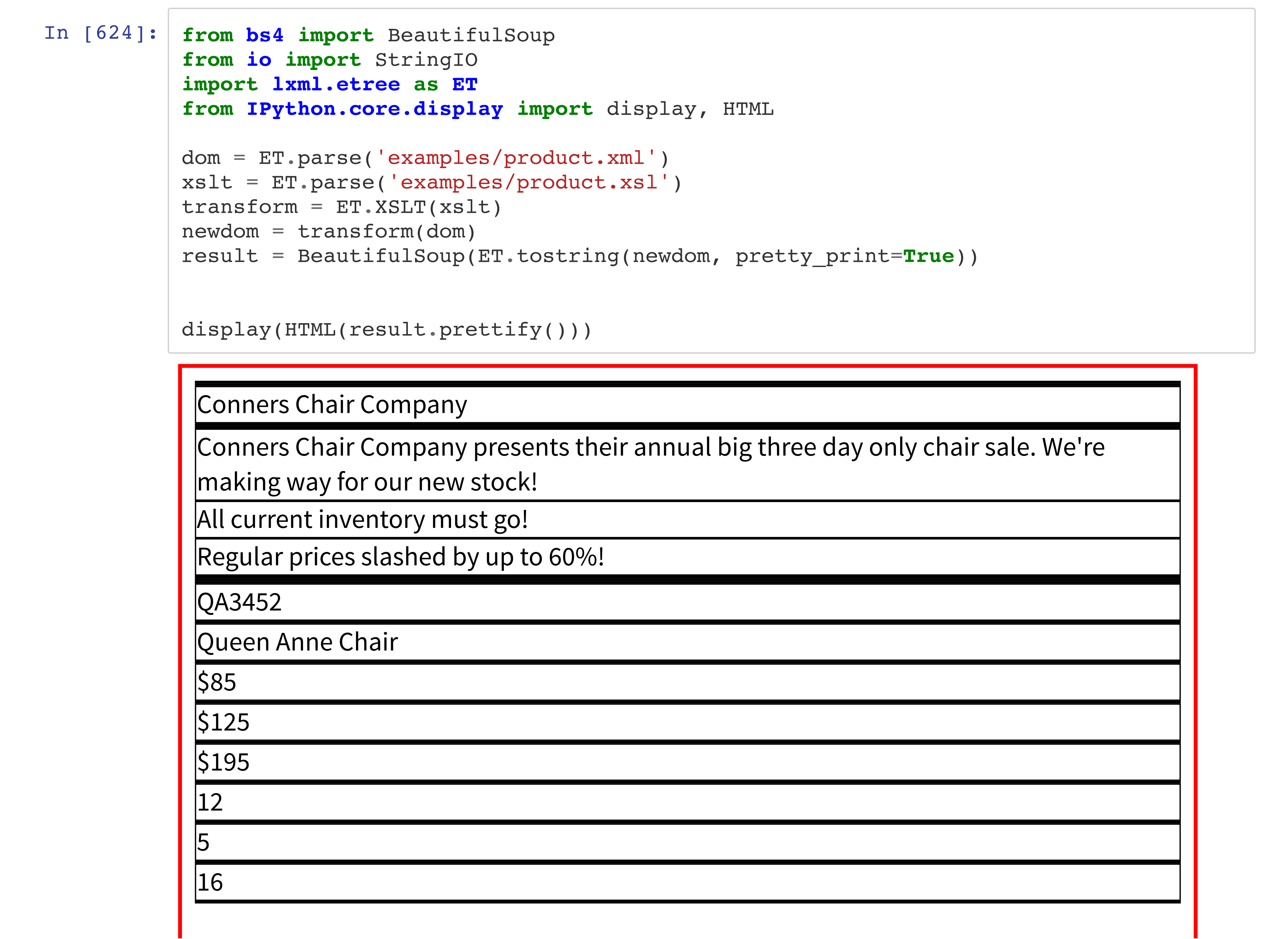
copy and paste your XML text data in this box and click **Submit**, or click **Open File** or **By URL** to load your XML data.

15 people like this. Sign Up to see what your friends like.

In [ ]:

```
# %load examples/product.xsl
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:template match="*">
        <xsl:apply-templates/>
    </xsl:template>
    <xsl:template match="text()">
       <div style="border: solid 1px">
          <xsl:value-of select="."/>
       </div>
    </xsl:template>
    <xsl:template match="/">
       <div style="border: solid 3px red; padding: 10px">
          <xsl:apply-templates/>
       </div>
    </xsl:template>
</xsl:transform>
```

# RESULT



In [624]:

```
from bs4 import BeautifulSoup
from io import StringIO
import lxml.etree as ET
from IPython.core.display import display, HTML

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(ET.tostring(newdom, pretty_print=True))


display(HTML(result.prettify()))
```

| |
|---|
| Conners Chair Company |
| Conners Chair Company presents their annual big three day only chair sale. We're making way for our new stock! |
| All current inventory must go! |
| Regular prices slashed by up to 60%! |
| QA3452 |
| Queen Anne Chair |
| $85 |
| $125 |
| $195 |
| 12 |
| 5 |
| 16 |

| |
|---|
| 3 |
| 20 |
| 14 |
| 34 |
| Matching Ottoman included |
| while supplies last |
| RC2342 |
| Early American Rocking Chair |
| with brown and tan plaid upholstery |
| $75 |
| $62 |
| $120 |
| 40 |
| 2 |
| BR3452 |
| Bentwood Rocker |
| $125 |
| $160 |
| $210 |
| 3 |
| Wally's Wonderful World of Furniture |
| Wally's Wonderful World of Furniture is closing its doors forever. Last chance to get great bargains. Make us an offer. We can't refuse! |

| |
|---|
| CDRS |
| Complete Dining Room Set |
| This five piece dining site set features swivel chairs with cushions in five exciting colors. |
| $435 |
| $699 |
| $999 |
| WWWdrt |
| Dining Room Table |
| $105 |
| $145 |
| $195 |
| 132 |
| WWWsc |
| Swivel Chair |
| $50 |
| $45 |
| $99 |
| 300 |
| WWWhch |
| Hutch |
| $346 |
| $425 |
| $600 |

| |
|---|
| 232 |
| HallBench |
| Hall Bench |
| $75 |
| $62 |
| $120 |
| 143 |
| 5 |
| SofaLoveSeat |
| Sofa and Love Seat |
| $125 |
| $145 |
| $175 |
| $250 |
| 3 |
| 36 |
| 1 |
| 432 |
| Crazy Marge's Bed Emporium |
| We never have a sale because we've got the lowest prices in town! Come in today and shop around. If you can find lower prices anywhere Crazy Marge will shave her husband's head!!! |
| We have all kinds, all sizes. Don't see what you want? Don't worry. We customize orders! |

| |
|---|
| 3253435 |
| Sleepeazy Mattresses |
| per set, any size |
| $162 |
| $300 |
| 23 |
| 15 |
| Free pillows |
| with every set |
| 5622345 |
| CozyComfort Mattresses |
| starting at only $99.99 |
| Queen |
| $59.00 |
| $69.00 |
| $99.00 |
| King |
| $159.00 |
| $209.00 |
| $359.00 |
| Free sheets |
| with every set |

# SELECT SPECIFIC DATA

- Let's try and select all present **vendor names** in **product.xml**

In [625]:

```
%%writefile examples/product1.xsl
<?xml version="1.0"?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

   <xsl:template match="/">
     <html>
        <body>
            <xsl:apply-templates select="//vendor"/>
        </body>
     </html>
    </xsl:template>


    <xsl:template match="//vendor">
       <p><xsl:value-of select="vendor_name"/></p>
    </xsl:template>


</xsl:transform>
```

```
Overwriting examples/product1.xsl
```

# RESULTING XML

In [626]:

```
from bs4 import BeautifulSoup
import lxml.etree as ET

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product1.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(str(newdom))
print(result.prettify())
```

```
<html>
 <body>
  <p>
   Conners Chair Company
  </p>
  <p>
   Wally's Wonderful World of Furniture
  </p>
  <p>
   Crazy Marge's Bed Emporium
  </p>
 </body>
</html>
```

# RESULTING XML VIEWED IN BROWSER:

In [627]:

```
from IPython.core.display import display, HTML
display(HTML(str(newdom)))
```

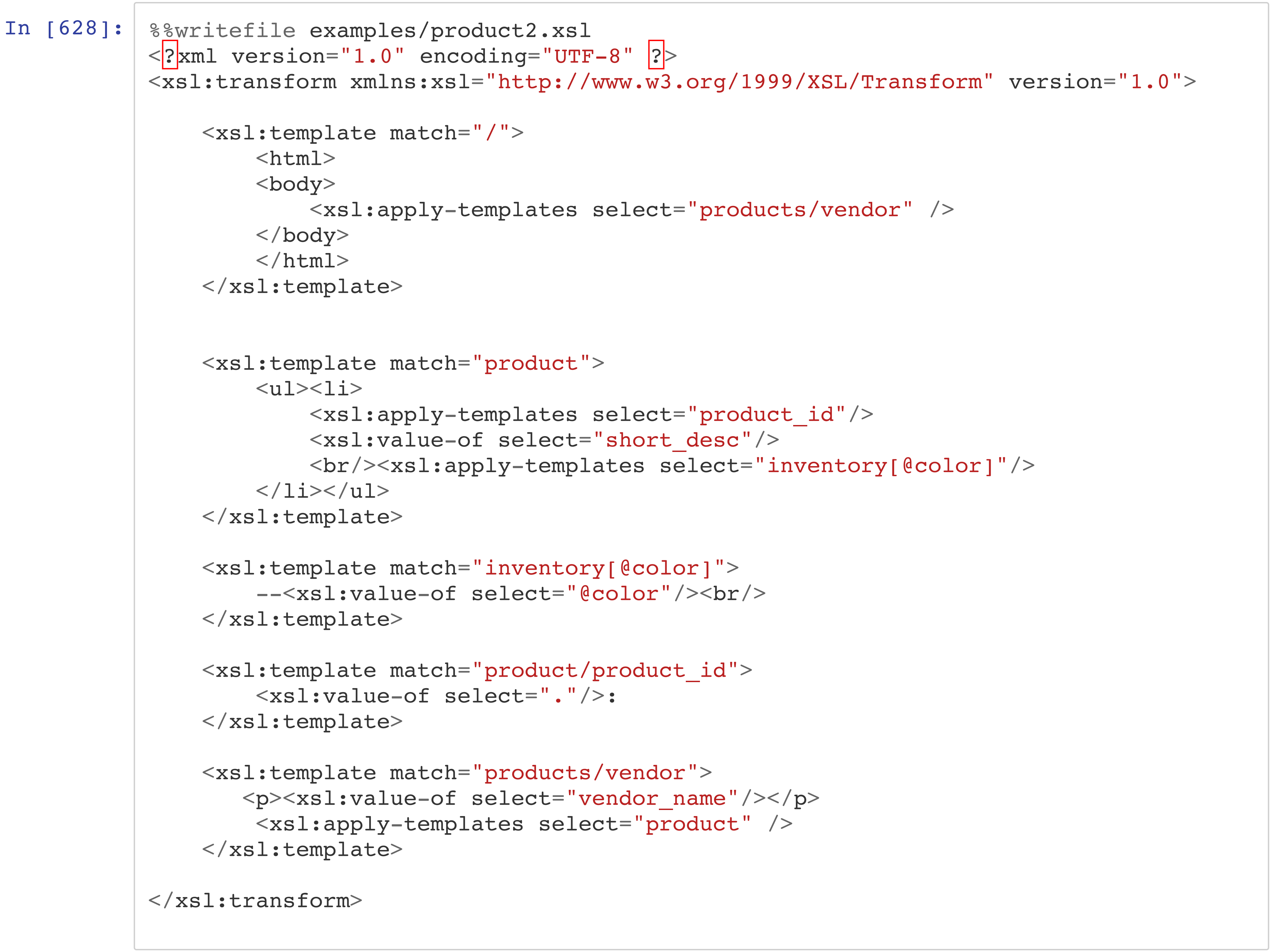
Conners Chair Company

Wally's Wonderful World of Furniture

Crazy Marge's Bed Emporium

# SELECT ATTRIBUTES

- Let's print out available inventory `color` instances



In [628]:

```
%%writefile examples/product2.xsl
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

    <xsl:template match="/">
        <html>
        <body>
            <xsl:apply-templates select="products/vendor" />
        </body>
        </html>
    </xsl:template>


    <xsl:template match="product">
        <ul><li>
            <xsl:apply-templates select="product_id"/>
            <xsl:value-of select="short_desc"/>
            <br/><xsl:apply-templates select="inventory[@color]"/>
        </li></ul>
    </xsl:template>

    <xsl:template match="inventory[@color]">
        --<xsl:value-of select="@color"/><br/>
    </xsl:template>

    <xsl:template match="product/product_id">
        <xsl:value-of select="."/>:
    </xsl:template>

    <xsl:template match="products/vendor">
       <p><xsl:value-of select="vendor_name"/></p>
        <xsl:apply-templates select="product" />
    </xsl:template>

</xsl:transform>
```

In [635]:

```python
from bs4 import BeautifulSoup
     import lxml.etree as ET
from IPython.core.display import display, HTML

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product2.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(str(newdom))
html = result.prettify()
print(html)
f=open("examples/product2.html","w+")
f.write(html)
```

```
<html>
 <body>
  <p>
   Conners Chair Company
  </p>
  <ul>
   <li>
    QA3452:
    Queen Anne Chair
    <br/>
    --royal blue
    <br/>
    --royal blue
    <br/>
    --flower print
    <br/>
    --flower print
    <br/>
    --seafoam green
    <br/>
    --teal
    <br/>
    --burgundy
    --burgundy
    <br/>
   </li>
  </ul>
  <ul>
   <li>
    RC2342:
    Early American Rocking Chair
    <br/>
   </li>
  </ul>
  <ul>
   <li>
    BR3452:
    Bentwood Rocker
    <br/>
   </li>
  </ul>
  <p>
   Wally's Wonderful World of Furniture
  </p>
  <ul>
   <li>
    HallBench:
    Hall Bench
    <br/>
   </li>
  </ul>
  <ul>
   <li>
    SofaLoveSeat:
    Sofa and Love Seat
    <br/>
    --magnolia print
    <br/>
    --magnolia print
    <br/>
    --nautical print
    <br/>
```

```
    --nautical print
    <br/>
   </li>
  </ul>
  <p>
   Crazy Marge's Bed Emporium
  </p>
  <ul>
   <li>
    3253435:
    Sleepeazy Mattresses
    <br/>
   </li>
  </ul>
  <ul>
   <li>
    5622345:
    CozyComfort Mattresses
    <br/>
   </li>
  </ul>
  <ul>
   <li>
    39981234:
    Floataway Waterbeds
    <br/>
   </li>
  </ul>
 </body>
</html>
```

```
Out[635]:  1111
```

In [4]:

```
from IPython.display import IFrame
IFrame(src='examples/product2.html', width="100%", height="800px")
```

Out[4]:

Conners Chair Company

- QA3452: Queen Anne Chair
  --royal blue
  --royal blue
  --flower print
  --flower print
  --seafoam green
  --teal
  --burgundy

- RC2342: Early American Rocking Chair

- BR3452: Bentwood Rocker

Wally's Wonderful World of Furniture

- HallBench: Hall Bench

- SofaLoveSeat: Sofa and Love Seat
  --magnolia print
  --magnolia print
  --nautical print
  --nautical print

Crazy Marge's Bed Emporium

- 3253435: Sleepeazy Mattresses

- 5622345: CozyComfort Mattresses

- 39981234: Floataway Waterbeds

# SELECT ATTRIBUTES WITH VALUES

- Let's select inventory `color` and a specific `location`

In [653]:

```
%%writefile examples/product3.xsl
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

    <xsl:template match="/">
    <html>
    <head><title></title></head>
    <body>
            <xsl:apply-templates select="/products/vendor" />
    </body>
    </html>
    </xsl:template>



    <xsl:template match="product">
        <ul><li>
            <xsl:apply-templates select="product_id"/>
            <xsl:value-of select="short_desc"/>
            <br/><xsl:apply-templates select="inventory[@color and @location='warehouse']"/>
        </li></ul>
    </xsl:template>

    <xsl:template match="inventory[@color and @location='warehouse']">
        --<xsl:value-of select="@color"/><br/>
    </xsl:template>

    <xsl:template match="product/product_id">
        <xsl:value-of select="."/>:
    </xsl:template>

    <xsl:template match="products/vendor">
       <p><xsl:value-of select="vendor_name"/></p>
       <xsl:apply-templates select=".//product" />
    </xsl:template>
```

In [654]:

```python
from bs4 import BeautifulSoup
import lxml.etree as ET
from IPython.core.display import display, HTML

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product3.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(str(newdom))
html = result.prettify()
print(html)
f=open("examples/product3.html","w+")
f.write(html)
```

```
<html>
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>
  </title>
 </head>
 <body>
  <p>
   Conners Chair Company
  </p>
  <ul>
   <li>
    QA3452:
    Queen Anne Chair
    <br/>
    --royal blue
    <br/>
    --flower print
    <br/>
    --seafoam green
    <br/>
    --teal
    <br/>
    <br/>
    --burgundy
    <br/>
   </li>
  </ul>
  <ul>
   <li>
    RC2342:
    Early American Rocking Chair
    <br/>
   </li>
  </ul>
  <ul>
   <li>
    BR3452:
    Bentwood Rocker
    <br/>
   </li>
  </ul>
  <p>
   Wally's Wonderful World of Furniture
  </p>
  <ul>
   <li>
    WWWdrt:
    Dining Room Table
    <br/>
   </li>
  </ul>
  <ul>
   <li>
    WWWsc:
    Swivel Chair
    <br/>
   </li>
  </ul>
  <ul>
   <li>
    WWWhch:
```

```
  Hutch
  <br/>
 </li>
</ul>
<ul>
 <li>
  HallBench:
  Hall Bench
  <br/>
 </li>
</ul>
<ul>
 <li>
  SofaLoveSeat:
  Sofa and Love Seat
  <br/>
  --magnolia print
  <br/>
  --nautical print
  <br/>
 </li>
</ul>
<p>
 Crazy Marge's Bed Emporium
</p>
<ul>
 <li>
  3253435:
  Sleepeazy Mattresses
  <br/>
 </li>
</ul>
<ul>
 <li>
  5622345:
  CozyComfort Mattresses
  <br/>
 </li>
```

```
  </ul>
  <ul>
   <li>
    39981234:
    Floataway Waterbeds
    <br/>
   </li>
  </ul>
 </body>
</html>
```

```
Out[654]:  1312
```

In [5]:

```python
from IPython.display import IFrame
IFrame(src='examples/product3.html', width="100%", height="800px")
```

Out[5]:

Conners Chair Company

- QA3452: Queen Anne Chair
  --royal blue
  --flower print
  --seafoam green
  --teal
  --burgundy
- RC2342: Early American Rocking Chair
- BR3452: Bentwood Rocker

Wally's Wonderful World of Furniture

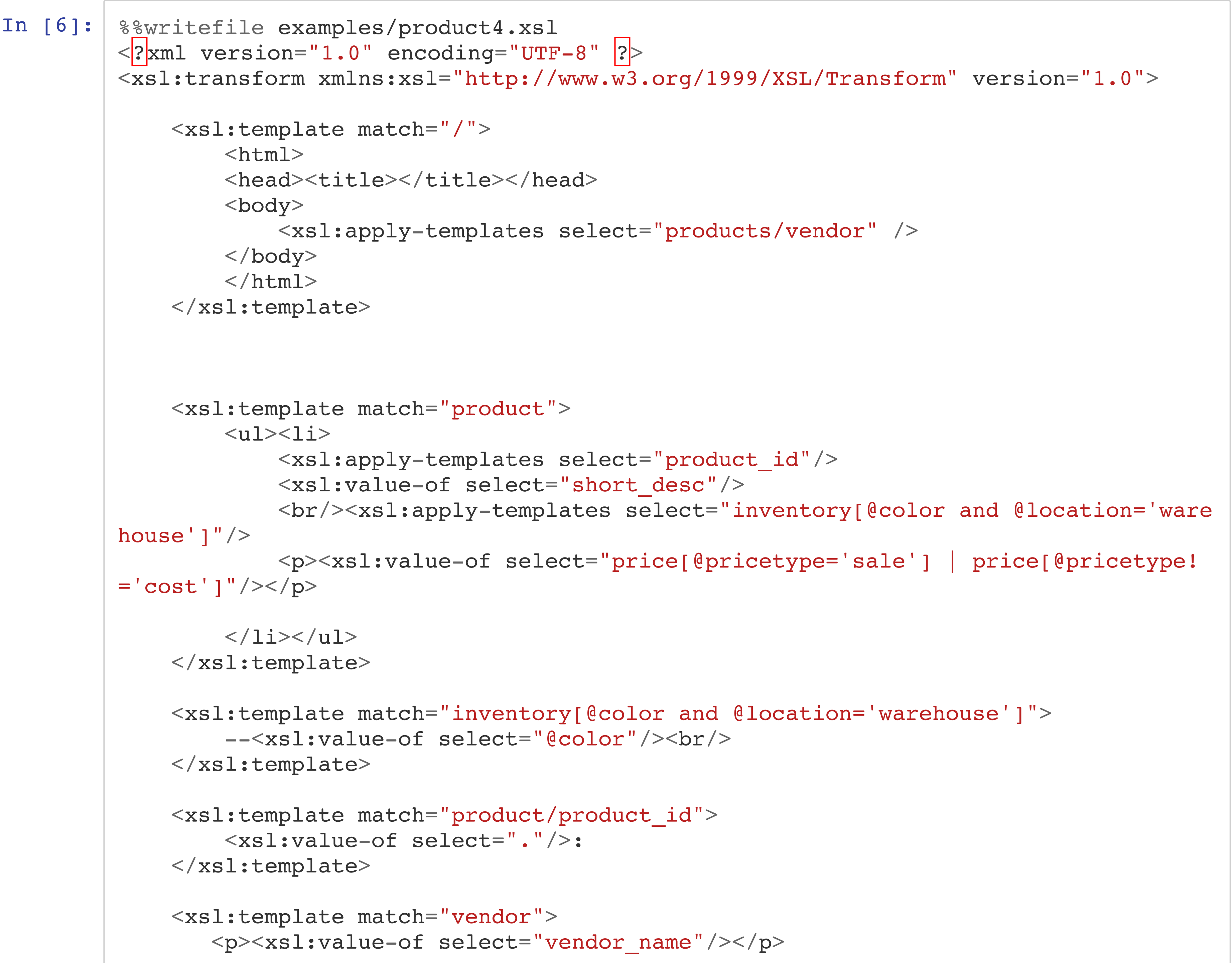
- WWWdrt: Dining Room Table
- WWWsc: Swivel Chair
- WWWhch: Hutch
- HallBench: Hall Bench
- SofaLoveSeat: Sofa and Love Seat
  --magnolia print
  --nautical print

Crazy Marge's Bed Emporium

- 3253435: Sleepeazy Mattresses
- 5622345: CozyComfort Mattresses
- 39981234: Floataway Waterbeds

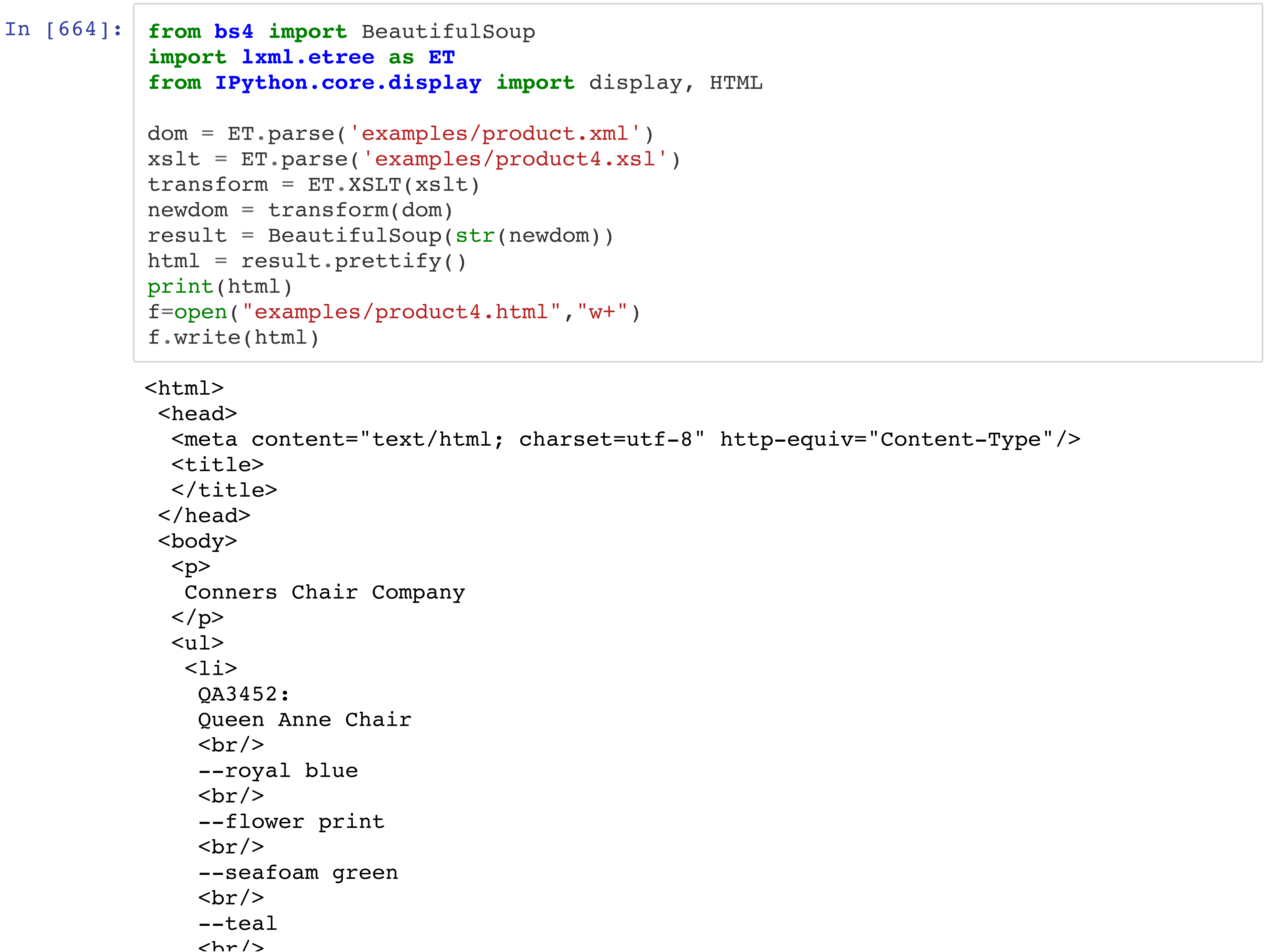
# VALUES AND LOGIC

- Let's select specific `price` based on attributes



In [6]:

```
%%writefile examples/product4.xsl
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

    <xsl:template match="/">
        <html>
        <head><title></title></head>
        <body>
            <xsl:apply-templates select="products/vendor" />
        </body>
        </html>
    </xsl:template>



    <xsl:template match="product">
        <ul><li>
            <xsl:apply-templates select="product_id"/>
            <xsl:value-of select="short_desc"/>
            <br/><xsl:apply-templates select="inventory[@color and @location='warehouse']"/>
            <p><xsl:value-of select="price[@pricetype='sale'] | price[@pricetype!='cost']"/></p>

        </li></ul>
    </xsl:template>

    <xsl:template match="inventory[@color and @location='warehouse']">
        --<xsl:value-of select="@color"/><br/>
    </xsl:template>

    <xsl:template match="product/product_id">
        <xsl:value-of select="."/>:
    </xsl:template>

    <xsl:template match="vendor">
       <p><xsl:value-of select="vendor_name"/></p>
```



In [664]:

```
from bs4 import BeautifulSoup
import lxml.etree as ET
from IPython.core.display import display, HTML

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product4.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(str(newdom))
html = result.prettify()
print(html)
f=open("examples/product4.html","w+")
f.write(html)
```

```
<html>
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>
  </title>
 </head>
 <body>
  <p>
   Conners Chair Company
  </p>
  <ul>
   <li>
    QA3452:
    Queen Anne Chair
    <br/>
    --royal blue
    <br/>
    --flower print
    <br/>
    --seafoam green
    <br/>
    --teal
    <br/>
  <br/>
  --burgundy
  <br/>
  <p>
   $125
  </p>
 </li>
</ul>
<ul>
 <li>
  RC2342:
  Early American Rocking Chair
  <br/>
  <p>
   $62
  </p>
 </li>
</ul>
<ul>
 <li>
  BR3452:
  Bentwood Rocker
  <br/>
  <p>
   $160
  </p>
 </li>
</ul>
<p>
 Wally's Wonderful World of Furniture
</p>
<ul>
 <li>
  WWWdrt:
  Dining Room Table
  <br/>
  <p>
   $145
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWsc:
  Swivel Chair
  <br/>
  <p>
   $45
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWhch:
  Hutch
  <br/>
  <p>
   $425
  </p>
 </li>
</ul>
<ul>
 <li>
  HallBench:
  Hall Bench
  <br/>
  <p>
   $62
  </p>
 </li>
</ul>
<ul>
 <li>
  SofaLoveSeat:
  Sofa and Love Seat
  <br/>
  --magnolia print
  <br/>
  --nautical print
  <br/>
  <p>
   $175
  </p>
 </li>
</ul>
<p>
 Crazy Marge's Bed Emporium
</p>
<ul>
 <li>
  3253435:
  Sleepeazy Mattresses
  <br/>
  <p>
   $300
  </p>
 </li>
</ul>
<ul>
 <li>
  5622345:
  CozyComfort Mattresses
  <br/>
  <p>
   starting at only $99.99
  </p>
 </li>
</ul>
<ul>
 <li>
  39981234:
  Floataway Waterbeds
  <br/>
  <p>
  </p>
```

```
   </li>
  </ul>
 </body>
</html>
```

```
Out[664]:  1615
```

In [7]:

```
from IPython.display import IFrame
IFrame(src='examples/product4.html', width="100%", height="800px")
```

Out[7]:

Conners Chair Company

- QA3452: Queen Anne Chair
  --royal blue
  --flower print
  --seafoam green
  --teal
  --burgundy

  $125

- RC2342: Early American Rocking Chair

  $62

- BR3452: Bentwood Rocker

  $160

Wally's Wonderful World of Furniture

- WWWdrt: Dining Room Table

  $145

- WWWsc: Swivel Chair

  $45

- WWWhch: Hutch

  $425

- HallBench: Hall Bench

  $62

- SofaLoveSeat: Sofa and Love Seat
  --magnolia print
  --nautical print

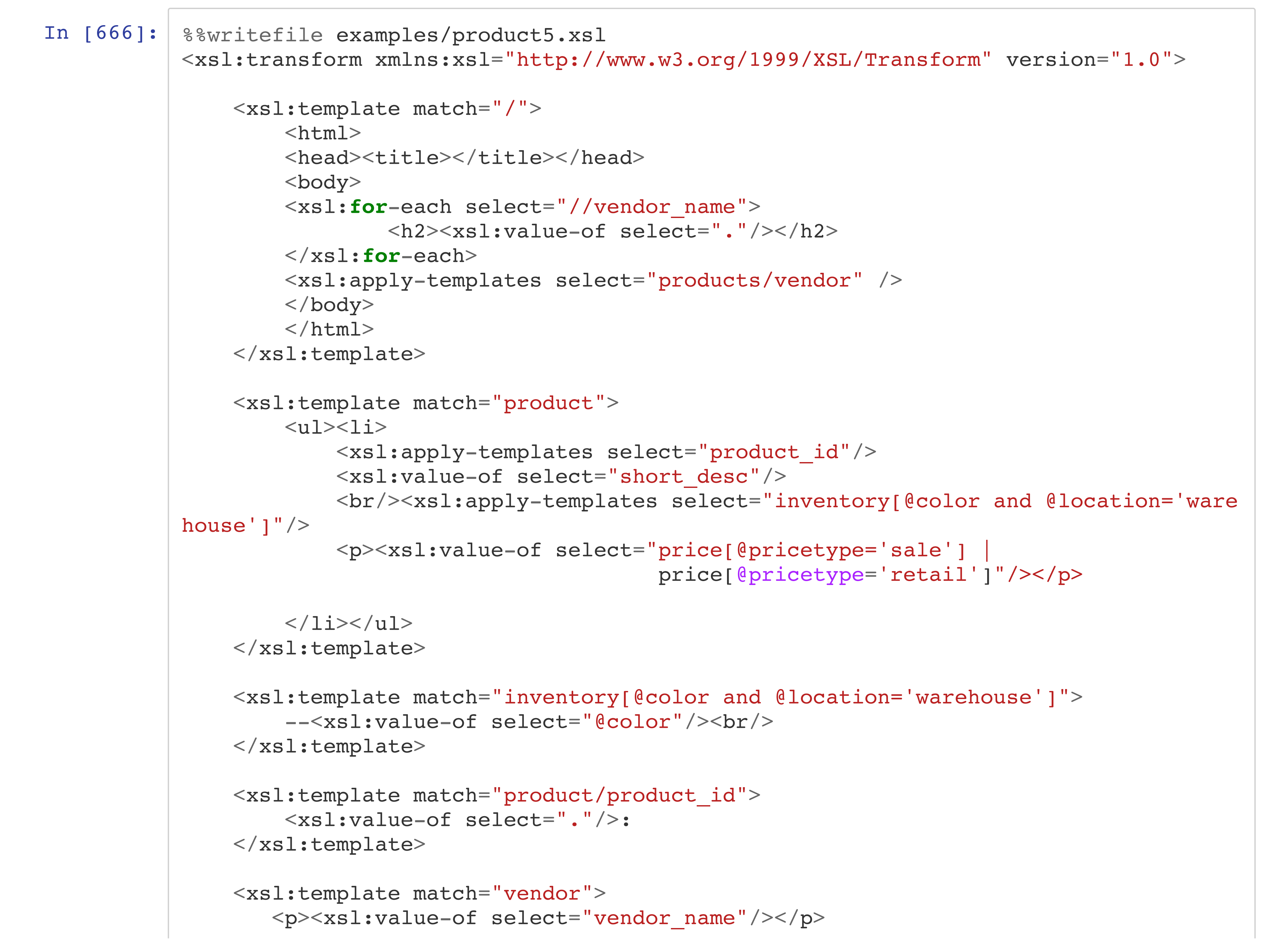
  $175

Crazy Marge's Bed Emporium

- 3253435: Sleepeazy Mattresses

# LOOPING

- Let's select specific `price` based on attributes



In [666]:

```
%%writefile examples/product5.xsl
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

    <xsl:template match="/">
        <html>
        <head><title></title></head>
        <body>
        <xsl:for-each select="//vendor_name">
                <h2><xsl:value-of select="."/></h2>
        </xsl:for-each>
        <xsl:apply-templates select="products/vendor" />
        </body>
        </html>
    </xsl:template>

    <xsl:template match="product">
        <ul><li>
            <xsl:apply-templates select="product_id"/>
            <xsl:value-of select="short_desc"/>
            <br/><xsl:apply-templates select="inventory[@color and @location='ware
house']"/>
            <p><xsl:value-of select="price[@pricetype='sale'] |
                                     price[@pricetype='retail']"/></p>

        </li></ul>
    </xsl:template>

    <xsl:template match="inventory[@color and @location='warehouse']">
        --<xsl:value-of select="@color"/><br/>
    </xsl:template>

    <xsl:template match="product/product_id">
        <xsl:value-of select="."/>:
    </xsl:template>

    <xsl:template match="vendor">
       <p><xsl:value-of select="vendor_name"/></p>
```

In [669]:

```
from bs4 import BeautifulSoup
import lxml.etree as ET
from IPython.core.display import display, HTML

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product5.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(str(newdom))
html = result.prettify()
print(html)
f=open("examples/product5.html","w+")
f.write(html)
```

```
<html>
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>
  </title>
 </head>
 <body>
  <h2>
   Conners Chair Company
  </h2>
  <h2>
   Wally's Wonderful World of Furniture
  </h2>
  <h2>
   Crazy Marge's Bed Emporium
  </h2>
  <p>
   Conners Chair Company
  </p>
  <ul>
   <li>
    QA3452:
```

```
  Queen Anne Chair
  <br/>
  --royal blue
  <br/>
  --flower print
  <br/>
  --seafoam green
  <br/>
  --teal
  <br/>
  --burgundy
  <br/>
  <p>
   $125
  </p>
 </li>
</ul>
<ul>
 <li>
  RC2342:
  Early American Rocking Chair
  <br/>
  <p>
   $62
  </p>
 </li>
</ul>
<ul>
 <li>
  BR3452:
  Bentwood Rocker
  <br/>
  <p>
   $160
  </p>
 </li>
</ul>
<p>
```

```
 Wally's Wonderful World of Furniture
</p>
<ul>
 <li>
  WWWdrt:
  Dining Room Table
  <br/>
  <p>
   $145
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWsc:
  Swivel Chair
  <br/>
  <p>
   $45
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWhch:
  Hutch
  <br/>
  <p>
   $425
  </p>
 </li>
</ul>
<ul>
 <li>
  HallBench:
  Hall Bench
  <br/>
  <p>
```

```
   $62
  </p>
 </li>
</ul>
<ul>
 <li>
  SofaLoveSeat:
  Sofa and Love Seat
  <br/>
  --magnolia print
  <br/>
  --nautical print
  <br/>
  <p>
   $175
  </p>
 </li>
</ul>
<p>
 Crazy Marge's Bed Emporium
</p>
<ul>
 <li>
  3253435:
  Sleepeazy Mattresses
  <br/>
  <p>
   $300
  </p>
 </li>
</ul>
<ul>
 <li>
  5622345:
  CozyComfort Mattresses
  <br/>
  <p>
  </p>
```

```
   </li>
  </ul>
  <ul>
   <li>
    39981234:
    Floataway Waterbeds
    <br/>
    <p>
    </p>
   </li>
  </ul>
 </body>
</html>
```

```
Out[669]:  1726
```

In [9]:

```
from IPython.display import IFrame
IFrame(src='examples/product5.html', width="100%", height="800px")
```

Out[9]:

# Conners Chair Company

# Wally's Wonderful World of Furniture

# Crazy Marge's Bed Emporium

Conners Chair Company

- QA3452: Queen Anne Chair
  --royal blue
  --flower print
  --seafoam green
  --teal
  --burgundy

  $125

- RC2342: Early American Rocking Chair

  $62

- BR3452: Bentwood Rocker

  $160

Wally's Wonderful World of Furniture

- WWWdrt: Dining Room Table

  $145

- WWWsc: Swivel Chair

  $45

- WWWhch: Hutch

  $425

- HallBench: Hall Bench

  $62

- SofaLoveSeat: Sofa and Love Seat

# SORTING

In [10]:

```
%%writefile examples/product6.xsl
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

    <xsl:template match="/">
        <html>
        <head><title></title></head>
        <body>
        <xsl:for-each select="//vendor_name">
                <xsl:sort select="."/>
                <h2><xsl:value-of select="."/></h2>
        </xsl:for-each>
        <xsl:apply-templates select="products/vendor" />
        </body>
        </html>
    </xsl:template>

    <xsl:template match="product">
        <ul><li>
            <xsl:apply-templates select="product_id"/>
            <xsl:value-of select="short_desc"/>
            <br/><xsl:apply-templates select="inventory[@color and @location='ware
house']"/>
            <p><xsl:value-of select="price[@pricetype='sale'] |
                                     price[@pricetype='retail']"/></p>

        </li></ul>
    </xsl:template>

    <xsl:template match="inventory[@color and @location='warehouse']">
        --<xsl:value-of select="@color"/><br/>
    </xsl:template>

    <xsl:template match="product/product_id">
        <xsl:value-of select="."/>:
    </xsl:template>
```

In [11]:

```
from bs4 import BeautifulSoup
import lxml.etree as ET
from IPython.core.display import display, HTML

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product6.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(str(newdom))
html = result.prettify()
print(html)
f=open("examples/product6.html","w+")
f.write(html)
```

```
<html>
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>
  </title>
 </head>
 <body>
  <h2>
   Conners Chair Company
  </h2>
  <h2>
   Crazy Marge's Bed Emporium
  </h2>
  <h2>
   Wally's Wonderful World of Furniture
  </h2>
  <p>
   Conners Chair Company
  </p>
  <ul>
   <li>
    QA3452:
```

```
  Queen Anne Chair
  <br/>
  --royal blue
  <br/>
  --flower print
  <br/>
  --seafoam green
  <br/>
  --teal
  <br/>
  --burgundy
  <br/>
  <p>
   $125
  </p>
 </li>
</ul>
<ul>
 <li>
  RC2342:
  Early American Rocking Chair
  <br/>
  <p>
   $62
  </p>
 </li>
</ul>
<ul>
 <li>
  BR3452:
  Bentwood Rocker
  <br/>
  <p>
   $160
  </p>
 </li>
</ul>
<p>
```

```
 Wally's Wonderful World of Furniture
</p>
<ul>
 <li>
  WWWdrt:
  Dining Room Table
  <br/>
  <p>
   $145
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWsc:
  Swivel Chair
  <br/>
  <p>
   $45
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWhch:
  Hutch
  <br/>
  <p>
   $425
  </p>
 </li>
</ul>
<ul>
 <li>
  HallBench:
  Hall Bench
  <br/>
  <p>
```

```
   $62
  </p>
 </li>
</ul>
<ul>
 <li>
  SofaLoveSeat:
  Sofa and Love Seat
  <br/>
  --magnolia print
  <br/>
  --nautical print
  <br/>
  <p>
   $175
  </p>
 </li>
</ul>
<p>
 Crazy Marge's Bed Emporium
</p>
<ul>
 <li>
  3253435:
  Sleepeazy Mattresses
  <br/>
  <p>
   $300
  </p>
 </li>
</ul>
<ul>
 <li>
  5622345:
  CozyComfort Mattresses
  <br/>
  <p>
  </p>
```

```
   </li>
  </ul>
  <ul>
   <li>
    39981234:
    Floataway Waterbeds
    <br/>
    <p>
    </p>
   </li>
  </ul>
 </body>
</html>
```

```
Out[11]: 1726
```

In [12]:

```python
from IPython.display import IFrame
IFrame(src='examples/product6.html', width="100%", height="800px")
```

Out[12]:

# Conners Chair Company

# Crazy Marge's Bed Emporium

# Wally's Wonderful World of Furniture

Conners Chair Company

- QA3452: Queen Anne Chair  
  --royal blue  
  --flower print  
  --seafoam green  
  --teal  
  --burgundy

  $125

- RC2342: Early American Rocking Chair

  $62

- BR3452: Bentwood Rocker

  $160

Wally's Wonderful World of Furniture

- WWWdrt: Dining Room Table

  $145

- WWWsc: Swivel Chair

  $45

- WWWhch: Hutch

  $425

- HallBench: Hall Bench

  $62

- SofaLoveSeat: Sofa and Love Seat

# CREATING ELEMENTS AND ATTRIBUTES

In [13]:

```
%%writefile examples/product7.xsl
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

    <xsl:template match="/">
        <html>
        <head><title></title></head>
        <body>
        <xsl:for-each select="//vendor_name">
                <xsl:sort select="."/>
                <xsl:element name="a">
                                <xsl:attribute name="href">
                                        #<xsl:value-of select="."/>
                                </xsl:attribute>
                                <h2><xsl:value-of select="."/></h2>
                        </xsl:element>
        </xsl:for-each>
        <xsl:apply-templates select="products/vendor" />
        </body>
        </html>
    </xsl:template>

    <xsl:template match="product">
        <ul><li>
            <xsl:apply-templates select="product_id"/>
            <xsl:value-of select="short_desc"/>
            <br/><xsl:apply-templates select="inventory[@color and @location='warehouse']"/>
            <p><xsl:value-of select="price[@pricetype='sale'] |
                                     price[@pricetype='retail']"/></p>

        </li></ul>
    </xsl:template>

    <xsl:template match="inventory[@color and @location='warehouse']">
        --<xsl:value-of select="@color"/><br/>
    </xsl:template>
```

In [14]:

```python
from bs4 import BeautifulSoup
import lxml.etree as ET
from IPython.core.display import display, HTML

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product7.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(str(newdom))
html = result.prettify()
print(html)
f=open("examples/product7.html","w+")
f.write(html)
```

```
<html>
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>
  </title>
 </head>
 <body>
  <a href="#Conners%20Chair%20Company">
   <h2>
    Conners Chair Company
   </h2>
  </a>
  <a href="#Crazy%20Marge's%20Bed%20Emporium">
   <h2>
    Crazy Marge's Bed Emporium
   </h2>
  </a>
  <a href="#Wally's%20Wonderful%20World%20of%20Furniture">
   <h2>
    Wally's Wonderful World of Furniture
   </h2>
  </a>
  <p>
```

```
<p>
 Conners Chair Company
</p>
<ul>
 <li>
  QA3452:
  Queen Anne Chair
  <br/>
  --royal blue
  <br/>
  --flower print
  <br/>
  --seafoam green
  <br/>
  --teal
  <br/>
  --burgundy
  <br/>
  <p>
   $125
  </p>
 </li>
</ul>
<ul>
 <li>
  RC2342:
  Early American Rocking Chair
  <br/>
  <p>
   $62
  </p>
 </li>
</ul>
<ul>
 <li>
  BR3452:
  Bentwood Rocker
  <br/>
  <p>
```

```
   $160
  </p>
 </li>
</ul>
<p>
 Wally's Wonderful World of Furniture
</p>
<ul>
 <li>
  WWWdrt:
  Dining Room Table
  <br/>
  <p>
   $145
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWsc:
  Swivel Chair
  <br/>
  <p>
   $45
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWhch:
  Hutch
  <br/>
  <p>
   $425
  </p>
 </li>
</ul>
<ul>
```

```
 <li>
  HallBench:
  Hall Bench
  <br/>
  <p>
   $62
  </p>
 </li>
</ul>
<ul>
 <li>
  SofaLoveSeat:
  Sofa and Love Seat
  <br/>
  --magnolia print
  <br/>
  --nautical print
  <br/>
  <p>
   $175
  </p>
 </li>
</ul>
<p>
 Crazy Marge's Bed Emporium
</p>
<ul>
 <li>
  3253435:
  Sleepeazy Mattresses
  <br/>
  <p>
   $300
  </p>
 </li>
</ul>
<ul>
 <li>
```

```
    5622345:
    CozyComfort Mattresses
    <br/>
    <p>
    </p>
   </li>
  </ul>
  <ul>
   <li>
    39981234:
    Floataway Waterbeds
    <br/>
    <p>
    </p>
   </li>
  </ul>
 </body>
</html>
```

```
Out[14]:  1902
```

In [15]:

```
from IPython.display import IFrame
IFrame(src='examples/product7.html', width="100%", height="800px")
```

Out[15]:

# Conners Chair Company

# Crazy Marge's Bed Emporium

# Wally's Wonderful World of Furniture

Conners Chair Company

- QA3452: Queen Anne Chair
  --royal blue
  --flower print
  --seafoam green
  --teal
  --burgundy

  $125

- RC2342: Early American Rocking Chair

  $62

- BR3452: Bentwood Rocker

  $160

Wally's Wonderful World of Furniture

- WWWdrt: Dining Room Table

  $145

- WWWsc: Swivel Chair

  $45

- WWWhch: Hutch

  $425

- HallBench: Hall Bench

  $62

- SofaLoveSeat: Sofa and Love Seat

# CONDITIONALS: IF

In [16]:

```
%%writefile examples/product8.xsl
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

    <xsl:template match="/">
        <html>
        <head><title></title></head>
        <body>
        <xsl:for-each select="//vendor_name">
                <xsl:sort select="."/>
                <xsl:if test="../@webvendor!='no'">
                        <xsl:element name="a">
                                <xsl:attribute name="href">
                                        #<xsl:value-of select="."/>
                                </xsl:attribute>
                                <h2><xsl:value-of select="."/></h2>
                        </xsl:element>
                </xsl:if>
        </xsl:for-each>
        <xsl:for-each select="//vendor">
                <xsl:sort select="vendor_name"/>
                   <xsl:if test="@webvendor!='no'">
                        <p><xsl:value-of select="vendor_name"/></p>
                        <xsl:apply-templates select=".//product"/>
                </xsl:if>
        </xsl:for-each>

        </body>
        </html>
    </xsl:template>

    <xsl:template match="product">
        <ul><li>
            <xsl:apply-templates select="product_id"/>
            <xsl:value-of select="short_desc"/>
            <br/><xsl:apply-templates select="inventory[@color and @location='warehouse']"/>
            <p><xsl:value-of select="price[@pricetype='sale'] |
```

In [17]:

```python
from bs4 import BeautifulSoup
import lxml.etree as ET
from IPython.core.display import display, HTML

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product8.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(str(newdom))
html = result.prettify()
print(html)
f=open("examples/product8.html","w+")
f.write(html)
```

```
<html>
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>
  </title>
 </head>
 <body>
  <a href="#Conners%20Chair%20Company">
   <h2>
    Conners Chair Company
   </h2>
  </a>
  <a href="#Wally's%20Wonderful%20World%20of%20Furniture">
   <h2>
    Wally's Wonderful World of Furniture
   </h2>
  </a>
  <p>
   Conners Chair Company
  </p>
  <ul>
   <li>
    QA3452:
```

```
  QA3452:
  Queen Anne Chair
  <br/>
  --royal blue
  <br/>
  --flower print
  <br/>
  --seafoam green
  <br/>
  --teal
  <br/>
  --burgundy
  <br/>
  <p>
   $125
  </p>
 </li>
</ul>
<ul>
 <li>
  RC2342:
  Early American Rocking Chair
  <br/>
  <p>
   $62
  </p>
 </li>
</ul>
<ul>
 <li>
  BR3452:
  Bentwood Rocker
  <br/>
  <p>
   $160
  </p>
 </li>
</ul>
<p>
```

```
 Wally's Wonderful World of Furniture
</p>
<ul>
 <li>
  WWWdrt:
  Dining Room Table
  <br/>
  <p>
   $145
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWsc:
  Swivel Chair
  <br/>
  <p>
   $45
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWhch:
  Hutch
  <br/>
  <p>
   $425
  </p>
 </li>
</ul>
<ul>
 <li>
  HallBench:
  Hall Bench
  <br/>
  <p>
```

```
     $62
    </p>
   </li>
  </ul>
  <ul>
   <li>
    SofaLoveSeat:
    Sofa and Love Seat
    <br/>
    --magnolia print
    <br/>
    --nautical print
    <br/>
    <p>
     $175
    </p>
   </li>
  </ul>
 </body>
</html>
```

```
Out[17]: 1454
```

In [18]:

```
from IPython.display import IFrame
IFrame(src='examples/product8.html', width="100%", height="800px")
```

Out[18]:

# Conners Chair Company

# Wally's Wonderful World of Furniture

Conners Chair Company

- QA3452: Queen Anne Chair
  --royal blue
  --flower print
  --seafoam green
  --teal
  --burgundy

  $125

- RC2342: Early American Rocking Chair

  $62

- BR3452: Bentwood Rocker

  $160

Wally's Wonderful World of Furniture

- WWWdrt: Dining Room Table

  $145

- WWWsc: Swivel Chair

  $45

- WWWhch: Hutch

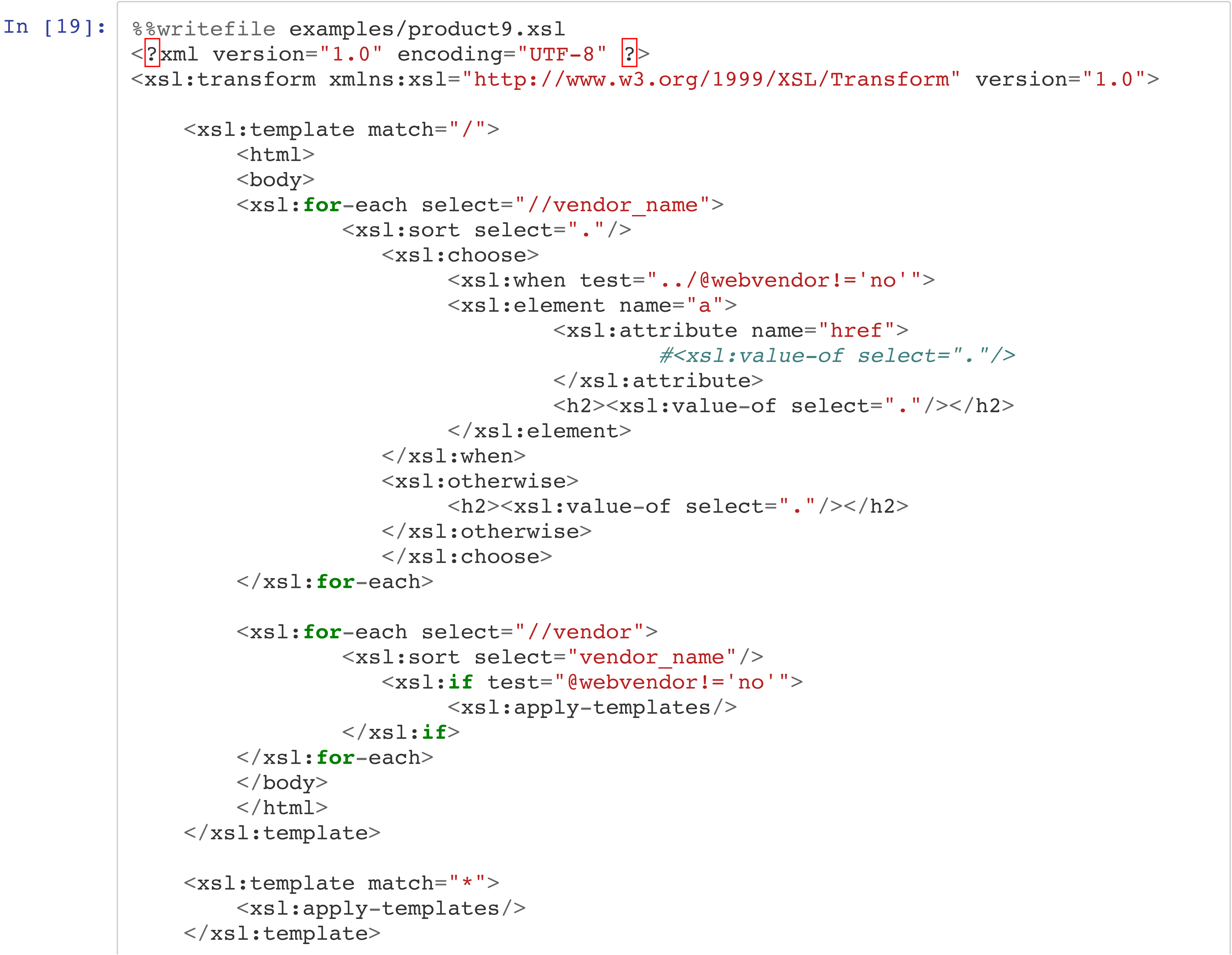
$425

- HallBench: Hall Bench

  $62

- SofaLoveSeat: Sofa and Love Seat
  --magnolia print
  --nautical print

# CONDITIONALS: CHOOSE



In [19]:

```
%%writefile examples/product9.xsl
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

    <xsl:template match="/">
        <html>
        <body>
        <xsl:for-each select="//vendor_name">
                <xsl:sort select="."/>
                   <xsl:choose>
                        <xsl:when test="../@webvendor!='no'">
                        <xsl:element name="a">
                                <xsl:attribute name="href">
                                        #<xsl:value-of select="."/>
                                </xsl:attribute>
                                <h2><xsl:value-of select="."/></h2>
                        </xsl:element>
                   </xsl:when>
                   <xsl:otherwise>
                        <h2><xsl:value-of select="."/></h2>
                   </xsl:otherwise>
                   </xsl:choose>
        </xsl:for-each>

        <xsl:for-each select="//vendor">
                <xsl:sort select="vendor_name"/>
                   <xsl:if test="@webvendor!='no'">
                        <xsl:apply-templates/>
                </xsl:if>
        </xsl:for-each>
        </body>
        </html>
    </xsl:template>

    <xsl:template match="*">
        <xsl:apply-templates/>
    </xsl:template>
    <xsl:template match="text()">
        <xsl:apply-templates/>
    </xsl:template>

    <xsl:template match="product">
        <ul><li>
            <xsl:apply-templates select="product_id"/>
            <xsl:value-of select="short_desc"/>
            <br/><xsl:apply-templates select="inventory[@color and @location='ware
house']"/>
            <p><xsl:value-of select=".//price[@pricetype='sale'] |
                                     .//price[@pricetype='retail']"/></p>

        </li></ul>
    </xsl:template>

    <xsl:template match="inventory[@color and @location='warehouse']">
        --<xsl:value-of select="@color"/><br/>
    </xsl:template>

    <xsl:template match="product/product_id">
        <xsl:value-of select="."/>:
    </xsl:template>

    <xsl:template match="vendor_name">
        <xsl:if test="../@webvendor!='no'">
        <xsl:element name="a">
                <xsl:attribute name="name"><xsl:value-of select="."/>
                </xsl:attribute>
                <p><xsl:value-of select="."/></p>
        </xsl:element>
        </xsl:if>
    </xsl:template>

</xsl:transform>
```

In [20]:

```python
from bs4 import BeautifulSoup
import lxml.etree as ET
from IPython.core.display import display, HTML

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product9.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(str(newdom))
html = result.prettify()
print(html)
f=open("examples/product9.html","w+")
f.write(html)
```

```
<html>
 <body>
  <a href="#Conners%20Chair%20Company">
   <h2>
    Conners Chair Company
   </h2>
  </a>
  <h2>
   Crazy Marge's Bed Emporium
  </h2>
  <a href="#Wally's%20Wonderful%20World%20of%20Furniture">
   <h2>
    Wally's Wonderful World of Furniture
   </h2>
  </a>
  <a name="Conners%20Chair%20Company">
   <p>
    Conners Chair Company
   </p>
  </a>
  <ul>
   <li>
    QA3452:
```

```
  QA3452:
  Queen Anne Chair
  <br/>
  --royal blue
  <br/>
  --flower print
  <br/>
  --seafoam green
  <br/>
  --teal
  <br/>
  --burgundy
  <br/>
  <p>
   $125
  </p>
 </li>
</ul>
<ul>
 <li>
  RC2342:
  Early American Rocking Chair
  <br/>
  <p>
   $62
  </p>
 </li>
</ul>
<ul>
 <li>
  BR3452:
  Bentwood Rocker
  <br/>
  <p>
   $160
  </p>
 </li>
</ul>
<a name="Wally's%20Wonderful%20World%20of%20Furniture">
```

```
 <p>
  Wally's Wonderful World of Furniture
 </p>
</a>
<ul>
 <li>
  WWWdrt:
  Dining Room Table
  <br/>
  <p>
   $145
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWsc:
  Swivel Chair
  <br/>
  <p>
   $45
  </p>
 </li>
</ul>
<ul>
 <li>
  WWWhch:
  Hutch
  <br/>
  <p>
   $425
  </p>
 </li>
</ul>
<ul>
 <li>
  HallBench:
  Hall Bench
```

```
    <br/>
    <p>
     $62
    </p>
   </li>
  </ul>
  <ul>
   <li>
    SofaLoveSeat:
    Sofa and Love Seat
    <br/>
    --magnolia print
    <br/>
    --nautical print
    <br/>
    <p>
     $175
    </p>
   </li>
  </ul>
 </body>
</html>
```

```
Out[20]: 1507
```

In [21]:

```
from IPython.display import IFrame
IFrame(src='examples/product9.html', width="100%", height="800px")
```

Out[21]:

# Conners Chair Company

# Crazy Marge's Bed Emporium

# Wally's Wonderful World of Furniture

Conners Chair Company

- QA3452: Queen Anne Chair
  --royal blue
  --flower print
  --seafoam green
  --teal
  --burgundy

  $125

- RC2342: Early American Rocking Chair

  $62

- BR3452: Bentwood Rocker

  $160

Wally's Wonderful World of Furniture

- WWWdrt: Dining Room Table

  $145

- WWWsc: Swivel Chair

  $45

- WWWhch: Hutch

  $425

- HallBench: Hall Bench

  $62

- SofaLoveSeat: Sofa and Love Seat

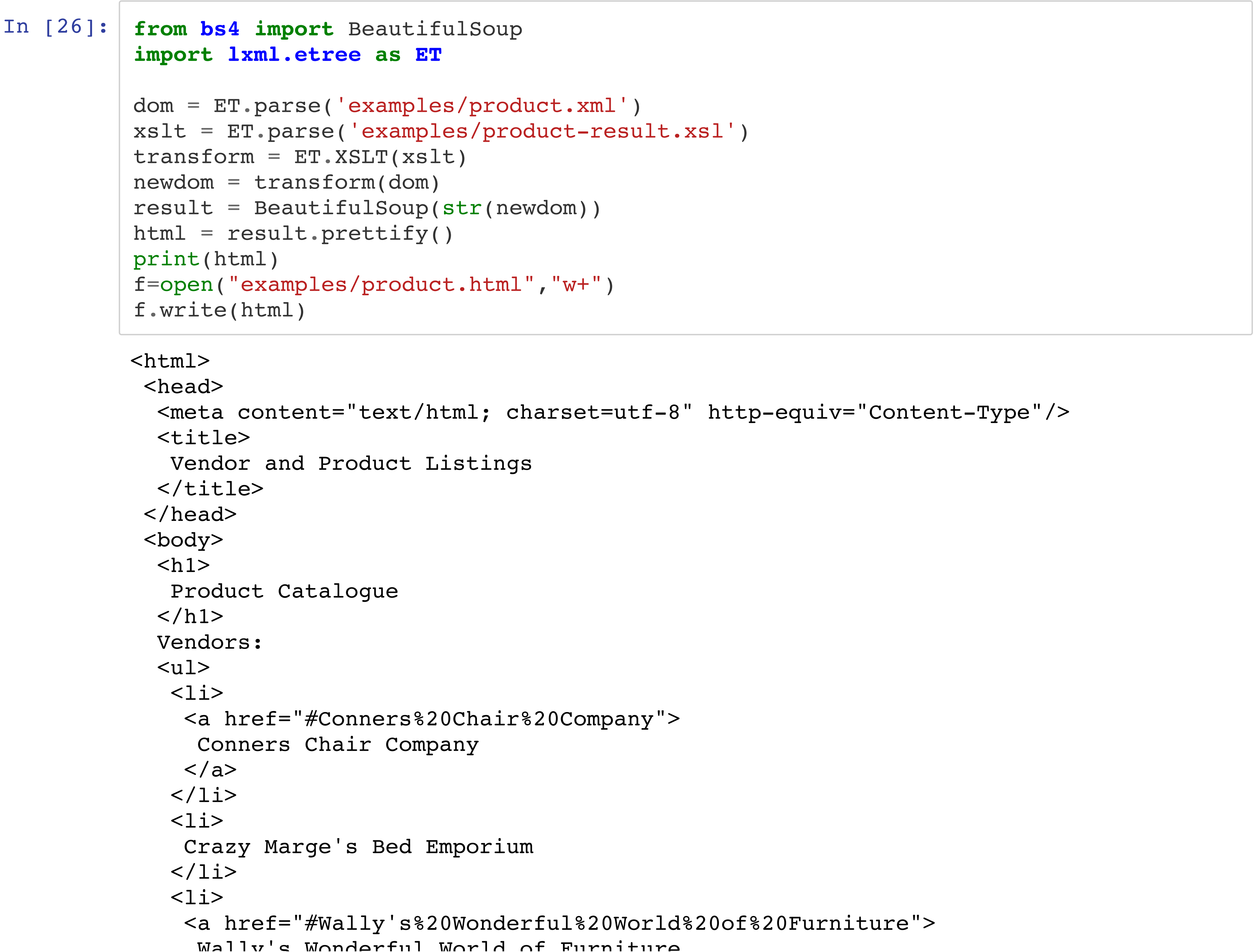
# FINAL FILE

In [22]:

```
%%writefile examples/product-result.xsl
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

    <xsl:template match="/">
    <html>
    <head><title>Vendor and Product Listings</title></head>
    <body>
    <h1>Product Catalogue</h1>
        Vendors:
        <ul>
        <xsl:for-each select="//vendor_name">
                <xsl:sort select="."/>
                   <xsl:choose>
                        <xsl:when test="../@webvendor!='no'">
                        <li>
                        <xsl:element name="a">
                                <xsl:attribute name="href">
                                        #<xsl:value-of select="."/>
                                </xsl:attribute>
                                <xsl:value-of select="."/>
                        </xsl:element>
                        </li>
                   </xsl:when>
                   <xsl:otherwise>
                        <li><xsl:value-of select="."/></li>
                   </xsl:otherwise>
                   </xsl:choose>
        </xsl:for-each>
        </ul>

        <xsl:for-each select="//vendor">
                <xsl:sort select="vendor_name"/>
                   <xsl:if test="@webvendor!='no'">
                        <xsl:apply-templates/>
                   </xsl:if>
    </xsl:for-each>
</body>
</html>
</xsl:template>

<xsl:template match="*">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="text()">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="product">
    <ul><li>
        <xsl:value-of select="short_desc"/>
        <xsl:apply-templates select="product_id"/>
        <xsl:if test="inventory[@color]">
            <p>Available colours:
            <br/><xsl:apply-templates select="inventory[@color]"/>
            </p>
        </xsl:if>
        <p>Price: <xsl:value-of select=".//price[@pricetype='sale'] |
                                 .//price[@pricetype='retail']"/></p>

    </li></ul>
</xsl:template>

<xsl:template match="inventory[@color and @location='warehouse']">
    --<xsl:value-of select="@color"/><br/>
</xsl:template>

<xsl:template match="product/product_id">
    (<xsl:value-of select="."/>)
</xsl:template>

<xsl:template match="vendor_name">
    <xsl:if test="../@webvendor!='no'">
```

```
        <xsl:element name="a">
                <xsl:attribute name="name"><xsl:value-of select="."/>
                </xsl:attribute>
                <h3><xsl:value-of select="."/></h3>
        </xsl:element>
        </xsl:if>
    </xsl:template>

</xsl:transform>
```

```
Overwriting examples/product-result.xsl
```



In [26]:

```
from bs4 import BeautifulSoup
import lxml.etree as ET

dom = ET.parse('examples/product.xml')
xslt = ET.parse('examples/product-result.xsl')
transform = ET.XSLT(xslt)
newdom = transform(dom)
result = BeautifulSoup(str(newdom))
html = result.prettify()
print(html)
f=open("examples/product.html","w+")
f.write(html)
```

```
<html>
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>
   Vendor and Product Listings
  </title>
 </head>
 <body>
  <h1>
   Product Catalogue
  </h1>
  Vendors:
  <ul>
   <li>
    <a href="#Conners%20Chair%20Company">
     Conners Chair Company
    </a>
   </li>
   <li>
    Crazy Marge's Bed Emporium
   </li>
   <li>
    <a href="#Wally's%20Wonderful%20World%20of%20Furniture">
     Wally's Wonderful World of Furniture
   Wally's Wonderful World of Furniture
  </a>
 </li>
</ul>
<a name="Conners%20Chair%20Company">
 <h3>
  Conners Chair Company
 </h3>
</a>
<ul>
 <li>
  Queen Anne Chair
      (QA3452)
  <p>
   Available colours:
   <br/>
   --royal blue
   <br/>
   --flower print
   <br/>
   --seafoam green
   <br/>
   --teal
   <br/>
   --burgundy
   <br/>
  </p>
  <p>
   Price: $125
  </p>
 </li>
</ul>
<ul>
 <li>
  Early American Rocking Chair
      (RC2342)
  <p>
   Price: $62
  </p>
 </li>
</ul>
<ul>
 <li>
  Bentwood Rocker
      (BR3452)
  <p>
   Price: $160
  </p>
 </li>
</ul>
<a name="Wally's%20Wonderful%20World%20of%20Furniture">
 <h3>
  Wally's Wonderful World of Furniture
 </h3>
</a>
<ul>
 <li>
  Dining Room Table
      (WWWdrt)
  <p>
   Price: $145
  </p>
 </li>
</ul>
<ul>
 <li>
  Swivel Chair
      (WWWsc)
  <p>
   Price: $45
  </p>
 </li>
</ul>
<ul>
 <li>
  Hutch
      (WWWhch)
```

```
    <p>
     Price: $425
    </p>
   </li>
  </ul>
  <ul>
   <li>
    Hall Bench
        (HallBench)
    <p>
     Price: $62
    </p>
   </li>
  </ul>
  <ul>
   <li>
    Sofa and Love Seat
        (SofaLoveSeat)
    <p>
     Available colours:
     <br/>
     --magnolia print
     <br/>
     --nautical print
     <br/>
    </p>
    <p>
     Price: $175
    </p>
   </li>
  </ul>
 </body>
</html>
```

```
Out[26]:  1860
```

In [1]:

```
from IPython.display import IFrame
IFrame(src='examples/product.html', width="100%", height="800px")
```

Out[1]:

# Product Catalogue

Vendors:

- Conners Chair Company
- Crazy Marge's Bed Emporium
- Wally's Wonderful World of Furniture

## Conners Chair Company

- Queen Anne Chair (QA3452)

  Available colours:<br>--royal blue<br>--flower print<br>--seafoam green<br>--teal<br>--burgundy

  Price: $125

- Early American Rocking Chair (RC2342)

  Price: $62

- Bentwood Rocker (BR3452)

  Price: $160

## Wally's Wonderful World of Furniture

- Dining Room Table (WWWdrt)

  Price: $145

- Swivel Chair (WWWsc)

  Price: $45

- Hutch (WWWhch)

  Price: $425

- Hall Bench (HallBench)

# TUTORIAL