


XML Syntax and Editors/Validator

Keywords: Syntax, EditorEditor – XMLViewer, XML Notepad, XML Validator

<https://codebeautify.org/xmlviewer#>

 **Code Beautify**

JSON Formatter | Hex Color Codes | HMAC Generator | My Ip | Search | Recent Links | More ▾ | Sign in |

XML Viewer★

Save & Share

XML Input

sample 🗑️ ↗️

```
1 <?xml version="1.0" encoding="euc-kr" standalone="yes"?>
2 <?xml-stylesheet type="text/css" href="memo1.css"?>
3 <MEMO>
4   <TO> To: Eugene</TO>
5   <FROM> From: Hong, Gil-Dong</FROM>
6   <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
7 </MEMO>
8
9
10
```

Load Url

Browse


Tree View

Beautify / Format

Result : XML Tree View

🗑️ ↗️

```
memo ..
├── to To: Eugene
├── from From: Hong, Gil-Dong
└── contents Can we dine out at 1 PM?
```

 **Code Beautify**

JSON Formatter | Hex Color Codes | HMAC Generator | My Ip | Search | Recent Links | More ▾ | %EC%8.. ▾ |

XML Viewer☆

Save & Share

XML Input

sample

```
1 <?xml version="1.0" encoding="euc-kr" standalone="yes"?>
2 <?xml-stylesheet type="text/css" href="memo1.css"?>
3 <MEMO>
4   <TO> To: Eugene</TO>
5   <FROM> From: Hong, Gil-Dong</FROM>
6   <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
7 </MEMO>
8
9
10
```

Load Url

Browse


Tree View

Beautify / Format

Minify

Result : Beautify XML

```
1 <?xml version="1.0" encoding="euc-kr" standalone="yes"?>
2 <?xml-stylesheet type="text/css" href="memo1.css"?>
3 <MEMO>
4   <TO> To: Eugene</TO>
5   <FROM> From: Hong, Gil-Dong</FROM>
6   <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
7 </MEMO>
8
9
10
```

 **Code Beautify**

JSON Formatter | Hex Color Codes | HMAC Generator | My Ip | Search | Recent Links | More ▾ | %EC%8.. ▾ |

XML Viewer☆

Save & Share

XML Input

sample

```
1 <?xml version="1.0" encoding="euc-kr" standalone="yes"?>
2 <?xml-stylesheet type="text/css" href="memo1.css"?>
3 <MEMO>
4   <TO> To: Eugene</TO>
5   <FROM> From: Hong, Gil-Dong</FROM>
6   <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
7 </MEMO>
8
9
10
```

Load Url

Browse

Tree View

Beautify / Format

Minify

Result : Minify XML

```
1 <?xml version="1.0" encoding="euc-kr" standalone="yes"
  ?><?xml-stylesheet type="text/css" href="memo1.css"
  ?><MEMO><TO> To: Eugene</TO><FROM> From: Hong, Gil
  -Dong</FROM><CONTENTS> Can we dine out at 1 PM
  ?</CONTENTS></MEMO>
2
3
4
```







<https://jsonformatter.org/xml-viewer>

{ JSON formatter }

XML Viewer

XML FORMATTER JSBEAUTIFIER SAVE RECENT LINKS %EC%84%B1%EA%B8%88%EC%98%

Input XML

Sample      

1 <?xml version="1.0" encoding="euc-kr" standalone="yes"?>

2 <?xml-stylesheet type="text/css" href="memo1.css"?>

3 <MEMO>

4 <TO> To: Eugene</TO>

5 <FROM> From: Hong, Gil-Dong</FROM>

6 <CONTENTS> Can we meet tomorrow?</CONTENTS>

7 </MEMO>

8

9

10

Load Data

Validate

XML Viewer

Format / Beautify

Minify / Compact

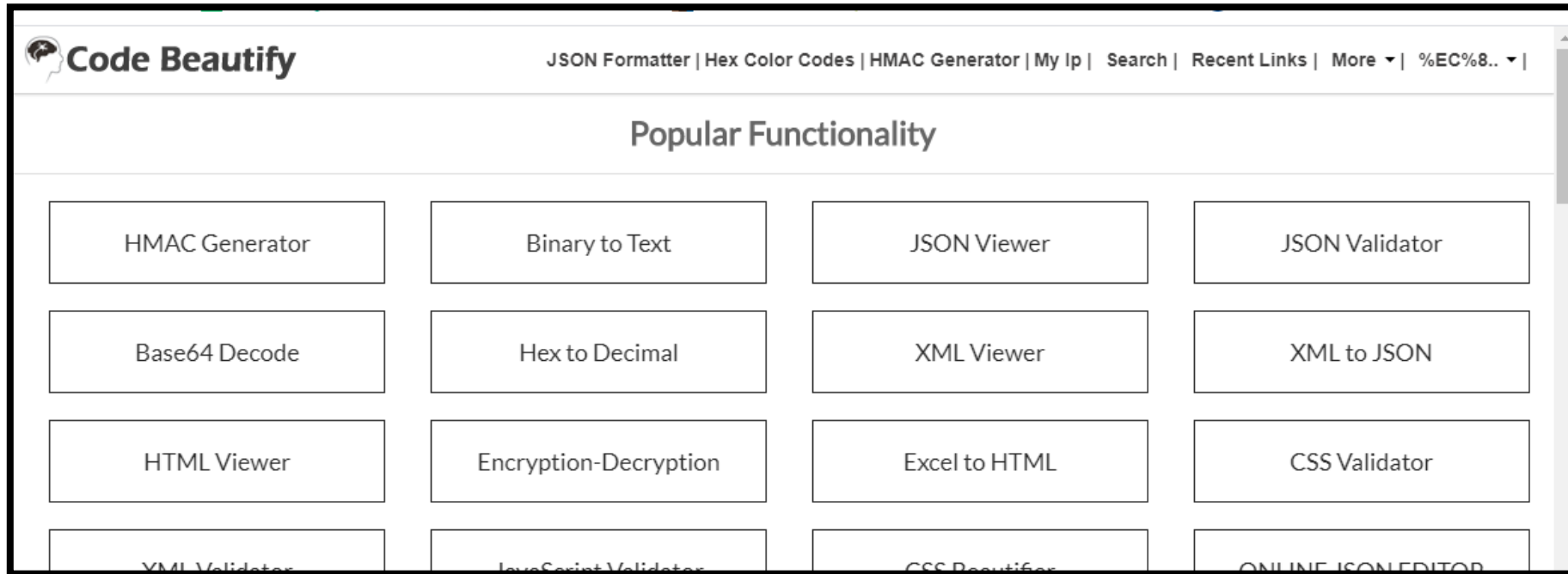
Download

Output XML


1 Valid XML

Sign in to Code beautify

<https://codebeautify.org/>



HTML Viewer after signing in

 **Code Beautify**

JSON Formatter | Hex Color Codes | HMAC Generator | My Ip | Search | Recent Links | More ▾ | %EC%8.. ▾ |

HTML Viewer★

Save & Share

HTML Input

sample

1 ▾

<HTML>

2 ▾

<HEAD>

3

<TITLE>MEMO</TITLE>

4

</HEAD>

5 ▾

<BODY>

6 ▾

<P style = "font-weight:bold">To : Eugene

7 ▾

<P style = "font-weight:bold">From : Hong, Gil

-Dong

8 ▾

<P style = "font-weight:bold; color: red"><em

9

>Can we dine out at 1 PM?

9

</BODY>

10

</HTML>

11

Load Url

Browse

Default Tab Sp: ▾

Beautify

Minify

Run

Result : Beautify HTML

1 ▾

<HTML>

2 ▾

<HEAD>

3

<TITLE>MEMO</TITLE>

4

</HEAD>

5 ▾

<BODY>

6 ▾

<P style = "font-weight:bold">To : Eugene

7

8 ▾

<P style = "font-weight:bold">From

9

10 ▾

<P style = "font-weight:bold; c

11 ▾

Can we dine out at 1 PM

12 ▾

13

</BODY>

14

</HTML>

15

HTML Viewer☆

[Save](#)

HTML Input

[sample](#)  

```
1 <HTML>
2 <HEAD>
3 <TITLE>MEMO</TITLE>
4 </HEAD>
5 <BODY>
6 <P style = "font-weight:bold">To : Eugene
7 <P style = "font-weight:bold">From : Hong, Gil
  -Dong
8 <P style = "font-weight:bold; color: red"><em
  >Can we dine out at 1 PM?<em>
9 </BODY>
10 </HTML>
11
```

[Load Url](#)[Browse](#)Default Tab Sp: [Beautify](#)[Minify](#)[Run](#)

Result : HTML Output

To : Eugene

From : Hong, Gil-Dong

Can we dine out at 1 PM?

Code Beautify

[JSON Formatter](#) | [Hex Color Codes](#) | [HMAC Generator](#) | [My Ip](#) | [Search](#) | [Recent Links](#) | [More](#) ▾ | [%EC%8..](#) ▾ |

HTML Viewer☆

[Save & Share](#)

HTML Input

[sample](#)  

```
1 <HTML>
2 <HEAD>
3 <TITLE>MEMO</TITLE>
4 </HEAD>
5 <BODY>
6 <P style = "font-weight:bold">To : Eugene
7 <P style = "font-weight:bold">From : Hong, Gil
  -Dong
8 <P style = "font-weight:bold; color: red"><em
  >Can we dine out at 1 PM?<em>
9 </BODY>
10 </HTML>
11
```

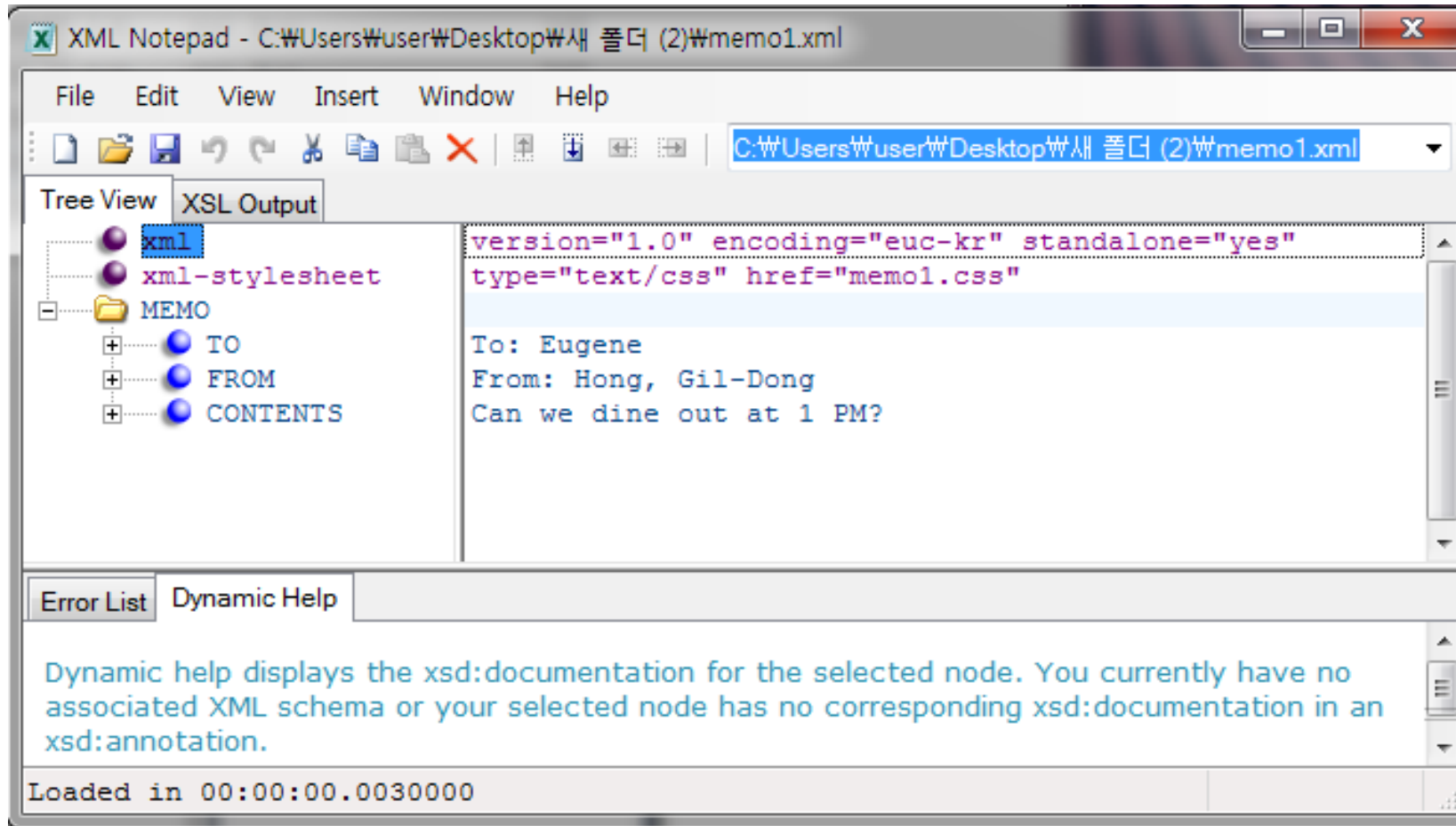
[Load Url](#)[Browse](#)Default Tab Sp: [Beautify](#)[Minify](#)[Run](#)

Result : Compact/Minify HTML

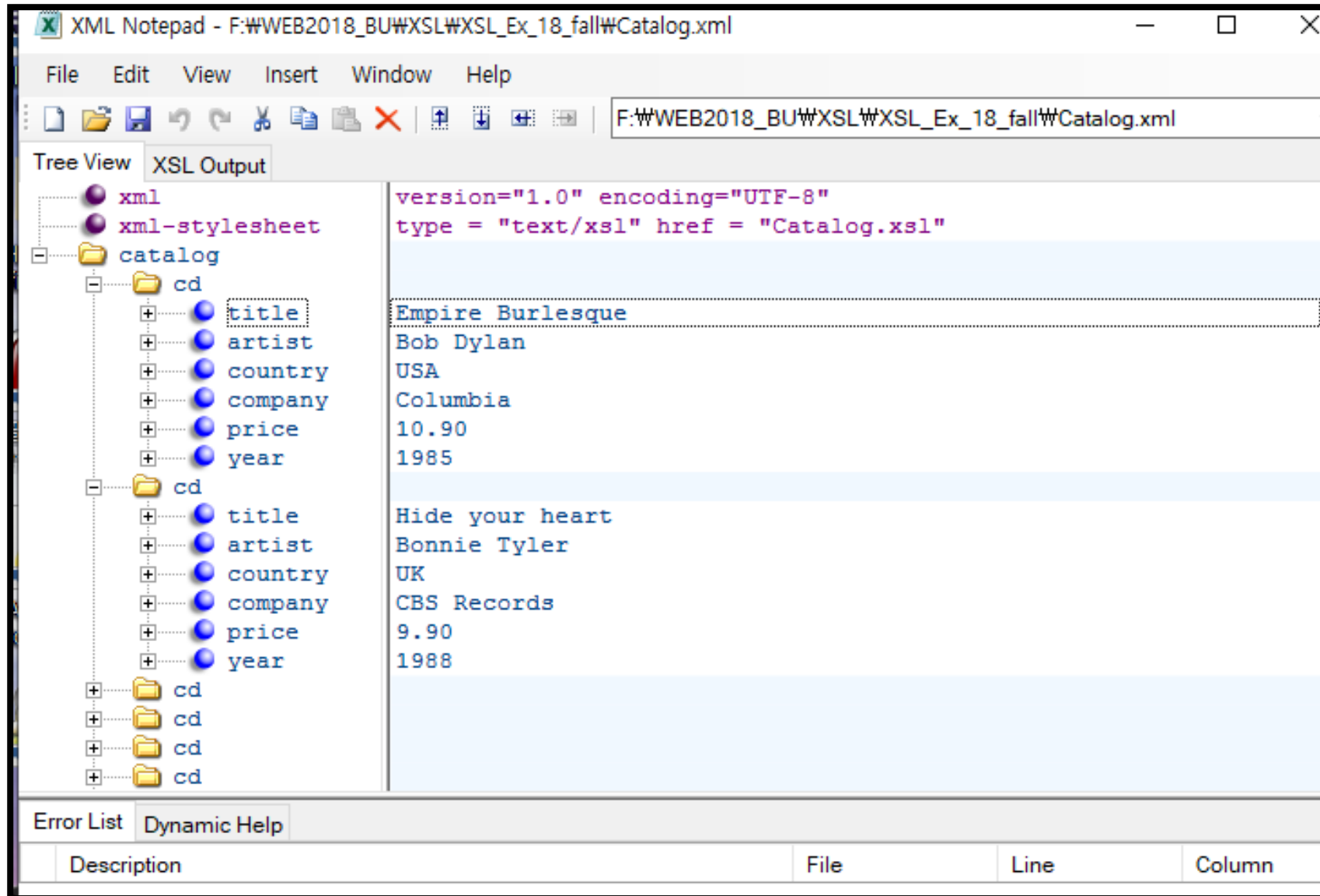
[sample](#)  

```
1 <HTML><HEAD><TITLE>MEMO</TITLE></HEAD><BODY><P
  style = "font-weight:bold">To : Eugene
2 <P style = "font-weight:bold">From : Hong, Gil
  -Dong
3 <P style = "font-weight:bold; color: red"><em
  >Can we dine out at 1 PM?<em></BODY></HTML>
4
```

XML Notepad from MS for Windows



XML Notepad supports XSL running



XML Notepad - F:\WEB2018_BU\XSL\XSL_Ex_18_fall\Catalog.xml

File Edit View Insert Window Help

F:\WEB2018_BU\XSL\XSL_Ex_18_fall\Catalog.xml

Tree View XSL Output

XSLT Location: Catalog.xsl Transform

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli

Error List Dynamic Help

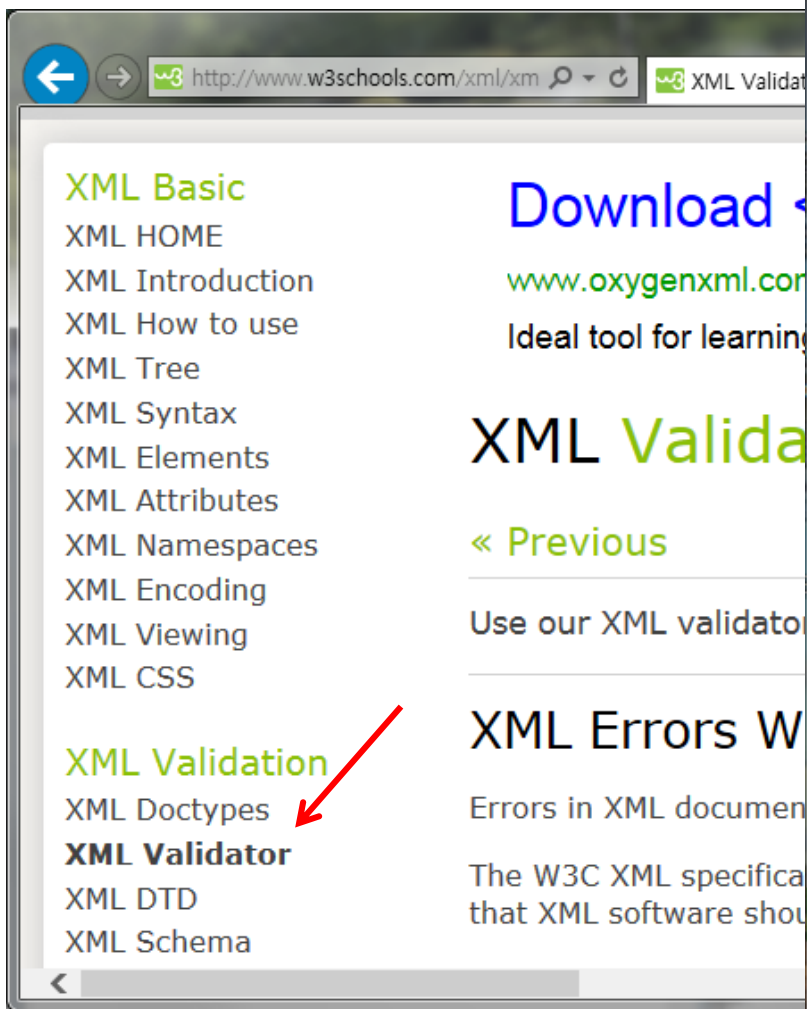
Description	File	Line	Column
-------------	------	------	--------

XSL Example

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type = "text/xsl" href =
"Catalog.xsl"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  <cd>
    <title>Hide your heart</title>...
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <body>
        <h2>My CD Collection</h2>
        <table border="1">
          <tr bgcolor="#9acd32">
            <th style="text-align:left">Title</th>
            <th style="text-align:left">Artist</th>
          </tr>
          <xsl:for-each select="catalog/cd">
            <tr>
              <td><xsl:value-of select="title"/></td>
              <td><xsl:value-of select="artist"/></td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

http://www.w3schools.com/xml/xml_validator.asp



[Home](#) [HTML](#) [CSS](#) [JAVASCRIPT](#) [SQL](#) [PYTHON](#) [MORE ▾](#)

- XML Introduction
- XML How to use
- XML Tree
- XML Syntax
- XML Elements
- XML Attributes
- XML Namespaces
- XML Display
- XML HttpRequest
- XML Parser
- XML DOM
- XML XPath
- XML XSLT
- XML XQuery
- XML XLink
- XML Validator**
- XML DTD
- XML Schema
- XML Server
- XML Examples
- XML Quiz
- XML Certificate

To help you syntax-check your XML, we have created an

Try to syntax-check correct XML :

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Try to syntax-check incorrect XML :

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</pheading>
<body>Don't forget me this weekend!</body>
</note>
```

Checking Syntax

xmlvalidator.new-studio.org/

XML Validator Online BETA

XML Validation

☒ XPath ☐ Schema ☐ DTD

Charset: UTF-8 (Unicode, worldwide) ▼

Choose XML file to validate: 파일 선택 memo1.xml

Check

XPath Explorer

Run

Validation Result



Validation successful!

Source Code

[view plain](#) [print](#) ?

```
1. <?xml version="1.0" encoding="euc-kr" standalone="yes"?>
2. <?xml-stylesheet type="text/css" href="memo1.css"?>
```

XML Validation

☐ XPath ☐ Schema ☒ DTD

Charset:

Choose XML file to validate: hello_w_dtd3.xml

Due to the technical issue, embedded DTD is allowed ONLY.

Check

XPath Explorer

Run

Validation Result



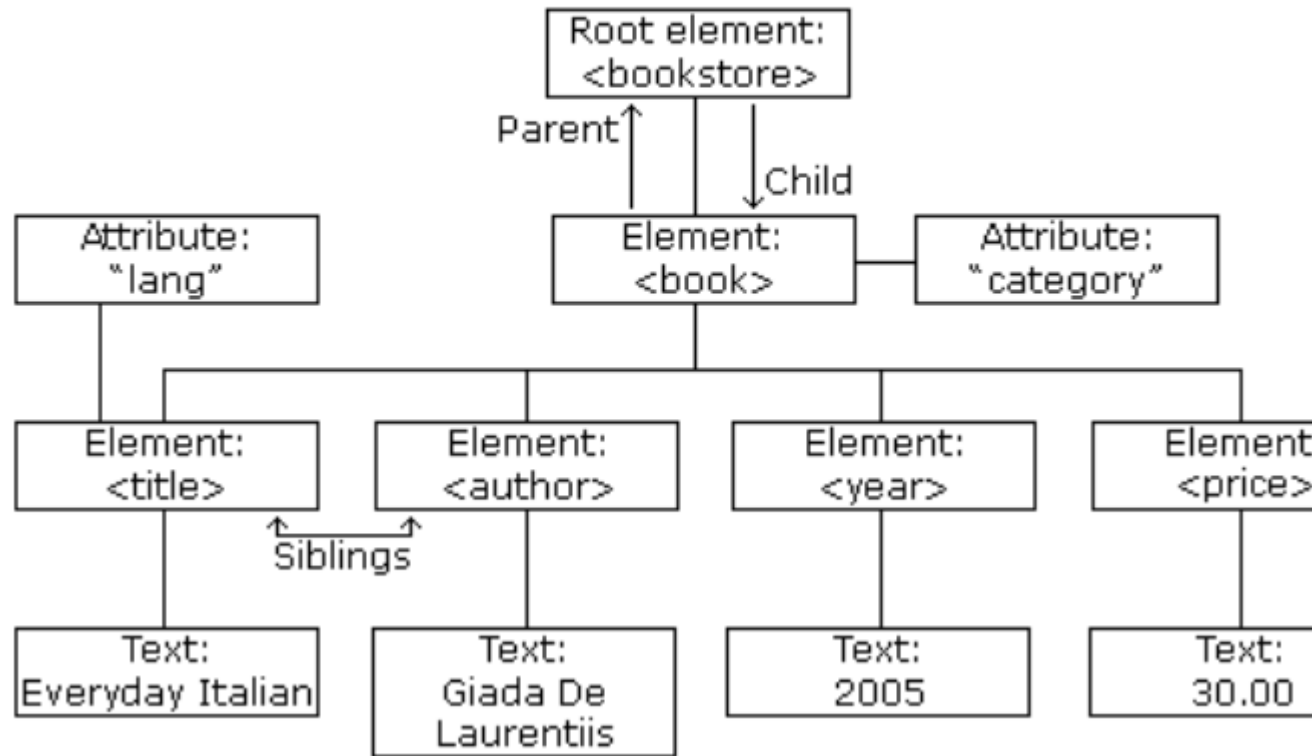
Validation successful!

Source Code

view plain print ?


```
1. <?xml version = "1.0"?>
2.
3. <!-- intro.xml -->
4.
5. <!DOCTYPE myMessage [
6. <!ELEMENT myMessage ( message , hello , blessing )>
7. <!ELEMENT message ( #PCDATA )>
8. <!ELEMENT hello ( #PCDATA )>
9. <!ELEMENT blessing ( #PCDATA )>
10. ]>
```

XML Tree Structure



```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Code Beautify shows a tree structure...

 **Code Beautify**

JSON Formatter | Hex Color Codes | HMAC Generator | My Ip | Search | Recent Links | More | %EC%8.. | **Save & Share**

XML Viewer

XML Input

sample

```
1 <?xml version="1.0" encoding="euc-kr"
  standalone="yes"?>
2 <?xml-stylesheet type="text/css" href
  ="memo1.css"?>
3 <MEMO>
4   <TO> To: Eugene</TO>
5   <FROM> From: Hong, Gil-Dong</FROM>
6   <CONTENTS> Can we dine out at 1 PM
    ?</CONTENTS>
7 </MEMO>
8
9
10
```

Load Url

Browse

Tree View

Beautify / Form

Minify

Result : XML Tree View

memo ..

to To: Eugene

from From: Hong, Gil-Dong

contents Can we dine out at 1 PM?

Syntax: XML Naming Rules

- Element names are **case-sensitive**
- Element names must start with a **letter** or **underscore**
- Element names cannot start with the letters xml (or XML, or Xml, etc) – **not anymore**
- Element names can contain letters, digits, **hyphens, underscores, and periods**
- Element names cannot contain spaces

JSON Formatter | Hex Color Codes | HTML Generator | my ip | Search | Recent Links | mo

XML Viewer☆

XML Input

 [sample](#)  

```
1 <?xml version="1.0" encoding="euc-kr"
   standa one="yes"?>
2 <?xml-stylesheet type="text/css" href
   = "emo1.css"?>
3 <xml>
4   <TO> To: Eugene</TO>
5   <FROM> From: Hong, Gil-Dong</FROM>
6   <CONTENTS> Can we dine out at 1 PM
   ?</CONTENTS>
7 </xml>
8
```

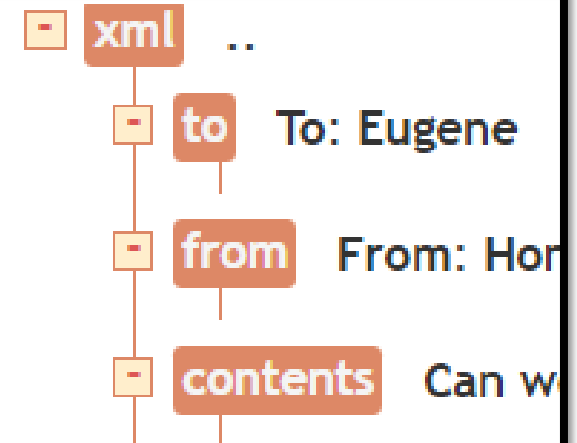
Load Url

Browse

Tree View

Beautify / Form

Result : XML Tree



Syntax: Best Naming Practices

- Create **descriptive** names, like this: <person>, <firstname>, <lastname>.
- Create **short and simple** names, like this: <book_title> not like this: <the_title_of_the_book>.
- **Avoid "-"**. If you name something "first-name", some software may think you want to **subtract** "name" from "first".
- **Avoid "."**. If you name something "first.name", some software may think that "name" is a property of the object "first".

Naming Styles

no naming styles defined for XML elements

Style	Example	Description
Lower case	<code><firstname></code>	All letters lower case
Upper case	<code><FIRSTNAME></code>	All letters upper case
Underscore	<code><first_name></code>	Underscore separates words
Pascal case	<code><FirstName></code>	Uppercase first letter in each word
Camel case	<code><firstName></code>	Uppercase first letter in each word except the first
Camel case is a common naming rule in JavaScripts.		

Element Attributes

Attributes are designed to contain data related to a specific element.

Attribute values must always be quoted. Either single or double quotes can be used.

<person gender="female">

<person gender='female'>

<gangster name='George "Shotgun" Ziegler'>

**<gangster name="George " Shotgun"
Ziegler"**

XML Elements vs. Attributes

```
<person gender="female">  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```

```
<person>  
  <gender>female</gender>  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```

no rules about when to use attributes or
when to use elements in XML

five predefined XML entity references (w. special meaning)

- < standing for <
- > standing for >
- & standing for &
- ' standing for ‘ apostrophe
- " standing for” quotation mark

XML Viewer

XML Input

[sample](#)  

```
1 <?xml version="1.0" encoding="euc-kr" standalone
  ="yes"?>
2 <?xml-stylesheet type="text/css" href="memo1.css"?>
3 <xml gender = 'a &apos;' temp='b &quot;|'>
4   <TO> To: Eugene</TO>
5   <FROM> From: Hong, Gil-Dong</FROM>
6   <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
7 </xml>
```

Load Url

Browse

Tree View

Beautify / Format

Result : XML Tree View

```
- xml ..
  @gender: a '
  @temp: b "
  - to To: Eugene
  - from From: Hong, Gil-Dong
  - contents Can we dine out at 1
```

XML Viewer☆

Save

XML Input

 sample  

```
1 <?xml version="1.0" encoding="euc-kr" standalone
  ="yes"?>
2 <?xml-stylesheet type="text/css" href="memo1.css"?>
3 <MEMO date = '"Sep. 8 2020"'>
4   <TO> To: Eugene</TO>
5   <FROM> From: Hong, Gil-Dong</FROM>
6   <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
7 </MEMO>
8
```

Load Url

Browse

Tree View

Beautify / Format

Minify

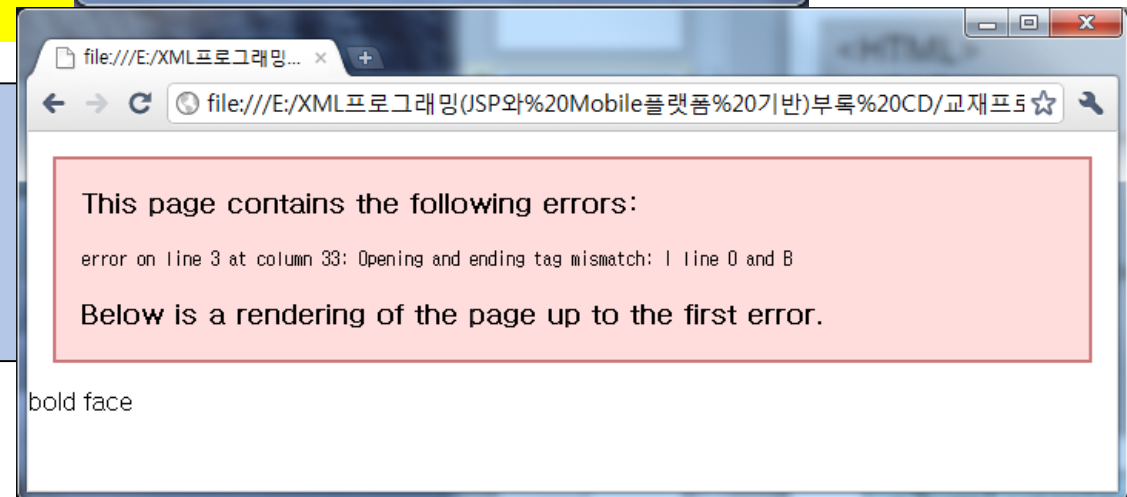
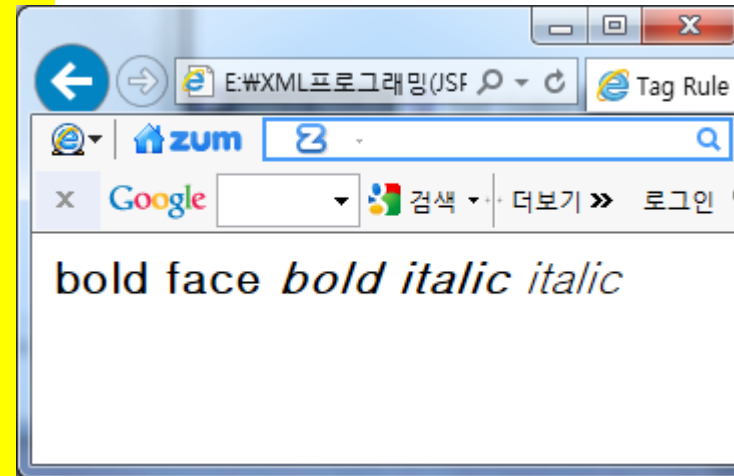
Result : XML Tree View

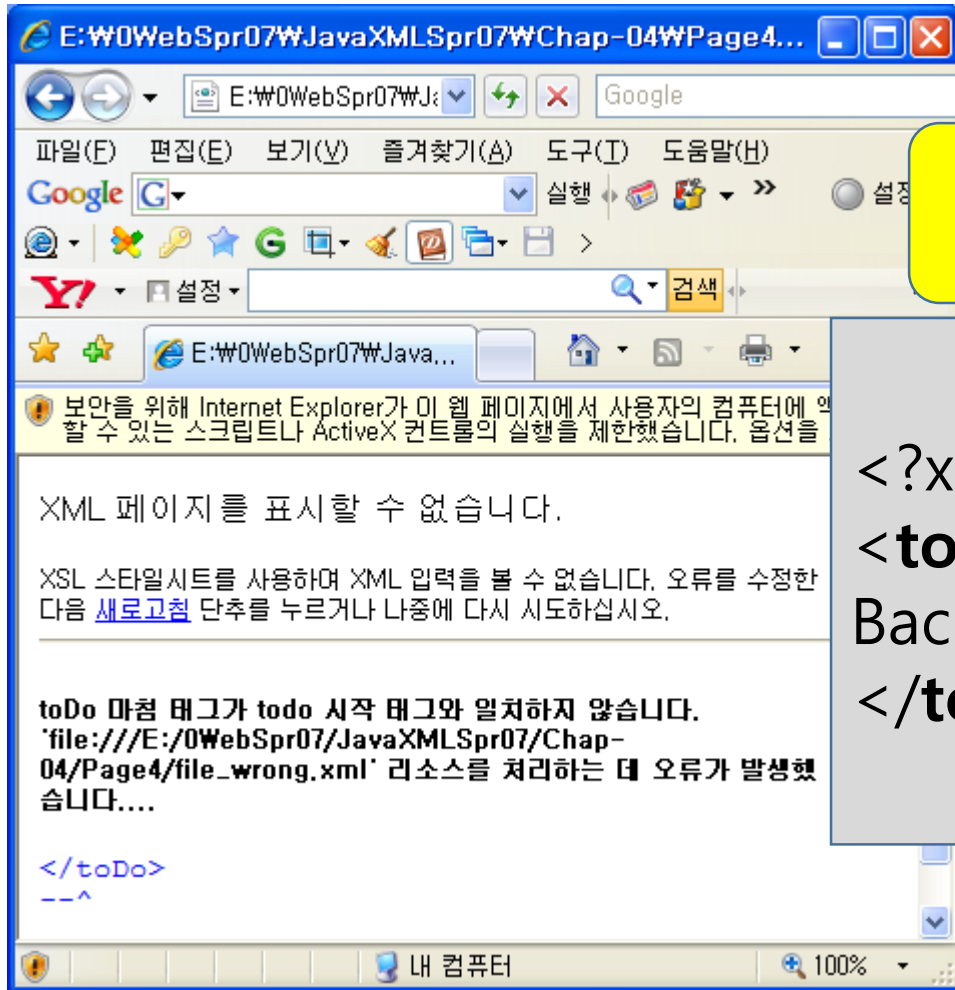


tag overlapping

```
<HTML>
<HEAD>
<TITLE>Tag Rule</TITLE>
</HEAD>
<BODY>
<B>bold face <I>bold italic </B>
italic</I>
</BODY>
</HTML>
```

```
<?xml version="1.0" encoding="euc-kr" ?>
<tag>
<B>bold face <I>bold italic </B>italic </I>
</tag>
```





Well-Formedness of XML Syntax

```
<?xml version="1.0" standalone="yes"?>  
<todo>  
Backup sales data for last month  
</toDo>
```

Case-Sensitive Element Name

Try to use legal XML element names and test it on your browser.

HTML Viewer☆

HTML Input

 sample  

```
1 <HTML>
2 <HEAD>
3 <TITLE>Tag Rule</TITLE>
4 </HEAD>
5 <BODY>
6 <B>bold face <I>bold italic </B> italic</I>
7 </BODY>
8 </HTML>
```

Load Url

Browse

Default Tab Space ▼

Beautify

Minify

Run

Result : HTML Output

bold face ***bold italic** italic*

XML Viewer☆

Save & Share

XML Input

 [sample](#)  

```
1 <?xml version="1.0" encoding="euc-kr" ?>
2 <tag>
3 <B>bold face <I>bold italic </B>italic </I>
4 </tag>
5
```

Load Url

Browse

Tree View

Beautify / Format

Minify

Ad closed by Google

Result : XML Tree View

