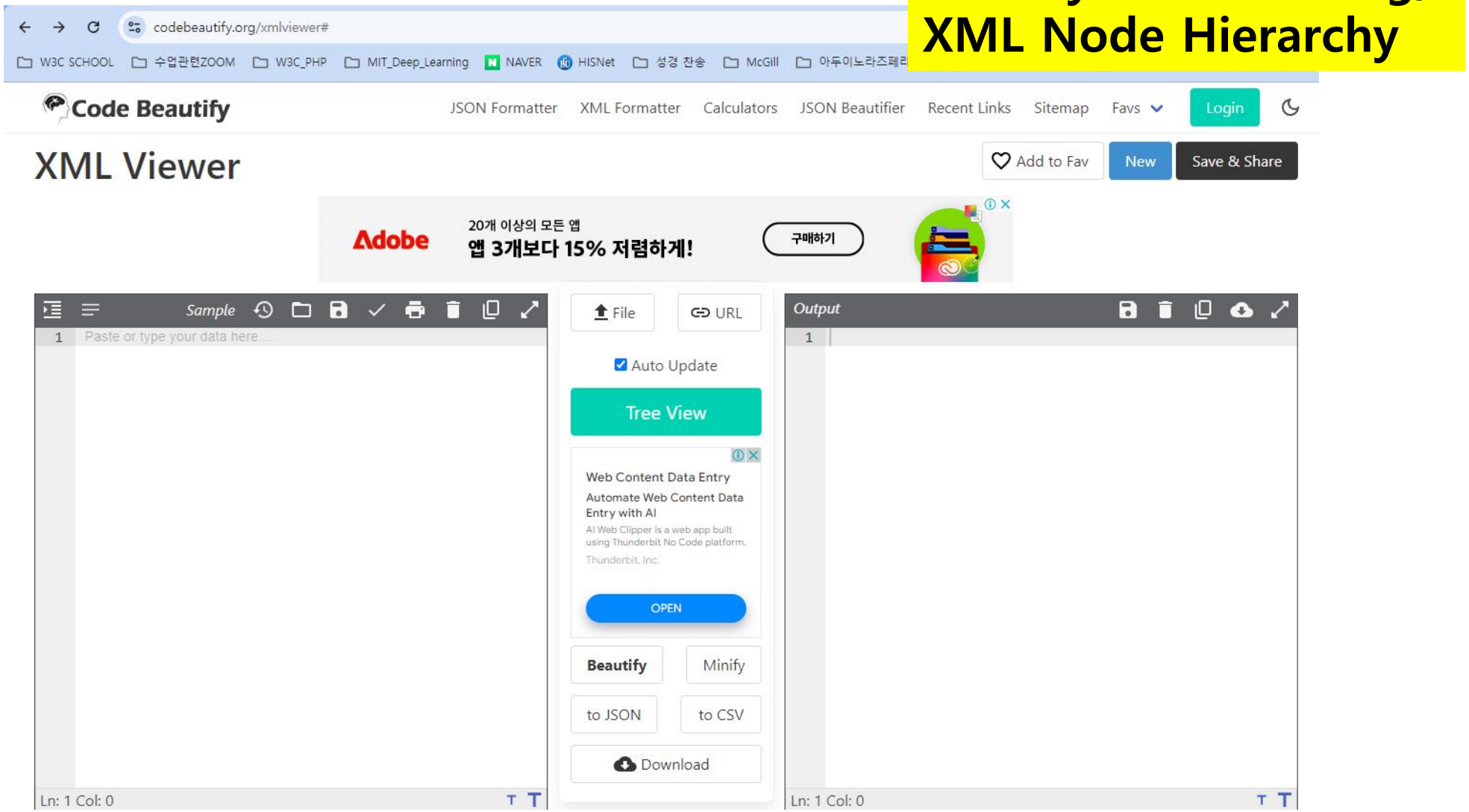
XML Syntax and Editors/Validator

https://www.w3schools.com/xml/

Keywords: Syntax, Editor – XMLViewer, XML Validator

https://codebeautify.org/xmlviewer#

XML Syntax Checking;
XML Node Hierarchy



https://codebeautify.org/xmlviewer#

Sample

```
1  <?xml version="1.0" encoding="euc-kr" standalone="yes"
       ?>
2  <?xml-stylesheet type="text/css" href
       ="memo1_Simplified2.css"?>
3  <MEMO>
4     <TO> To: &quot; Eugene &quot;</TO>
5     <FROM> From: Hong, Gil-Dong</FROM>
6     <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
7  </MEMO>
8
9
10
```

File | URL

Auto Update

Tree View

Beautify | Minify

to JSON | to CSV

Download

Output

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

Output

```
1  {
2    "MEMO": {
3      "TO": "To: &quot; Eugene &quot;",
4      "FROM": "From: Hong, Gil-Dong",
5      "CONTENTS": "Can we dine out at 1 PM?"
6    }
7  }
```

Output

```
1  MEMO/TO,MEMO/FROM,MEMO/CONTENTS
2  "To: "" Eugene """,From: Hong, Gil-Dong",Can we
       out at 1 PM?
3
```

https://codebeautify.org/xmlviewer

Sample

```
1  <?xml version="1.0" encoding="euc-kr" standalone="yes"
       ?>
2  <?xml-stylesheet type="text/css" href
       ="memo1_Simplified2.css"?>
3  <MEMO>
4     <TO> To: &quot; Eugene &quot;</TO>
5     <FROM> From: Hong, Gil-Dong</FROM>
6     <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
7  </MEMO>
8
9
10
```

File | URL

Auto Update

Tree View

Beautify | Minify

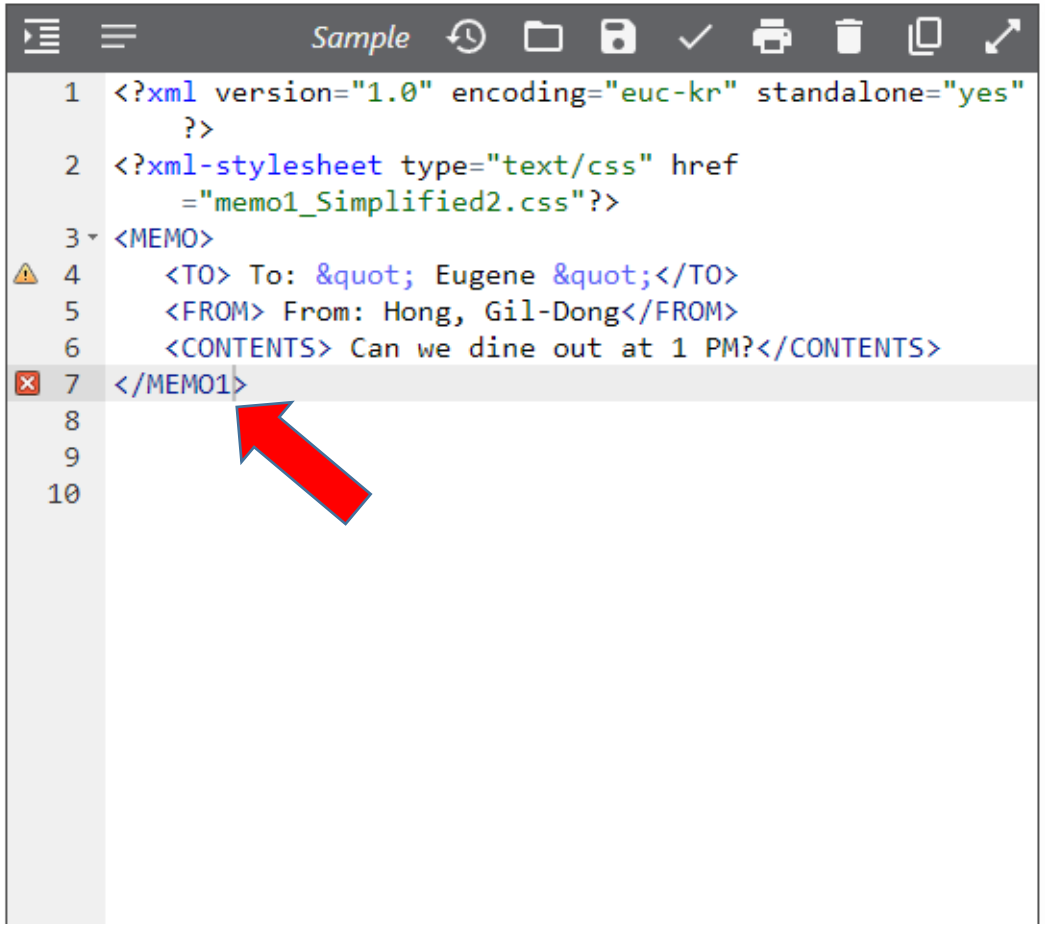
Output

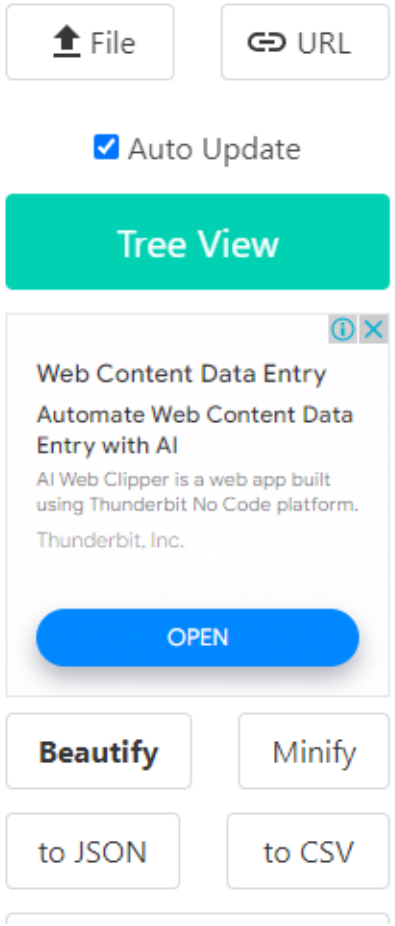
```
1  <?xml version="1.0" encoding="euc-kr" standalone="yes
       ?>
2  <?xml-stylesheet type="text/css" href
       ="memo1_Simplified2.css"?>
3  <MEMO>
4     <TO> To: &quot; Eugene &quot;</TO>
5     <FROM> From: Hong, Gil-Dong</FROM>
6     <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
7  </MEMO>
8
```

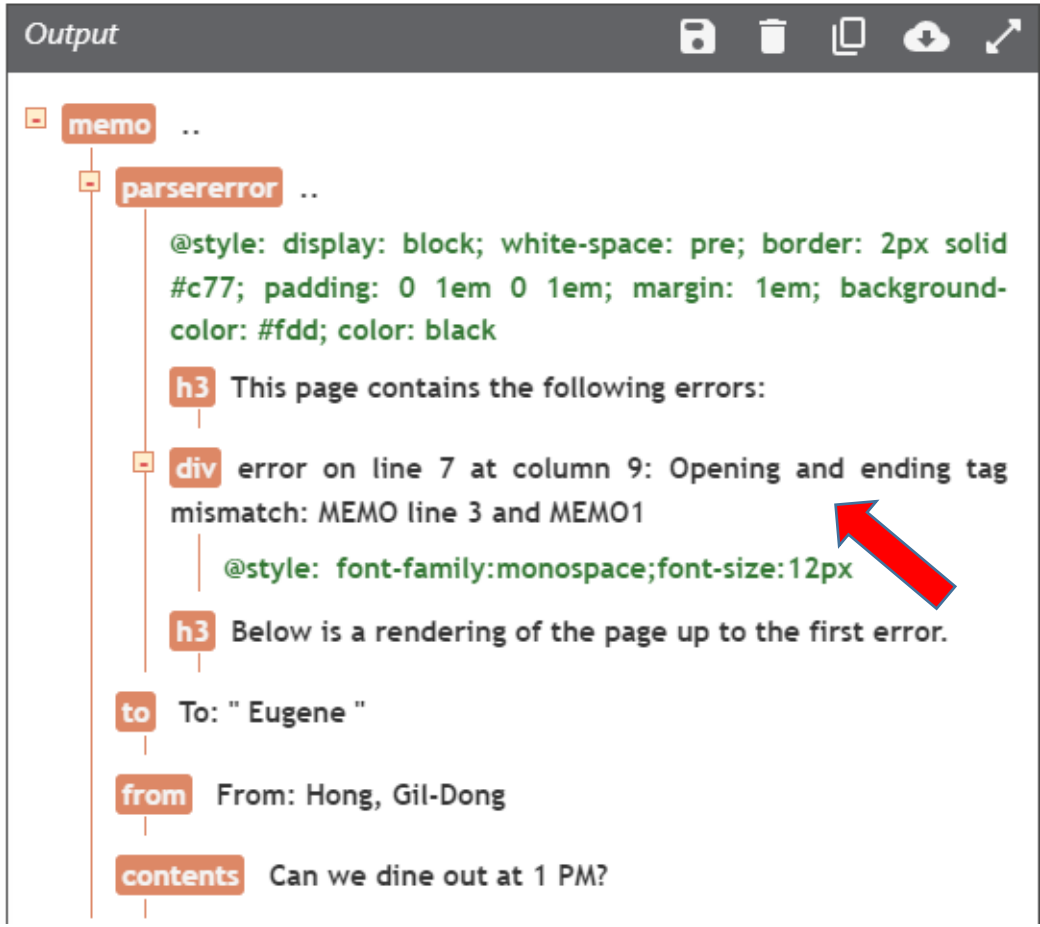
Output

```
1  <?xml version="1.0" encoding="euc-kr" standalone="yes"
       ?><?xml-stylesheet type="text/css" href
       ="memo1_Simplified2.css"?><MEMO><TO> To: &quot;
       Eugene &quot;</TO><FROM> From: Hong, Gil-Dong</FROM
       ><CONTENTS> Can we dine out at 1 PM?</CONTENTS
       ></MEMO>
```

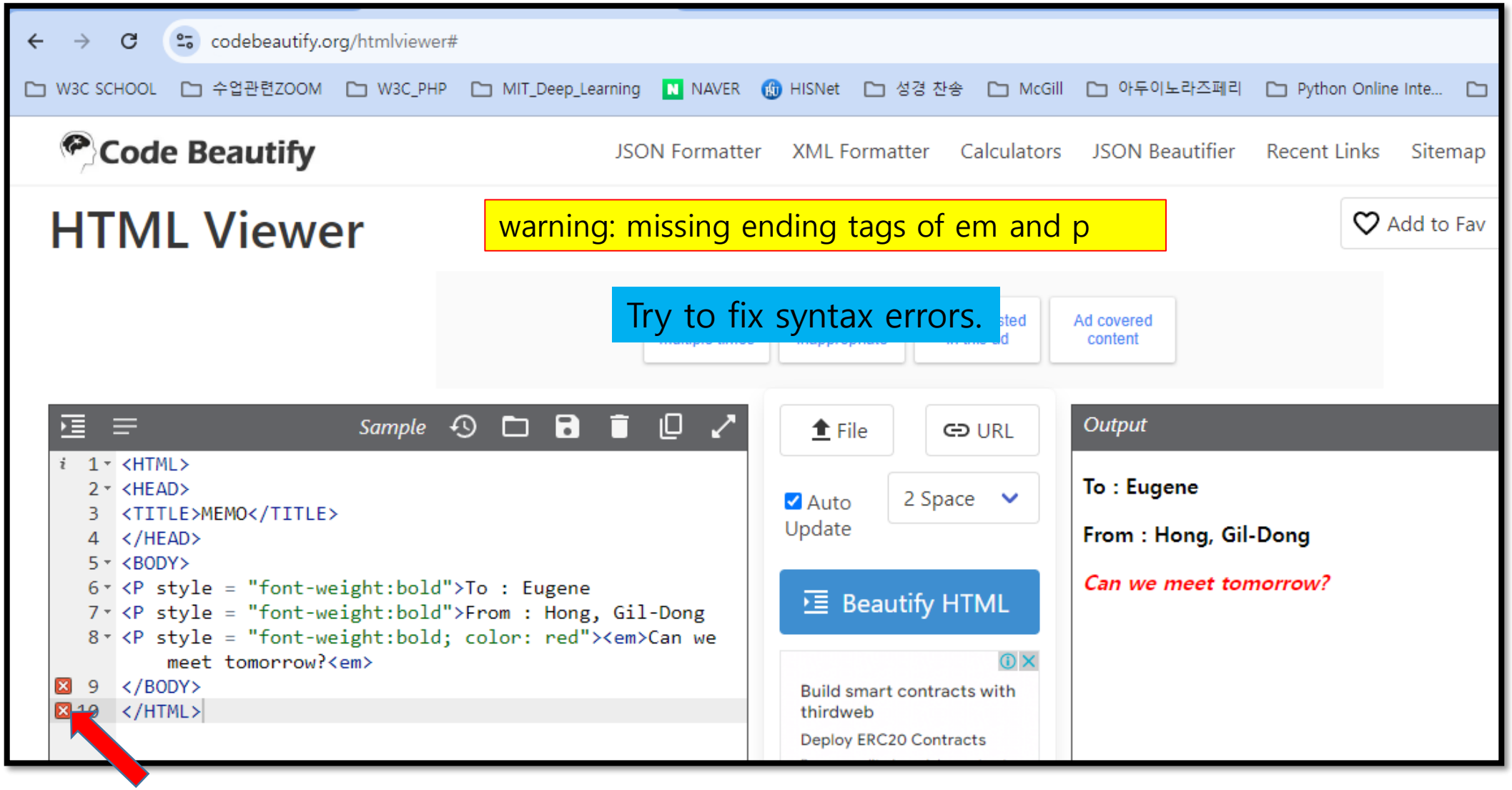
https://codebeautify.org/xmlviewer

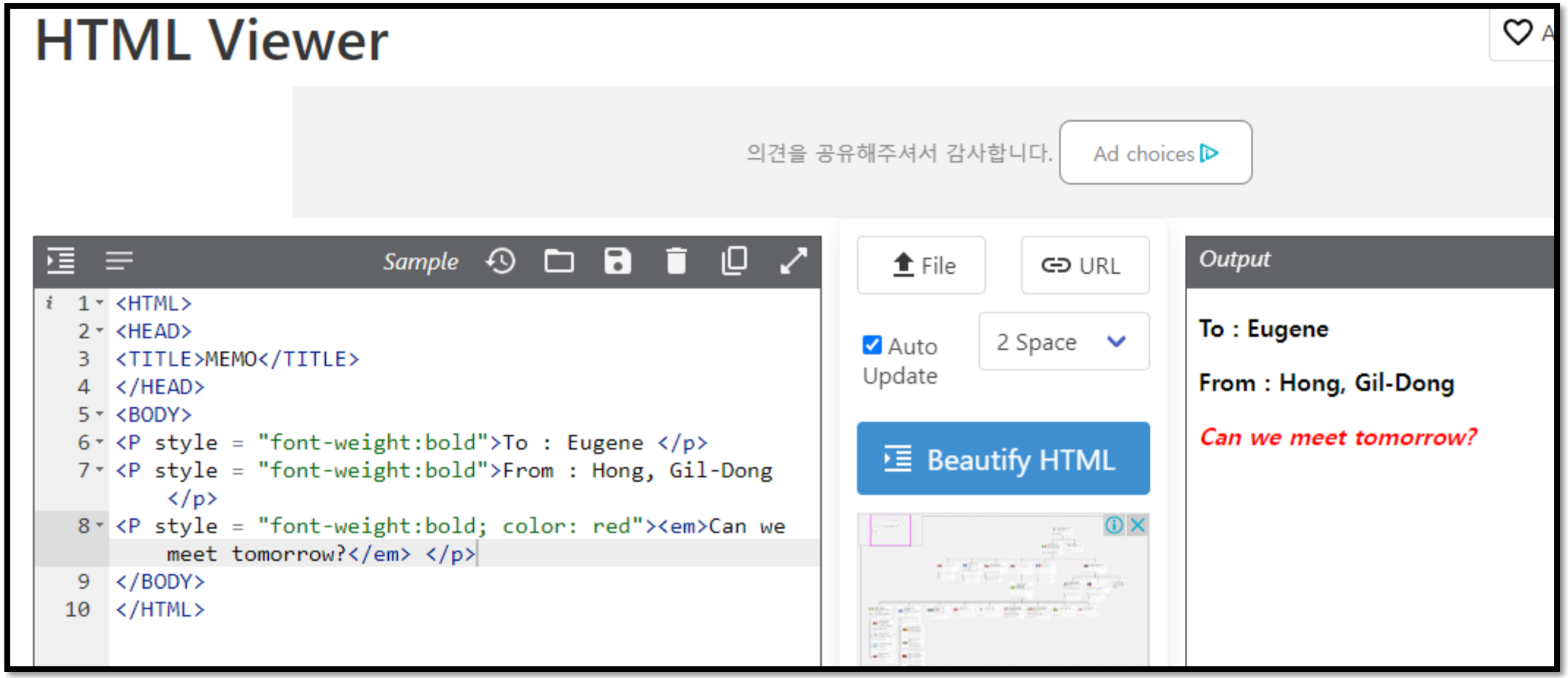




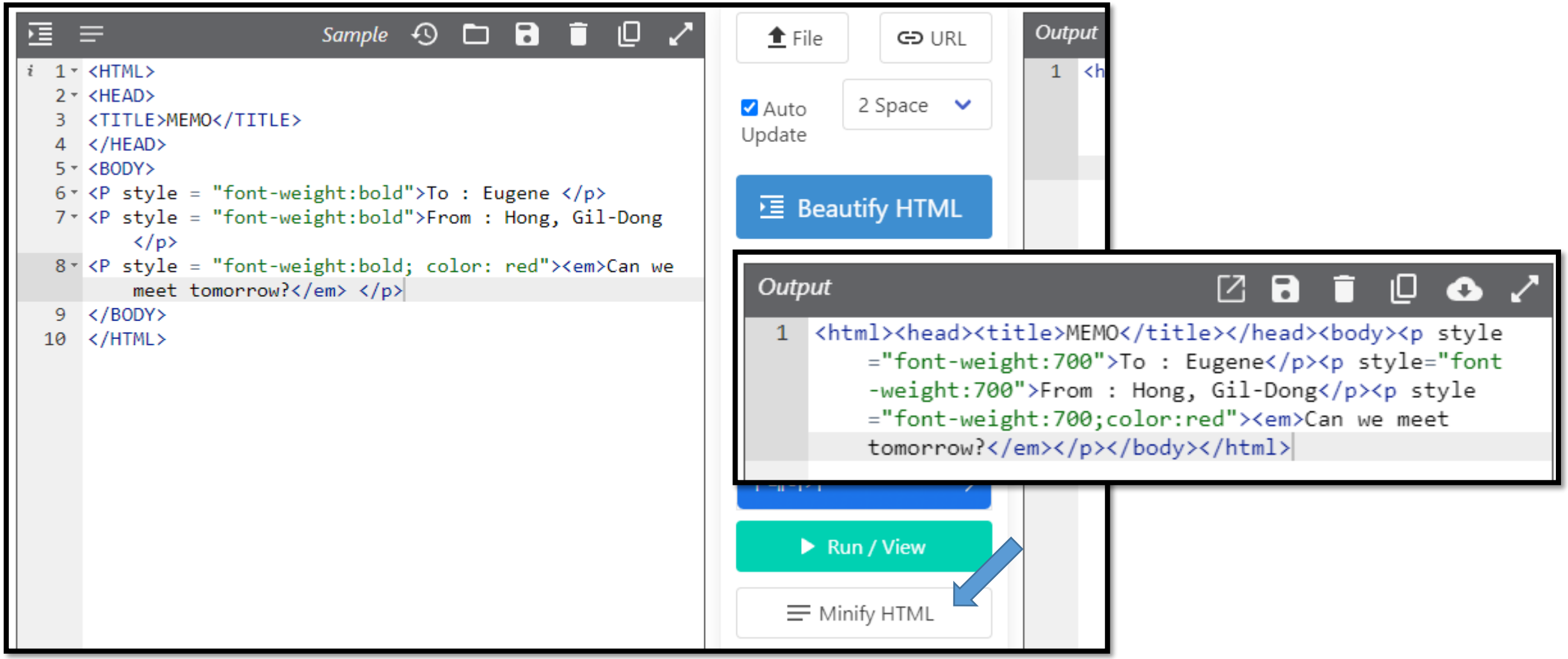


HTML Viewer
https://codebeautify.org/htmlviewer




HTML Viewer
의견을 공유해주셔서 감사합니다.
Ad choices
Sample
1 <HTML>
2 <HEAD>
3 <TITLE>MEMO</TITLE>
4 </HEAD>
5 <BODY>
6 <P style = "font-weight:bold">To : Eugene </p>
7 <P style = "font-weight:bold">From : Hong, Gil-Dong
</p>
8 <P style = "font-weight:bold; color: red">Can we
meet tomorrow? </p>
9 </BODY>
10 </HTML>
File
URL
Auto Update
2 Space
Beautify HTML
Output
To : Eugene
From : Hong, Gil-Dong
Can we meet tomorrow?

https://codebeautify.org/htmlviewer



HTML Viewer

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
       meet tomorrow?
9  </BODY>
10 </HTML>
11
```

File | URL

Auto Update | 2 Space

Beautify HTML

Run / View

Minify HTML

Output

To : Eugene

From : Hong, Gil-Dong

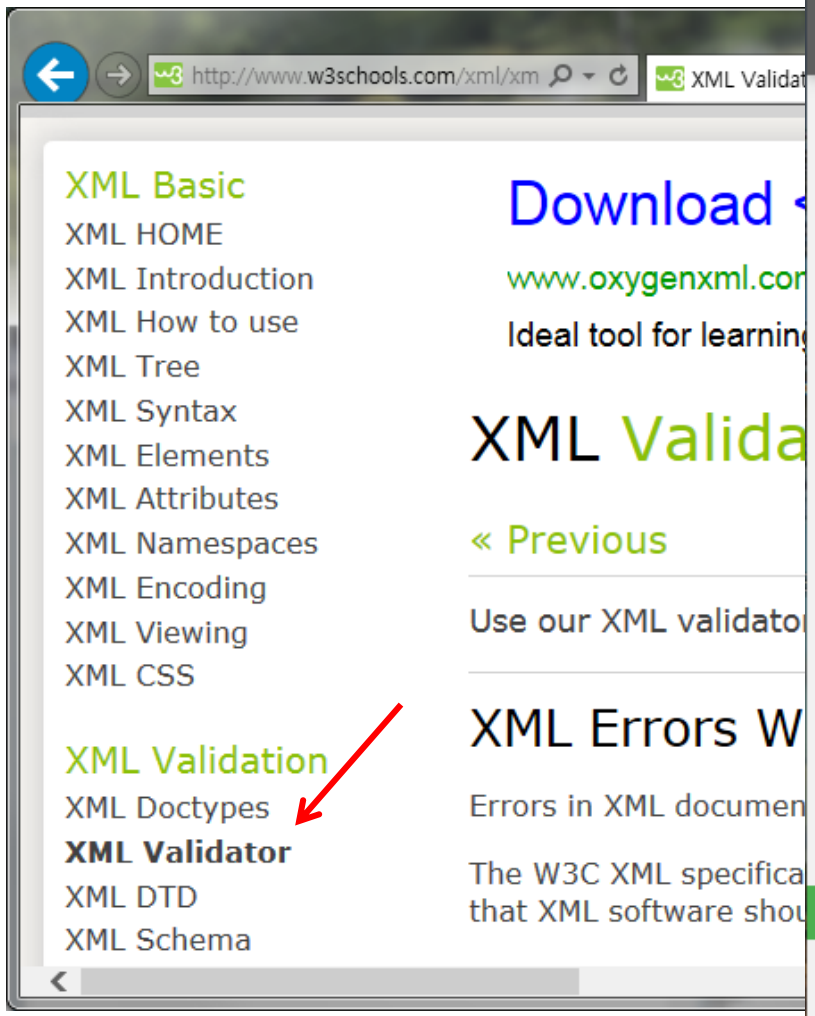
Can we meet tomorrow?

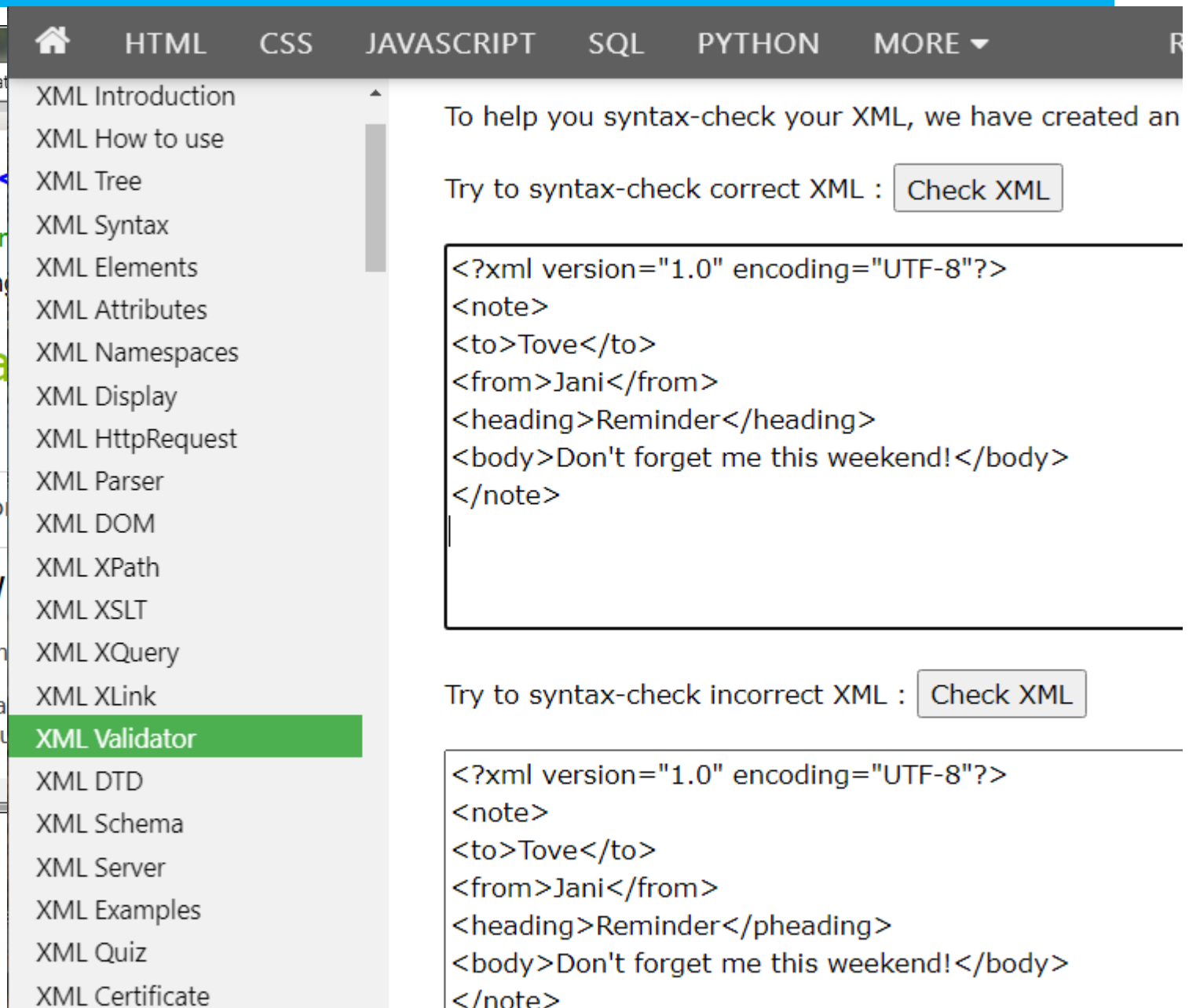
Sample

```
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
       meet tomorrow?</em></p>
9  </BODY>
10 </HTML>
11
```

warning: missing ending tags of em and p

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

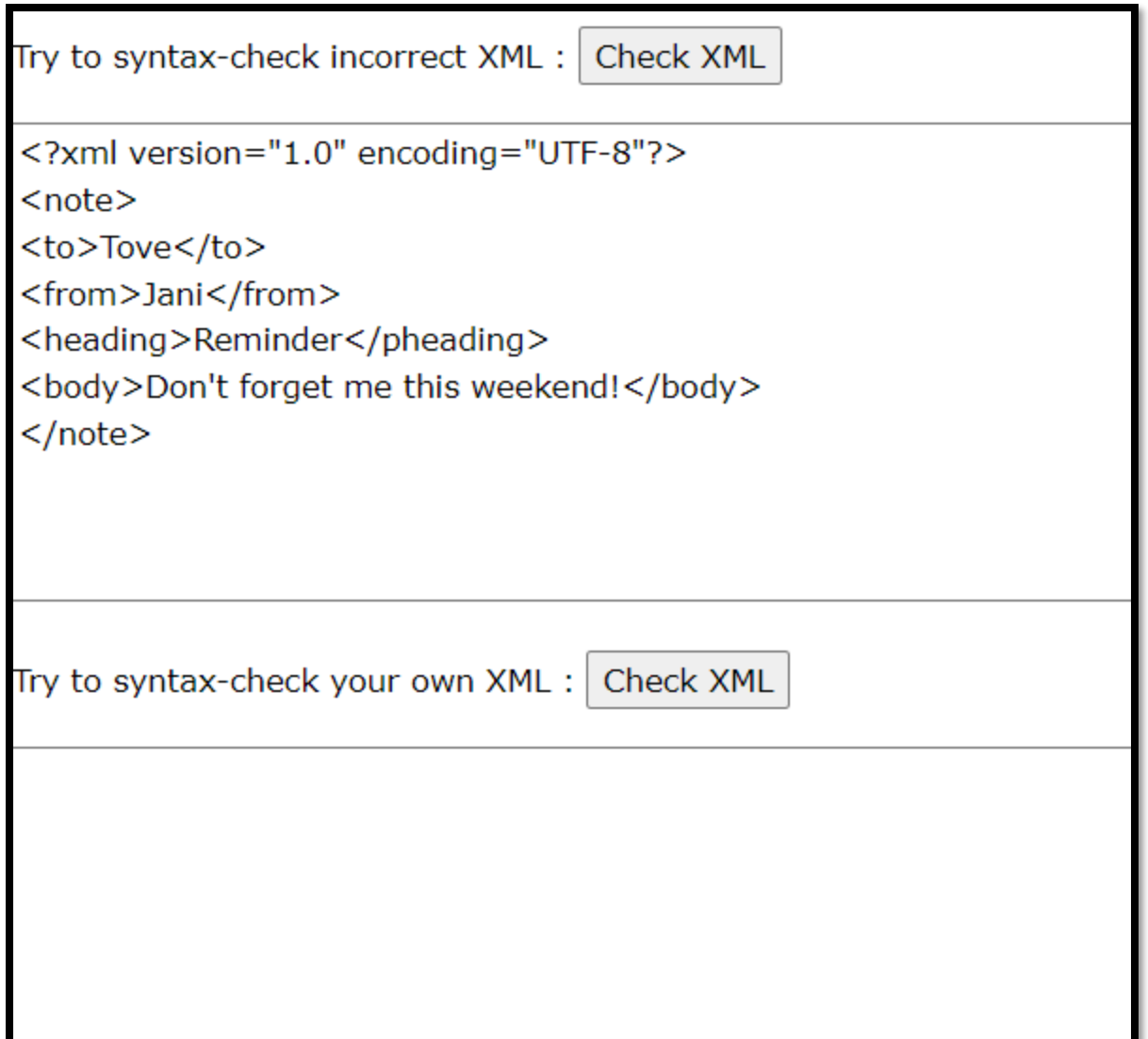




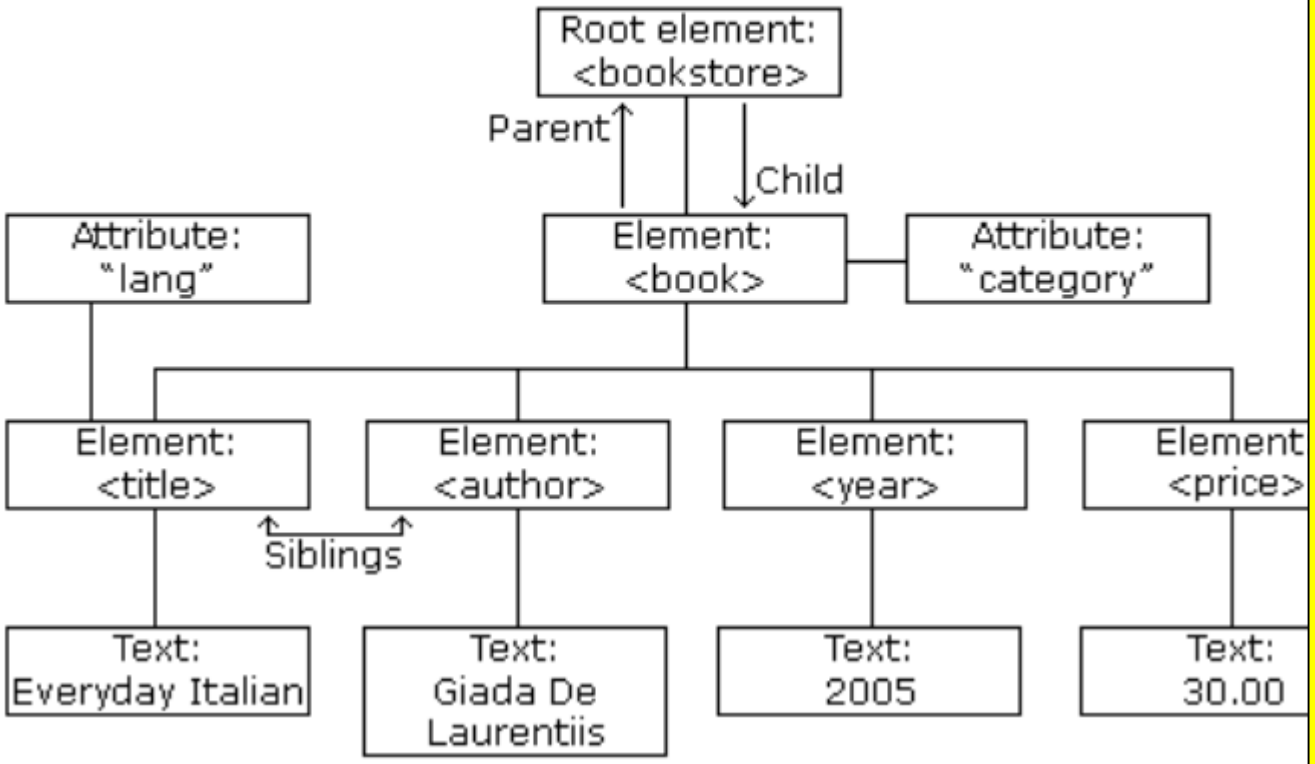
Checking Syntax

Try using XML with W3Schools Validator.

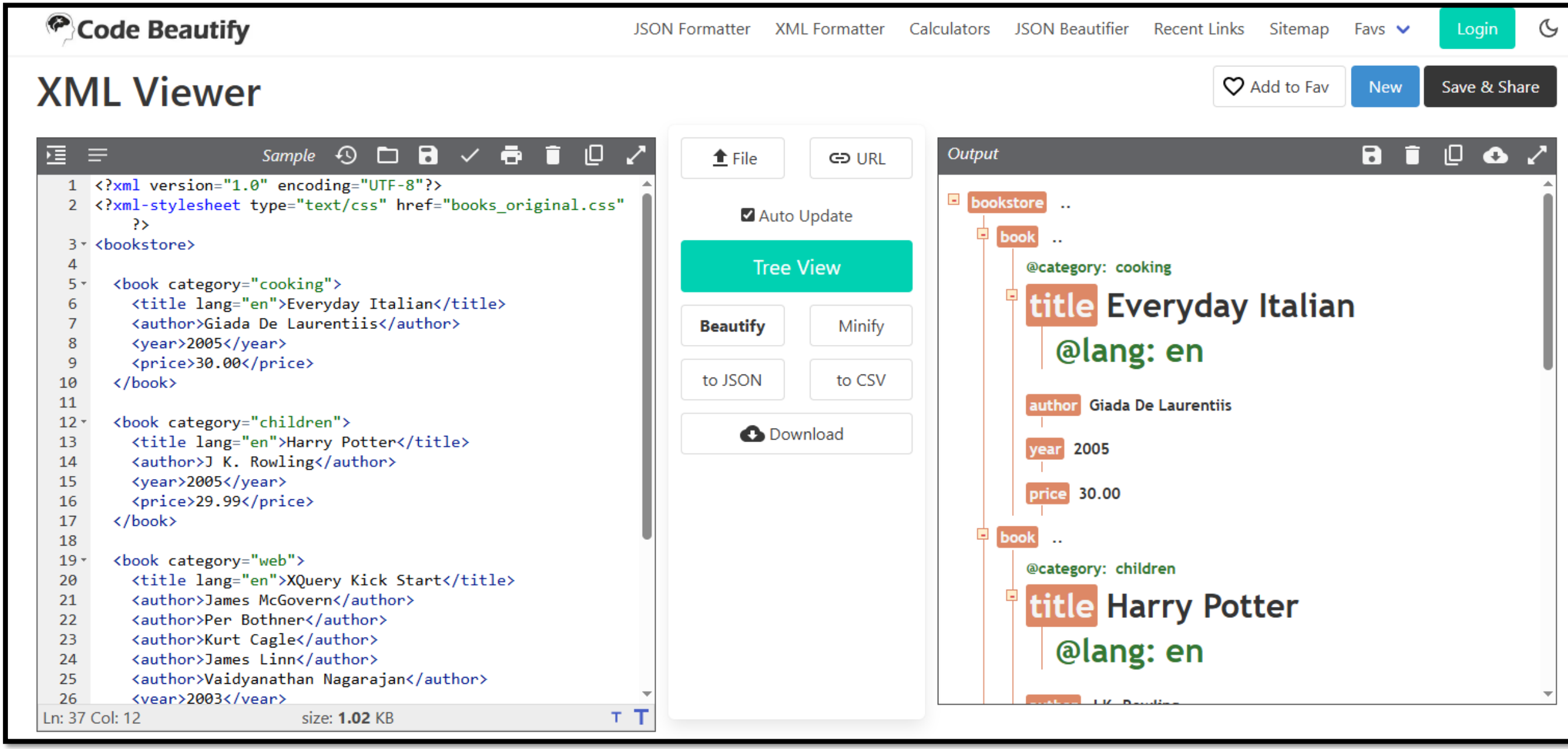
https://www.w3schools.com/xml/xml_validator.asp



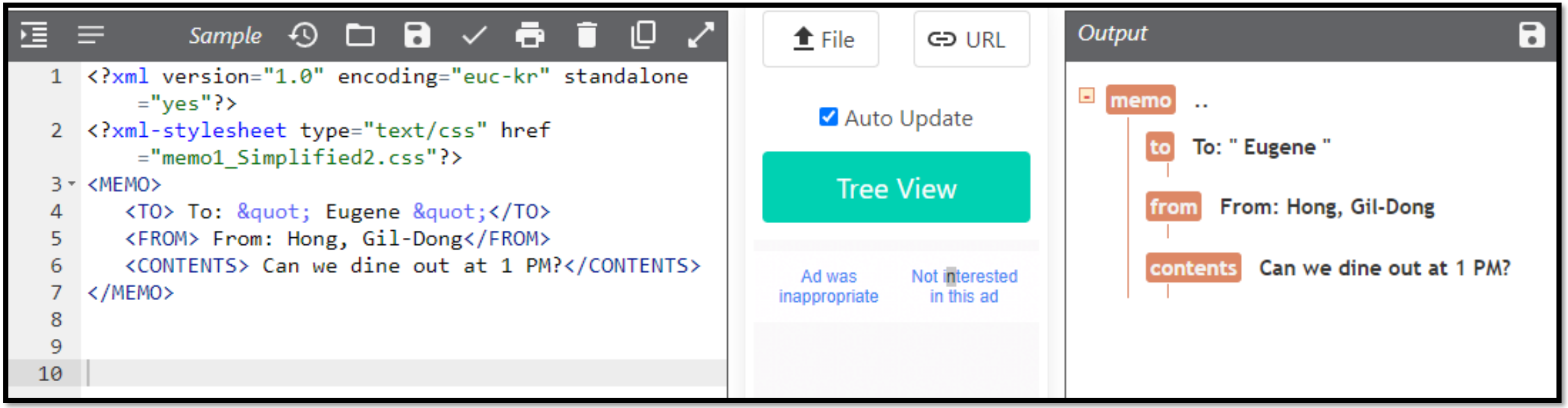
XML Tree Structure




```
<?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
JSON Formatter
XML Formatter
Calculators
JSON Beautifier
Recent Links
Sitemap
Favs
Login
XML Viewer
Add to Fav
New
Save & Share
Sample
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/css" href="books_original.css" ?>
3 <bookstore>
4
5 <book category="cooking">
6 <title lang="en">Everyday Italian</title>
7 <author>Giada De Laurentiis</author>
8 <year>2005</year>
9 <price>30.00</price>
10 </book>
11
12 <book category="children">
13 <title lang="en">Harry Potter</title>
14 <author>J K. Rowling</author>
15 <year>2005</year>
16 <price>29.99</price>
17 </book>
18
19 <book category="web">
20 <title lang="en">XQuery Kick Start</title>
21 <author>James McGovern</author>
22 <author>Per Bothner</author>
23 <author>Kurt Cagle</author>
24 <author>James Linn</author>
25 <author>Vaidyanathan Nagarajan</author>
26 <year>2003</year>
Ln: 37 Col: 12
size: 1.02 KB
File
URL
Auto Update
Tree View
Beautify
Minify
to JSON
to CSV
Download
Output
bookstore
book
@category: cooking
title Everyday Italian
@lang: en
author Giada De Laurentiis
year 2005
price 30.00
book
@category: children
title Harry Potter
@lang: en

Code Beautify shows a tree structure…



What is an XML Element?

An XML element is everything from (including) the element's start tag to (including) the element's end tag.

```
<price>29.99</price>
```

An element can contain:

- text
- attributes
- other elements
- or a mix of the above

```
<bookstore>
  <book category="children">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Empty XML Elements

An element with no content is said to be empty.

In XML, you can indicate an empty element like this:

```
<element></element>
```

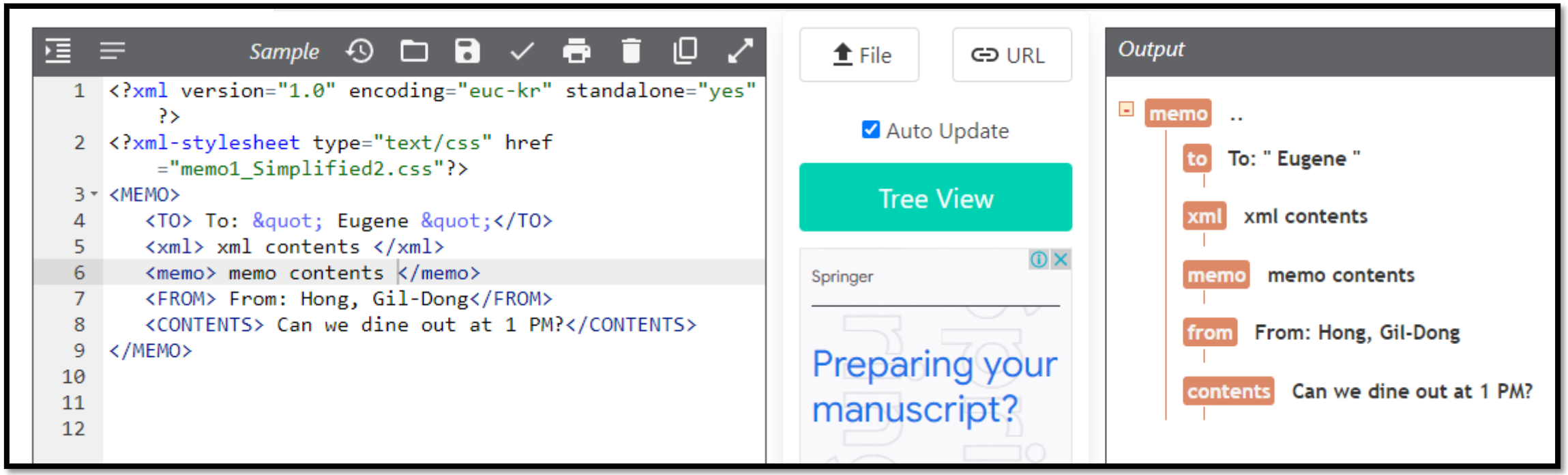
You can also use a so called self-closing tag:

```
<element />
```

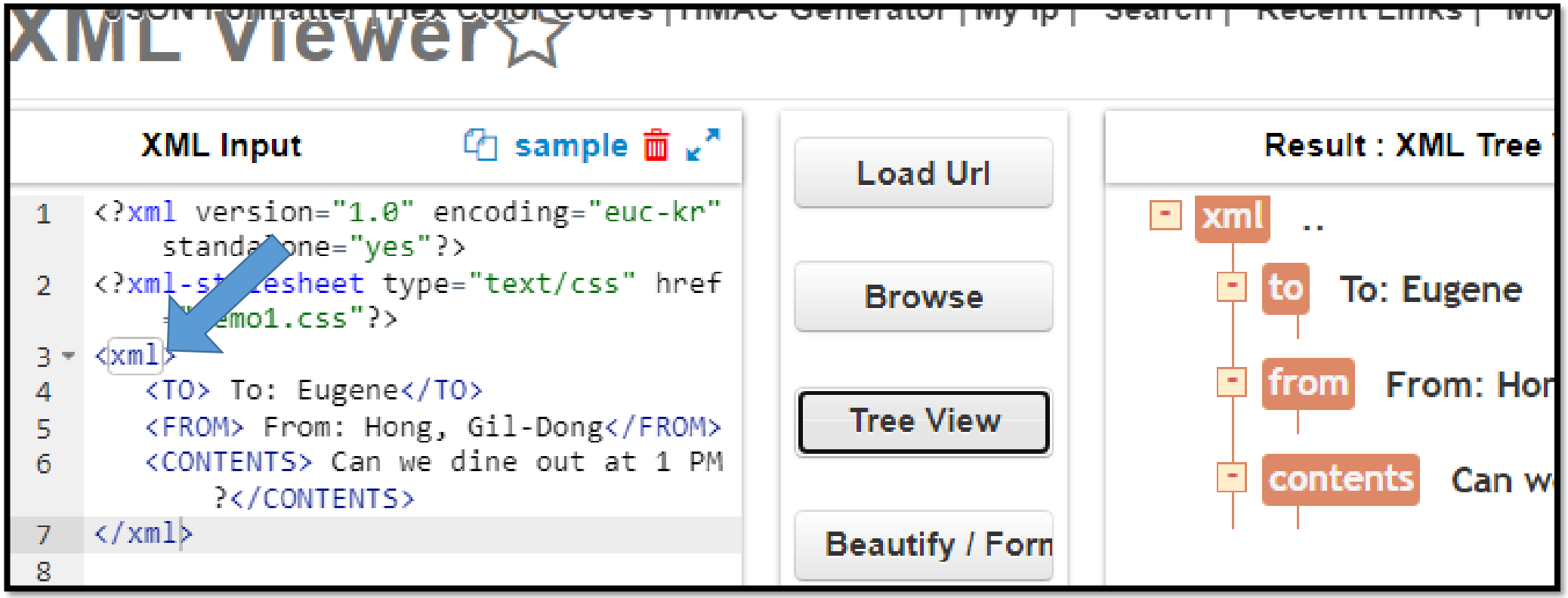
Syntax: XML Naming Rules

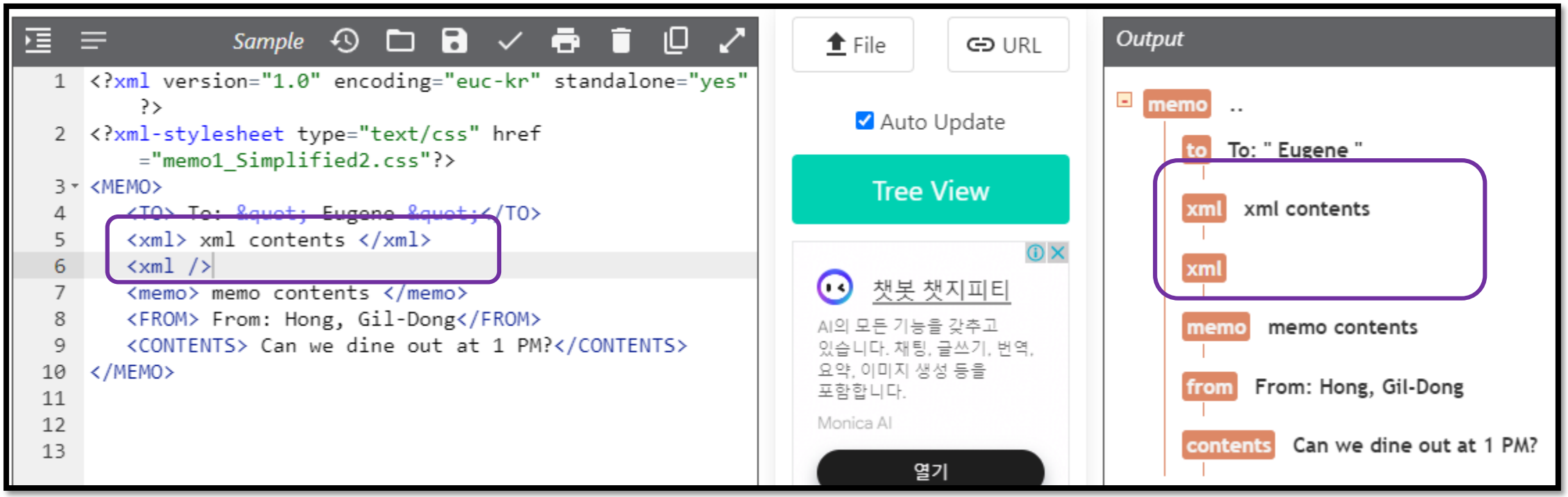
- Element names are **case-sensitive**
- Element names must start with a **letter** or **underscore**
- Element names cannot start with the letters xml (or XML, or Xml, etc) – not anymore, but discouraged
- Element names can contain letters, digits, **hyphens**, **underscores**, and **periods**
- Element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

Exercise XML syntax with “Code Beautify”.



automatically making ending tag…


XML Viewer
XML Input
sample
1 <?xml version="1.0" encoding="euc-kr"
standalone="yes"?>
2 <?xml-stylesheet type="text/css" href
="demo1.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
xml
to
To: Eugene
from
From: Hor
contents
Can w


Sample
1 <?xml version="1.0" encoding="euc-kr" standalone="yes" ?>
2 <?xml-stylesheet type="text/css" href ="memo1_Simplified2.css"?>
3 <MEMO>
4 <TO> To: " Eugene "</TO>
5 <xml> xml contents </xml>
6 <xml />
7 <memo> memo contents </memo>
8 <FROM> From: Hong, Gil-Dong</FROM>
9 <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
10 </MEMO>
11
12
13
File
URL
Auto Update
Tree View
챗봇 챗지피티
AI의 모든 기능을 갖추고 있습니다. 채팅, 글쓰기, 번역, 요약, 이미지 생성 등을 포함합니다.
Monica AI
열기
Output
memo ..
to To: " Eugene "
xml xml contents
xml
memo memo contents
from From: Hong, Gil-Dong
contents Can we dine out at 1 PM?

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	**<firstname>**	**All letters lower case**
Upper case	**<FIRSTNAME>**	**All letters upper case**
Underscore	**<first_name>**	**Underscore separates words**
Pascal case	**<FirstName>**	**Uppercase first letter in each word**
Camel case	**<firstName>**	**Uppercase first letter in each word except the first**

Camel case is a common naming rule in JavaScripts.

XML Tags are Case Sensitive

XML tags are case sensitive. The tag <Letter> is different from the tag <letter>.

Opening and closing tags must be written with the same case:

```
<message>This is correct</message>
```

XML Attribute Values Must Always be Quoted

XML elements can have attributes in name/value pairs just like in HTML.

In XML, the attribute values must always be quoted:

```
<note date="12/11/2007">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

XML Elements Must be Properly Nested

In HTML, you might see improperly nested elements:

```
<b><i>This text is bold and italic</b></i>
```

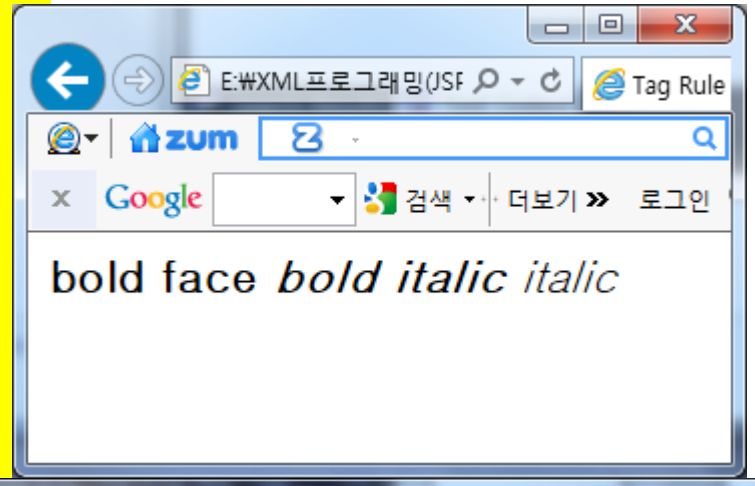
In XML, all elements **must** be properly nested within each other:

```
<b><i>This text is bold and italic</i></b>
```

tag overlapping

Type the source for HTML and XML, then apply them to the browser.

```
<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>
```