XML DTD

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

DTD

- A DTD is a **Document Type Definition**.
- A DTD defines the structure and the legal elements and attributes of an XML document
- With a DTD, independent groups of people can agree on a standard DTD for interchanging data.
- An application can use a DTD to verify that XML data is valid.

Internal DTD

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note ( to, from, heading, body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

An External DTD Declaration

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me
      this weekend!</body>
</note>
```

```
<!ELEMENT note
(to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

The Building Blocks of XML Documents

- Elements
- Attributes
- Entities
- PCDATA
- CDATA

Attributes

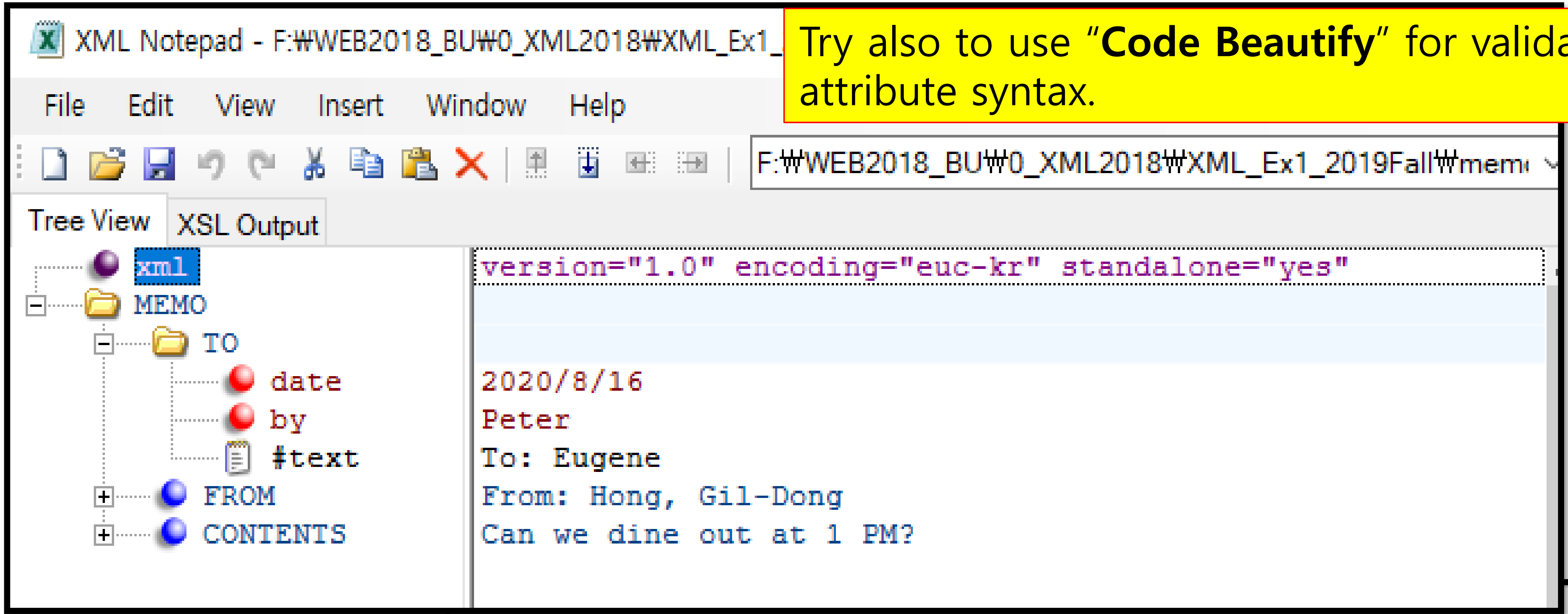
Attributes provide **extra information about elements**.

Attributes are always placed **inside the opening tag** of an element. Attributes always come in **name/value pairs**. The following "img" element has additional information about a source file:

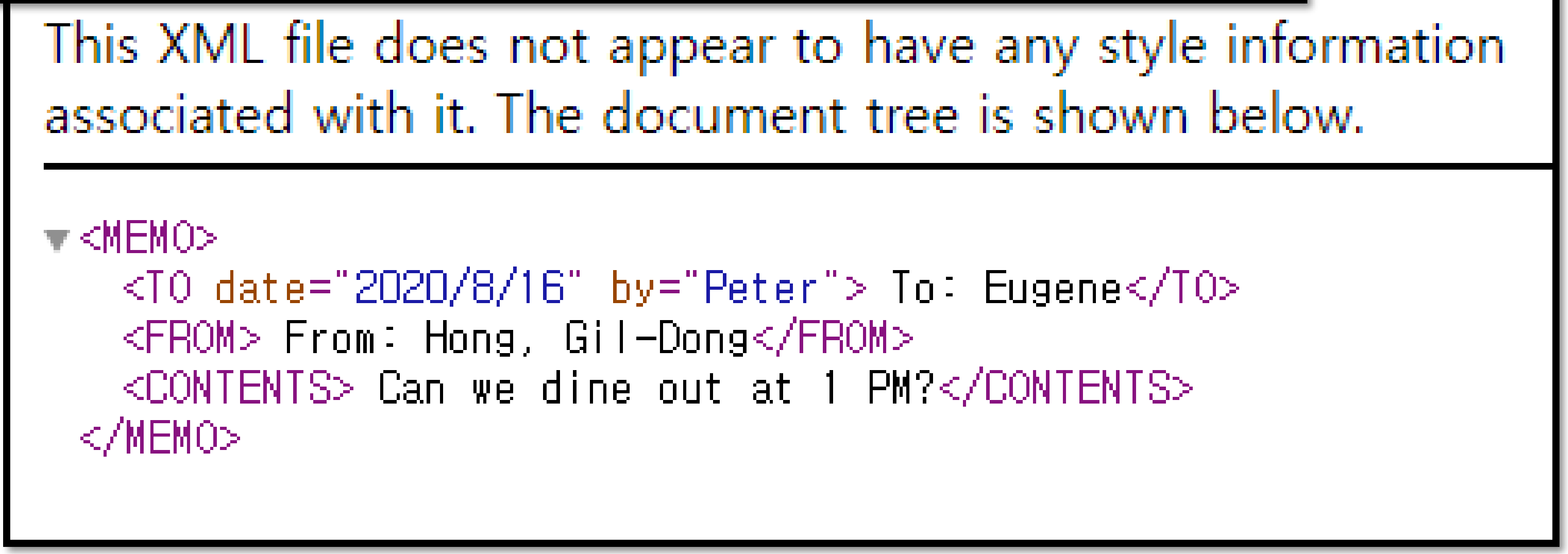
<img src="computer.gif" />

```
<?xml version="1.0" encoding="euc-kr" standalone="yes"?>
<MEMO>
   <TO date = '2020/8/16' by = 'Peter'> To: Eugene</TO>
   <FROM> From: Hong, Gil-Dong</FROM>
   <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
</MEMO>
```

Try also to use “**Code Beautify**” for validating attribute syntax.


XML Notepad - F:₩WEB2018_BU₩0_XML2018₩XML_Ex1_
File
Edit
View
Insert
Window
Help
F:₩WEB2018_BU₩0_XML2018₩XML_Ex1_2019Fall₩mem
Tree View
XSL Output

```
xml          version="1.0" encoding="euc-kr" standalone="yes"
MEMO
  TO
    date     2020/8/16
    by       Peter
    #text    To: Eugene
  FROM       From: Hong, Gil-Dong
  CONTENTS   Can we dine out at 1 PM?
```

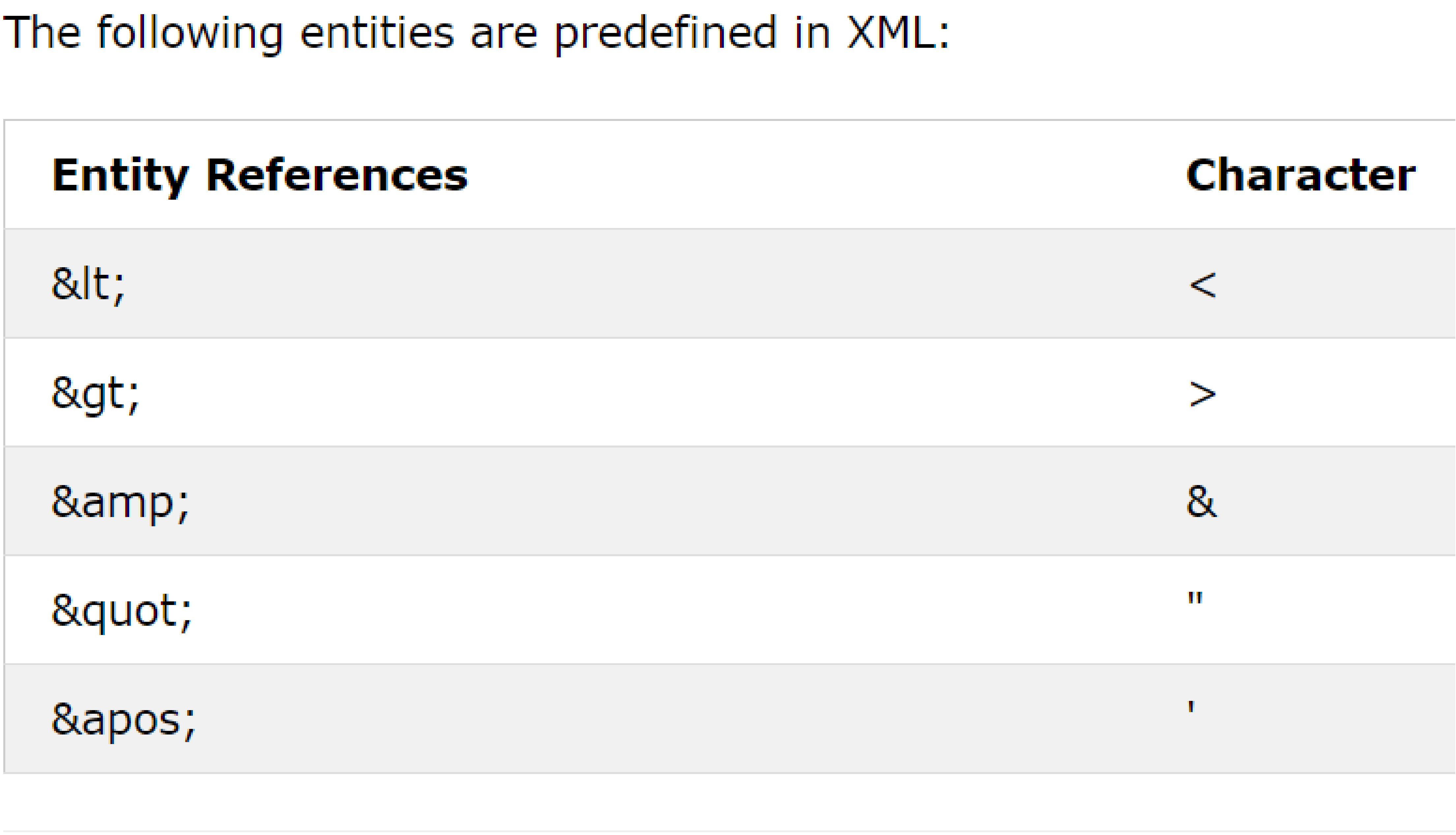


This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<MEMO>
   <TO date="2020/8/16" by="Peter"> To: Eugene</TO>
   <FROM> From: Hong, Gil-Dong</FROM>
   <CONTENTS> Can we dine out at 1 PM?</CONTENTS>
 </MEMO>
```

Entities

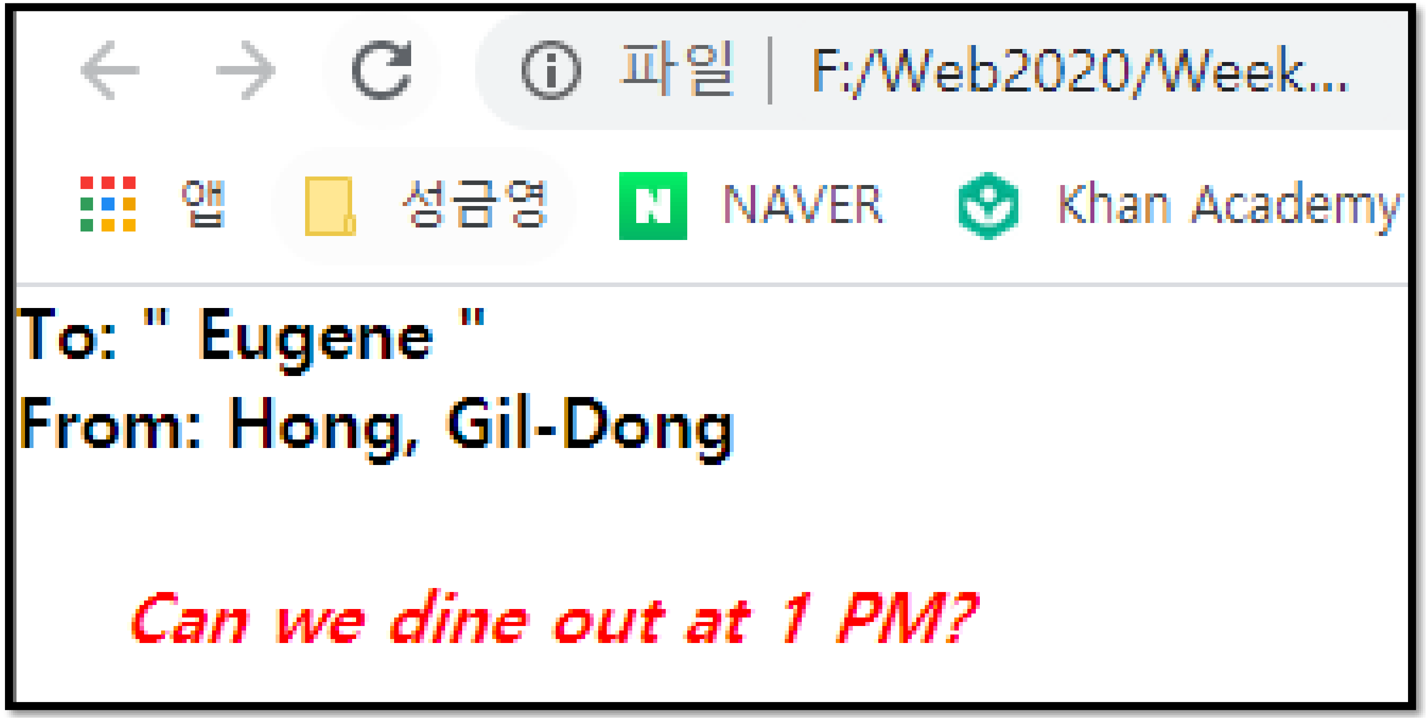
- Some characters have a **special meaning** in XML, like the less than sign (<) that defines the start of an XML tag.
- Most of you know the HTML entity: **"&nbsp**;". This "no-breaking-space" entity is used in HTML to insert an extra space in a document. **Entities are expanded when a document is parsed by an XML parser**.



The following entities are predefined in XML:

Entity References	Character
&lt;	<
&gt;	>
&amp;	&
&quot;	"
&apos;	'

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



```
To: " Eugene "
From: Hong, Gil-Dong


     Can we dine out at 1 PM?
```

PCDATA

- parsed character data.
- character data as the text found between the start tag and the end tag of an XML element.
- **PCDATA is text that WILL be parsed by a parser**.
- **The text will be examined by the parser** for entities and markup.
- should not contain any &, <, or > characters;
- represented by the &amp; &lt; and &gt;

CDATA

- CDATA means character data.
- **CDATA is text that will NOT be parsed by a parser**. Tags inside the text will NOT be treated as markup and entities and will not be expanded.

Declaring Elements

In a DTD, XML elements are declared with the following syntax:

<**!ELEMENT** element-name **category**>
or
<!ELEMENT element-name (**element-content**)>

Empty elements are declared with the category keyword **EMPTY**:

<!ELEMENT element-name **EMPTY**>

Example:

<!ELEMENT br **EMPTY**>

XML example:

<br />

Elements with Parsed Character Data

- Elements with only parsed character data are declared with #PCDATA inside parentheses:
- <!ELEMENT element-name (#PCDATA)>

  Example:

  <!ELEMENT from (#PCDATA)>

Elements with any Contents

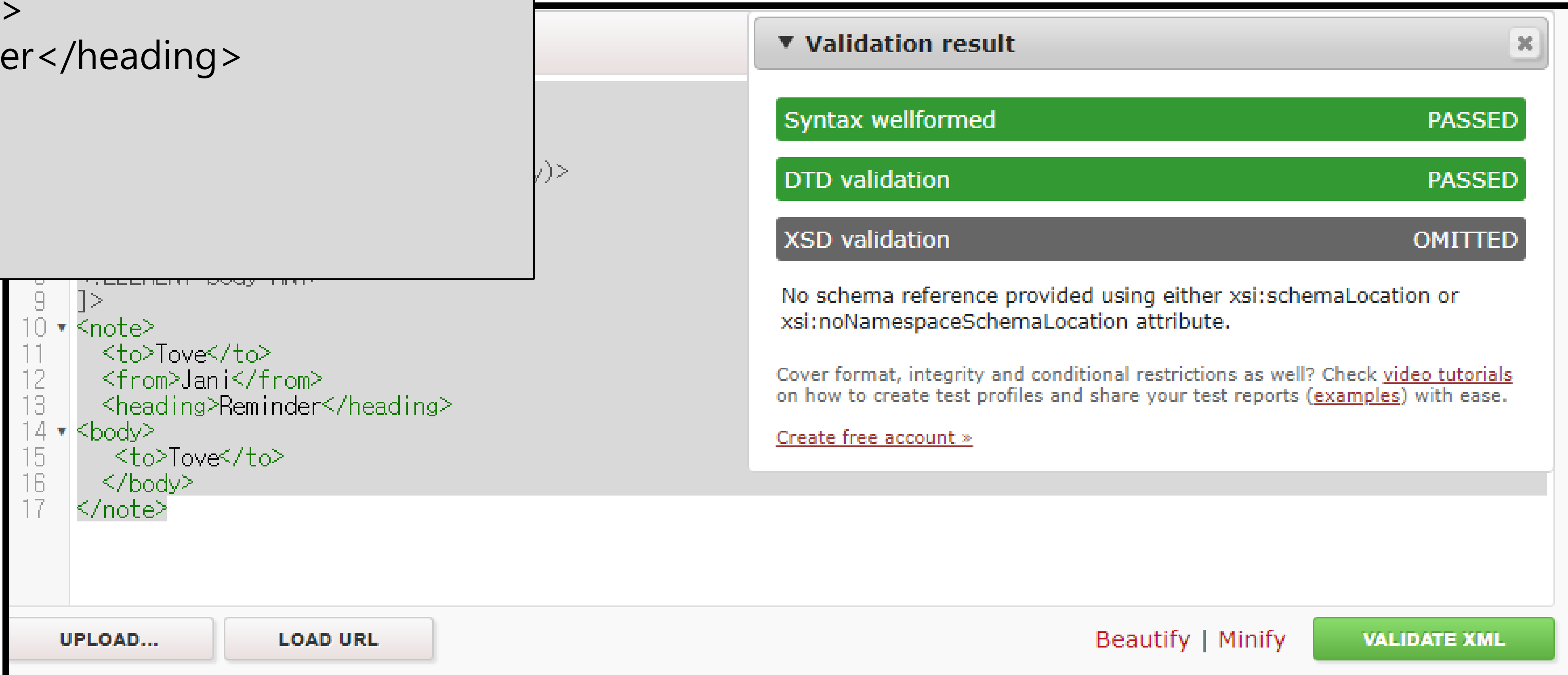
- Elements declared with the category keyword ANY, can contain any combination of parsable data:
- <!ELEMENT element-name ANY>

  Example:

  <!ELEMENT note ANY>

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body ANY>
]>
<note>
            <to>Tove</to>
            <from>Jani</from>
            <heading>Reminder</heading>
<body>
   <to>Tove</to>
  </body>
</note>
```

https://www.truugo.com/xml_validator/
Recommended Site



```
)>
]>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
<body>
   <to>Tove</to>
  </body>
</note>
```

▼ Validation result

Syntax wellformed PASSED

DTD validation PASSED

XSD validation OMITTED

No schema reference provided using either xsi:schemaLocation or xsi:noNamespaceSchemaLocation attribute.

Cover format, integrity and conditional restrictions as well? Check video tutorials on how to create test profiles and share your test reports (examples) with ease.

Create free account »

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

When children are declared in a sequence separated by commas, the children must appear in the same sequence

Declaring Only One Occurrence of an Element

```
<!ELEMENT element-name (child-name)>
```

Example:

```
<!ELEMENT note (message)>
```

Declaring Minimum One Occurrence of an Element

```
<!ELEMENT element-name (child-name+)>
```

Example:

```
<!ELEMENT note (message+)>
```

Declaring Zero or More Occurrences of an Element

```
<!ELEMENT element-name (child-name*)>
```

Example:

```
<!ELEMENT note (message*)>
```

Declaring Zero or One Occurrences of an Element

<!ELEMENT element-name (child-name?)>

Example:

<!ELEMENT note (message?)>

Declaring either/or Content

```
<!ELEMENT note (to,from,header,(message|body))>
```

Declaring Mixed Content

```
<!ELEMENT note (#PCDATA|to|from|header|message)*>
```

The example above declares that the "note" element can contain zero or more occurrences of parsed character data, "to", "from", "header", or "message" elements.

Important Guide with DTD Attribute

If you use attributes as containers for data, you end up with documents that are difficult to read and maintain. Try to use **elements** to describe data. Use attributes only to provide information that is not relevant to the data

```xml
<note date="12/11/2002">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

```xml
<note>
  <date>12/11/2002</date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

```xml
<note>
  <date>
    <day>12</day>
    <month>11</month>
    <year>2002</year>
  </date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Don't end up like this (this is not how XML should be used):

```
<note day="12" month="11" year="2002"
to="Tove" from="Jani" heading="Reminder"
body="Don't forget me this weekend!">
</note>
```