



DTD

(Document Type Definition)

- Purpose is to define the legal building blocks of an XML document
- It defines the document structure with a list of legal elements
- DTD can be declared
 - in XML document
 - external

Example – in XML

```
<?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>
```

Example - external

DTD file "note.dtd"

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

XML file

```
<?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>
```

Why to use DTD?

- With DTD:
 - XML file has a description of its own format
 - independent groups of people can agree to use a common DTD for interchanging data.

- Application can use DTD to validate XML file

XML building blocks

- From DTD point of view:
 - Elements
 - Tags
 - Attributes
 - Entities
 - PCDATA
 - CDATA

XML building blocks - Elements

- Elements are the **main building blocks** of both XML and HTML documents
- Examples of HTML elements:
 - "body", "table"
- Examples of XML elements:
 - "note", "message"
- Elements can contain:
 - text, other elements, or be empty.

Elements

- XML elements are declared with a DTD element declaration

```
<!ELEMENT element-name category>
```

or

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


Element category and content

■ Category

- EMPTY
- ANY

```
<!ELEMENT br EMPTY>
```

```
<!ELEMENT form ANY>
```

■ Content

- (#PCDATA)
- (child-element-name,
child-element-name, ...)

```
<!ELEMENT body (#PCDATA)>
```

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

- ...

Element category and content (2)

- *either/or*

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

- *mixed content*

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

Occurrence of the same element

Occurrence	Example
only one	<code><!ELEMENT note (message)></code>
one or more	<code><!ELEMENT note (message+)></code>
zero or more	<code><!ELEMENT note (message*)></code>
zero or one	<code><!ELEMENT note (message?)></code>

XML building blocks - Tags

- Tags are used **to markup elements**
- A starting tag like `<element_name>` marks up the **beginning** of an element
- ending tag like `</element_name>` marks up the **end** of an element.

body element marked up with body tags:

```
<body>body text in between</body>.
```

XML building blocks - Attributes

- Attributes provide **extra information about elements**
- Attributes always:
 - placed inside the starting tag of an element
 - come in name/value pairs

```

```

Attributes

- Attributes are declared with an ATTLIST declaration

```
<!ATTLIST element-name attribute-name  
attribute-type default-value>
```

DTD example:

```
<!ATTLIST payment type CDATA "check">
```

XML example:

```
<payment type="check" />
```

Attribute type

Value	Explanation: the value
CDATA	is character data
(en1 en2 ..)	must be one from an enumerated list
ID	is a unique id
IDREF	is the id of another element
IDREFS	is a list of other ids
NMTOKEN	is a valid XML name
NMTOKENS	is a list of valid XML names
ENTITY	is an entity
ENTITIES	is a list of entities
NOTATION	is a name of a notation
xml:	is a predefined xml value

Attribute default value

Value	Explanation
value	The default value of the attribute
#REQUIRED	The attribute value must be included in the element
#IMPLIED	The attribute does not have to be included
#FIXED value	The attribute value is fixed

DTD:

```
<!ELEMENT square EMPTY>
```

```
<!ATTLIST square width CDATA "0">
```

Valid XML:

```
<square width="100" />
```


Default value types

- #IMPLIED
- #REQUIRED
- #FIXED

DTD:

```
<!ATTLIST contact fax CDATA #IMPLIED>
```

Valid XML: `<contact fax="555-667788" />`

Valid XML: `<contact />`

DTD:

```
<!ATTLIST person number CDATA #REQUIRED>
```

Valid XML: `<person number="5677" />`

Invalid XML: `<person />`

DTD:

```
<!ATTLIST sender company CDATA #FIXED "Microsoft">
```

Valid XML: `<sender company="Microsoft" />`

Invalid XML: `<sender company="W3Schools" />`

XML building blocks - Entities

- Entities are variables used to **define common text**

Entities	Entity References
<	<
>	>
&	&
'	'
“	"

PCDATA, CDATA

PCDATA	CDATA
Parsed character data	Character data
It is the text between the start tag and the end tag of an element	
The text:	
will be parsed by parser	will NOT be parsed by a parser
Tags inside the text:	
will be treated as markup and entities will be expanded	will NOT be treated as markup and entities will not be expanded