# DTD - Overview

XML Document Type Declaration, commonly known as DTD, is a way to describe precisely the XML language. DTDs check the validity of structure and vocabulary of an XML document against the grammatical rules of the appropriate XML language.

An XML document can be defined as −

* **Well-formed** − If the XML document adheres to all the general XML rules such as tags must be properly nested, opening and closing tags must be balanced, and empty tags must end with '/>', then it is called as *well-formed*.

OR

* **Valid** − An XML document said to be valid when it is not only *well-formed*, but it also conforms to available DTD that specifies which tags it uses, what attributes those tags can contain, and which tags can occur inside other tags, among other properties.

## Types

DTD can be classified on its declaration basis in the XML document, such as −

* Internal DTD
* External DTD

When a DTD is declared within the file it is called **Internal DTD** and if it is declared in a separate file it is called **External DTD**. XML DTD can be either specified inside the document, or it can be kept in a separate document and then the document can be linked to the DTD document to use it.

## Syntax

Basic syntax of a DTD is as follows −

<!DOCTYPE element DTD identifier

[

declaration1

declaration2

........

]>

In the above syntax −

* **DTD** starts with <!DOCTYPE delimiter.
* An **element** tells the parser to parse the document from the specified root element.
* **DTD identifier** is an identifier for the document type definition, which may be the path to a file on the system or URL to a file on the internet. If the DTD is pointing to external path, it is called **external subset.**
* The **square brackets [ ]** enclose an optional list of entity declarations called **internal subset**

## Internal DTD

A DTD is referred to as an internal DTD if elements are declared within the XML files. To reference it as internal DTD, *standalone* attribute in XML declaration must be set to **yes**. This means the declaration works independent of external source.

### Syntax

The syntax of internal DTD is as shown −

<!DOCTYPE root-element [element-declarations]>

where *root-element* is the name of root element and *element-declarations* is where you declare the elements.

### Example

Following is a simple example of internal DTD −

<?xml version = "1.0" encoding = "UTF-8">

<!DOCTYPE address [

<!ELEMENT address (name,company,phone)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT company (#PCDATA)>

<!ELEMENT phone (#PCDATA)>

]>

<address>

<name>Tanmay Patil</name>

<company>GEHU</company>

<phone>(011) 123-4567</phone>

</address>

Let us go through the above code −

**Start Declaration** − Begin the XML declaration with following statement.

<?xml version = "1.0" encoding = "UTF-8" standalone = "yes" ?>

**DTD** − Immediately after the XML header, the *document type declaration* follows, commonly referred to as the DOCTYPE −

<!DOCTYPE address [

The DOCTYPE declaration has an exclamation mark (!) at the start of the element name. The DOCTYPE informs the parser that a DTD is associated with this XML document.

**DTD Body** − The DOCTYPE declaration is followed by body of the DTD, where you declare elements, attributes, entities, and notations −

<!ELEMENT address (name,company,phone)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT company (#PCDATA)>

<!ELEMENT phone\_no (#PCDATA)>

Several elements are declared here that make up the vocabulary of the <name> document. <!ELEMENT name (#PCDATA)> defines the element *name* to be of type "#PCDATA". Here #PCDATA means parse-able text data.

**End Declaration** − Finally, the declaration section of the DTD is closed using a closing bracket and a closing angle bracket (]>). This effectively ends the definition, and thereafter, the XML document follows immediately.

### Rules

* The document type declaration must appear at the start of the document (preceded only by the XML header) - it is not permitted anywhere else within the document.
* Similar to the DOCTYPE declaration, the element declarations must start with an exclamation mark.
* The Name in the document type declaration must match the element type of the root element.