DTD - ELEMENTS

XML elements can be defined as building blocks of an XML document. Elements can behave as a container to hold text, elements, attributes, media objects or mix of all.

A DTD element is declared with an ELEMENT declaration. When an XML file is validated by DTD, parser initially checks for the root element and then the child elements are validated.

Syntax

All DTD element declarations have this general form:

```
<!ELEMENT elementname (content)>
```

- *ELEMENT* declaration is used to indicate the parser that you are about to define an element.
- *elementname* is the element name (also called the *generic identifier*) that you are defining.
- *content* defines what content *ifany* can go within the element.

Element Content Types

Content of elements declaration in a DTD can be categorized as below:

- Empty content
- Element content
- Mixed content
- Any content

Empty Content

This is a special case of element declaration. This element declaration does not contain any content. These are declared with the keyword **EMPTY**.

Syntax

Following is the syntax for empty element declaration:

```
<!ELEMENT elementname EMPTY >
```

In the above syntax:

- **ELEMENT** is the element declaration of category *EMPTY*
- **elementname** is the name of empty element.

Example

Following is a simple example demonstrating empty element declaration:

```
<?xml version="1.0"?>
<!DOCTYPE hr[
   <!ELEMENT address EMPTY>
]>
<address />
```

In this example *address* is declared as an empty element. The markup for *address* element would appear as <address />.

Element Content

In element declaration with element content, the content would be allowable elements within parentheses. We can also include more than one element.

Syntax

Following is a syntax of element declaration with element content:

```
<!ELEMENT elementname (child1, child2...)>
```

- **ELEMENT** is the element declaration tag
- **elementname** is the name of the element.
- *child1*, *child2..* are the elements and each element must have its own definition within the DTD.

Example

Below example demonstrates a simple example for element declaration with element content:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE address [
   <!ELEMENT address (name,company,phone)>
   <!ELEMENT name (#PCDATA)>
   <!ELEMENT company (#PCDATA)>
   <!ELEMENT phone (#PCDATA)>
]>
<address>
   <name>Tanmay Patil</name>
   <company>TutorialsPoint</company>
   <phone>(011) 123-4567</phone>
</address>
```

In the above example, *address* is the parent element and *name*, *company* and *phone_no* are its child elements.

List of Operators and Syntax Rules

Below table shows the list of operators and syntax rules which can be applied in defining child elements:

Operator	Syntax	Description	Example
+	<!ELEMENT element-name *child1* + >	It indicates that child element can occur one or more times inside parent element.	<!ELEMENT address *name* + > Child element *name* can occur one or more times inside the element name *address*.
*	<!ELEMENT element-name *child1* * >	It indicates that child element can occur zero or more times inside parent element.	<!ELEMENT address *name* * > Child element *name* can occur zero or more times inside the element name *address*.
?	<!ELEMENT element-name *child1*?>	It indicates that child element can occur zero or one time inside parent element.	<!ELEMENT address *name*?> Child element *name* can occur zero or one time inside the element name *address*.

,	<!ELEMENT element-name *child1, child2*>	It gives sequence of child elements separated by comma which must be included in the the element-name.	<!ELEMENT address *name, company* > Sequence of child elements *name*, *company*, which must occur in the same order inside the element name *address*.
|	<!ELEMENT element-name *child1*|*child2*>	It allows making choices in the child element.	<!ELEMENT address *name*|*company* > It allows you to choose either of child elements i.e. *name* or *company*, which must occur in inside the element name *address*.

Rules

We need to follow certain rules if there is more than one element content:

- **Sequences -** Often the elements within DTD documents must appear in a distinct order. If this is the case, you define the content using a sequence. For example:

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

  The declaration indicates that the <address> element must have exactly three children - <name>, <company>, and <phone> - and that they must appear in this order.

- **Choices**: Suppose you need to allow one element or another, but not both. In such cases you must use the pipe | character. The pipe functions as an exclusive OR. For example:

  ```
  <!ELEMENT address (mobile | landline)>
  ```

Mixed Element Content

This is the combination of #PCDATA and children elements. PCDATA stands for parsed character data, that is, text that is not markup. Within mixed content models, text can appear by itself or it can be interspersed between elements. The rules for mixed content models are similar to the element content as discussed in the previous section.

Syntax

Following is a generic syntax for mixed element content:

```
<!ELEMENT elementname (#PCDATA|child1|child2)*>
```

- **ELEMENT** is the element declaration tag.
- **elementname** is the name of the element.
- **PCDATA** is the text that is not markup. #PCDATA must come first in the mixed content declaration.
- *child1, child2..* are the elements and each element must have its own definition within the DTD.
- The operator * must follow the mixed content declaration if children elements are included
- The #PCDATA and children element declarations must be separated by the | operator.

Example

Following is a simple example demonstrating the mixed content element declaration in a DTD.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE address [
   <!ELEMENT address (#PCDATA|name)*>
   <!ELEMENT name (#PCDATA)>

]>
<address>
   Here's a bit of text mixed up with the child element.
   <name>Tanmay Patil</name>
</address>
```

ANY Element Content

You can declare an element using the ANY keyword in the content. It is most often referred to as mixed category element. ANY is useful when you have yet to decide the allowable contents of the element.

Syntax

Following is the syntax for declaring elements with ANY content:

```
<!ELEMENT elementname ANY>
```

Here, the ANY keyword indicates that text *PCDATA* and/or any elements declared within the DTD can be used within the content of the <elementname> element. They can be used in any order any number of times. However, the ANY keyword does not allow you to include elements that are not declared within the DTD.

Example

Following is a simple example demonstrating the element declaration with ANY content:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE address [
   <!ELEMENT address ANY>
]>
<address>
  Here's a bit of sample text
</address>
```