

DTD

1. DTD: Document Type Declaration

2. Internal DTD

```
<!DOCTYPE root-element [DTD specification ]>
```

3. External DTD

```
<!DOCTYPE root-element SYSTEM "file-name">
```

Example

address.xml

```
<?xml version="1.0"?>
  <address>
    <street>12 City Road</street>
    <city>Melbourne</city>
    <state>Victoria</state>
    <zip>8001</zip>
  </address>
```

External DTD Example

address.xml

```
<?xml version="1.0"?>
<!DOCTYPE root-element SYSTEM "address.dtd">
  <address>
    <street>12 City Road</street>
    <city>Melbourne</city>
    <state>Victoria</state>
    <zip>8001</zip>
  </address>
```

address.dtd

```
<!ELEMENT address (street, city, state, zip)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT zip (#PCDATA)>
```

Internal DTD Example

```
<?xml version="1.0"?>
<!DOCTYPE address
[
<!ELEMENT address (street, city, state, zip)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT zip (#PCDATA)>
]>
```

```
<address>
  <street>12 City Road</street>
  <city>Melbourne</city>
  <state>Victoria</state>
  <zip>8001</zip>
</address>
```

Other Options

- EMPTY

```
<!ELEMENT married EMPTY>
```

- ANY

```
<!ELEMENT position ANY>
```

- Disjunction

```
<!ELEMENT address  
  ((street | pobox), city, state, zip)>
```

More Features

```
<!ELEMENT workdays (#PCDATA|(day+))>>
```

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

```
<workdays>N/A</workdays>
```

```
<workdays>
```

```
  <day> Monday</day>
```

```
  <day> Tuesday</day>
```

```
  <day> Friday</day>
```

```
</workdays>
```

Attributes

```
<salary currency="AUD">5600</salary>
```

```
<!ATTLIST salary currency #CDATA #IMPLIED>
```

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

```
<!ATTLIST salary currency #CDATA #IMPLIED "USA">
```

XML Schema

- XML Schema Definition (XSD)
- Go Beyond the capabilities of DTD
- Object Oriented
- Written in XML

Example

```
<xsd:schema xmlns:xsd=
  "http://www.w3.org/2001/XMLSchema"/>
<xsd:element name="address">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="street"
        type="xsd:string"/>
      <xsd:element name="city"
        type="xsd:string"/>
      <xsd:element name="state"
        type="xsd:string"/>
      <xsd:element name="zip"
        type="xsd:integer"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

An Alternative Method

```
<xsd:element name="address" type="adresstype"/>
```

```
<xsd:complexType name="adresstype">
```

```
  <xsd:sequence>
```

```
    <xsd:element name="street"
```

```
      type="xsd:string"/>
```

```
    <xsd:element name="city"
```

```
      type="xsd:string"/>
```

```
    <xsd:element name="state"
```

```
      type="xsd:string"/>
```

```
    <xsd:element name="zip"
```

```
      type="xsd:integer"/>
```

```
  </xsd:sequence>
```

```
</xsd:complexType>
```

Indicators

- `<all>`

```
<xsd:complexType>
  <xsd:all>
    .
    .
  </xsd:all>
</xsd:complexType>
```

- `<choice>`

```
<xsd:complexType>
  <xsd:choice>
    <xsd:element name="street" type="xsd:string" />
    <xsd:element name="pobox" type="xsd:string" />
  </xsd:choice>
</xsd:complexType>
```

- `<sequence>`

Simple Schema Elements

```
<xs:element name="xxx" type="yyy"/>
```

```
<xs:element name="city" type="xsd:string"/>
```

- xsd:string
- xsd:decimal
- xsd:integer
- xst:float
- xsd:boolean
- xsd:date
- xsd:time

Declaring Attributes

```
<xs:attribute name="xxx" type="yyy"/>
```

```
<salary currency="AUD">5600</Salary>
```

```
<xsd:element name="salary">  
  <xsd:complexType>  
    <xsd:simpleContent>  
      <xsd:extension base="xsd:integer" >  
        <xsd:attribute name="currency"  
          type="xsd:string"/>  
      </xsd:extension>  
    </xsd:simpleContent>  
  </xsd:complexType>  
</xsd:element>
```

Example

```
<?xml version="1.0"?>
<address xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "http://www.monmouth.edu/address.xsd">
  <address>
    <street>12 City Road</street>
    <city>Melbourne</city>
    <state>Victoria</state>
    <zip>8001</zip>
  </address>
```