Ch. 7: Entities and Notations in DTDs

Seoul National University, Internet Database Laboratory

July, 2015

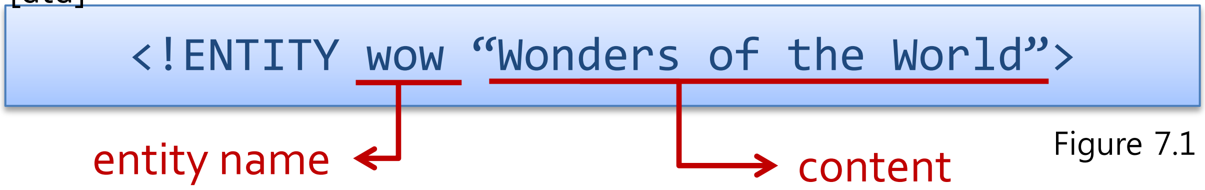
Contents

- Creating a General Entity
- Using General Entities
- Creating an External General Entity
- Using External General Entities
- Creating Entities for Unparsed Content
- Embedding Unparsed Content
- Creating and Using Parameter Entities
- Creating an External Parameter Entity

Creating a General Entity

- Entity is a shortcut of the text
- Internal general entities
  - Are defined in a DTD and simply represent text

[dtd]



```
<!ENTITY wow “Wonders of the World”>
```

Figure 7.1

- Entity name must follow the rules for valid XML names
- Five built-in general entities
  - ***&amp;***, ***&lt;***, ***&gt;***, ***&quot;***, ***&apost;***

Using General Entities inside XML

[dtd]

```
<!ENTITY wow “Wonders of the World”>
```

[xml]

```
<story>
  The first and most interesting fact
  about the gardens is that there is
  significant controversy about
  whether the gardens existed at all.
...
  Regardless of the final outcome.
  it is interesting to note that the
  imagination of the poets and
  ancient historians have created one
  of the &wow;
</story>
```

Figure 7.2

‘&’ + Entity name + ‘;’

[html]

```
...
<p align="center">
  <strong>HANGING GARDENS OF BABYLON
    </strong><br>
  <img src="gardens.jpg"
    width="264" height="175"></p>
  The Hanging Gardens of Babylon was
    built in 600 BC and was destroyed
    by earthquake in 226 BC.

  The first and most interesting fact
    about the gardens is that there
    is significant controversy about
    whether the gardens existed at all.
   <br><br>
...
  Regardless of the final outcome, it
    is interesting to note that the
    imagination of the poets and
    ancient historians have created
    one of the Wonders of the World.
    <br><br>
...
```

Figure 7.3

&wow “expands” when XML is parsed

Using General Entities

- **Use** general entity **after** it has been **defined** in the DTD
  - If not, the parser will return an error

- Special symbols in entity reference tables of Unicode
  - E.g.) **&246**; generate the ö symbol
  - They are not general entities
    - Do not need to be declared in the DTD

- General entities are used in **XML document only**
  - Cannot be used in XSLT documents

- You may use an entity within another entity's definition
  - As long as there is no circular reference

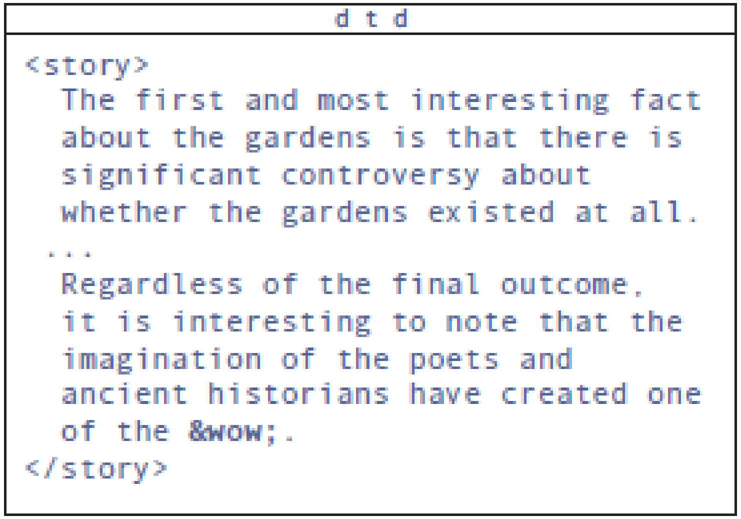
Creating an External General Entity

- Can be reused in multiple DTD documents
- Save it in a separate external document
- Create the content for the entity in an external file
- Save the file as text-only using an extension of ***.ent***
  - Not required, But most common

[dtd]

```
<!ENTITY wow “Wonders of the World”>
```

Figure 7.4

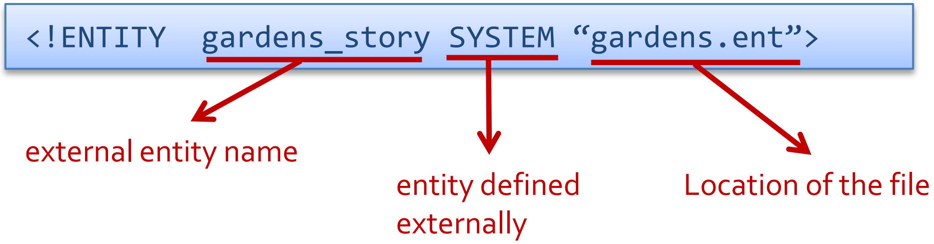


```
dtd
<story>
  The first and most interesting fact
  about the gardens is that there is
  significant controversy about
  whether the gardens existed at all.
 ...
  Regardless of the final outcome,
  it is interesting to note that the
  imagination of the poets and
  ancient historians have created one
  of the &wow;.
</story>
```

gardens.ent

[dtd]

Figure 7.5



```
<!ENTITY  gardens_story SYSTEM “gardens.ent”>
```

external entity name

entity defined externally

Location of the file

&garden_story: garden.ent에 들어있는 story tag의 contents

Using External General Entities

- To use external general entities

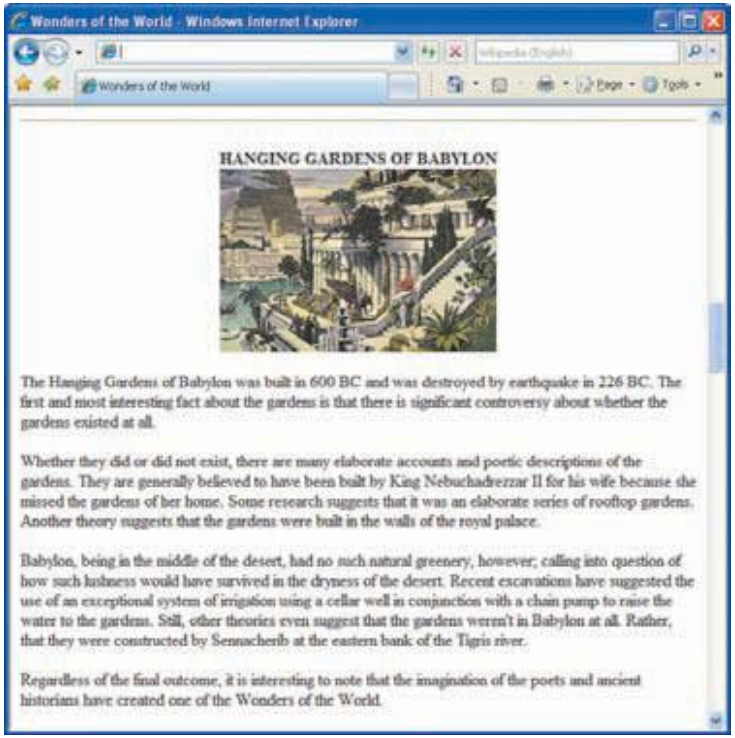
[xml]

```
<?xml version="1.0" standalone="no"?>
...
<wonder>
  <name  language="English">
    Hanging Gardens of Babylon</name>
  <location>Al Hillah, Iraq</location>
  <height units="feet">0</height>
  <history>
    <year_built era="BC">
      600</year_built>
    <year_destroyed era="BC">
      226</year_destroyed>
    <how_destroyed>
      earthquake</how_destroyed>
    &gardens_story;
  </history>
...
```

Tells the XML parser that the document will rely on an external file

'&' + entity name + ';'

Figure 7.6



HANGING GARDENS OF BABYLON

The Hanging Gardens of Babylon was built in 600 BC and was destroyed by earthquake in 226 BC. The first and most interesting fact about the gardens is that there is significant controversy about whether the gardens existed at all.

Whether they did or did not exist, there are many elaborate accounts and poetic descriptions of the gardens. They are generally believed to have been built by King Nebuchadrezzar II for his wife because she missed the gardens of her home. Some research suggests that it was an elaborate series of rooftop gardens. Another theory suggests that the gardens were built in the walls of the royal palace.

Babylon, being in the middle of the desert, had no such natural greenery, however, calling into question of how such lushness would have survived in the dryness of the desert. Recent excavations have suggested the use of an exceptional system of irrigation using a cellar well in conjunction with a chain pump to raise the water to the gardens. Still, other theories even suggest that the gardens weren't in Babylon at all. Rather, that they were constructed by Sennacherib at the eastern bank of the Tigris river.

Regardless of the final outcome, it is interesting to note that the imagination of the poets and ancient historians have created one of the Wonders of the World.

Figure 7.7

Creating Entities for Unparsed Content

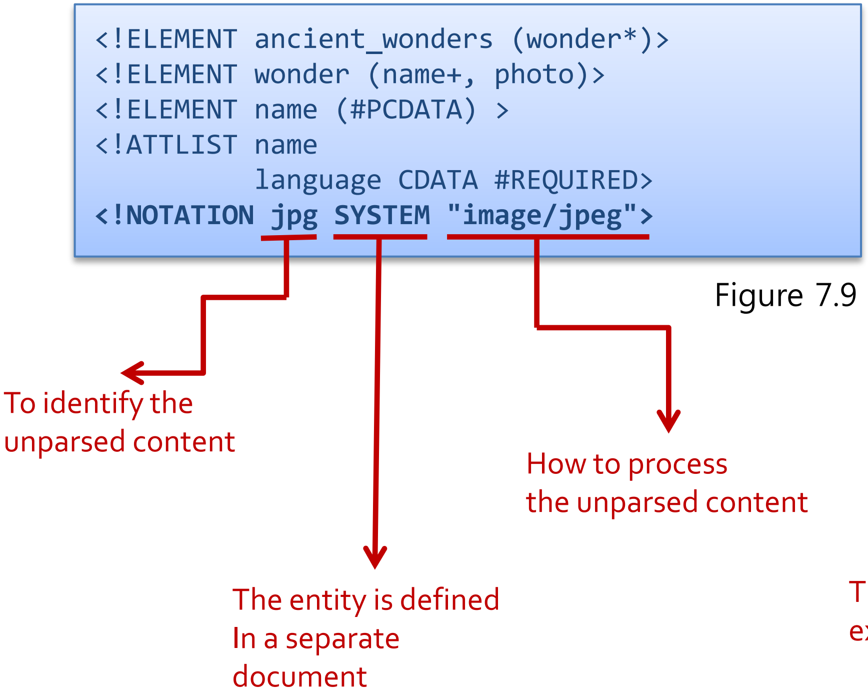
- Parsed entities
  - Entities that contain text
  - The XML parser looks at them and analyzes them
- Unparsed entities
  - Anything; plain text, an image file, a video file, a PDF file, or anything else
  - Bypassed by the XML parser
  - Embed non-text or non-XML content into an XML document
- Typical unparsed data : A JPEG image – lighthouse.jpg

Figure 7.8

Creating Entities for Unparsed Content

- To create a notation about the unparsed content

[dtd]



```
<!ELEMENT ancient_wonders (wonder*)>
<!ELEMENT wonder (name+, photo)>
<!ELEMENT name (#PCDATA) >
<!ATTLIST name
          language CDATA #REQUIRED>
<!NOTATION jpg SYSTEM "image/jpeg">
```

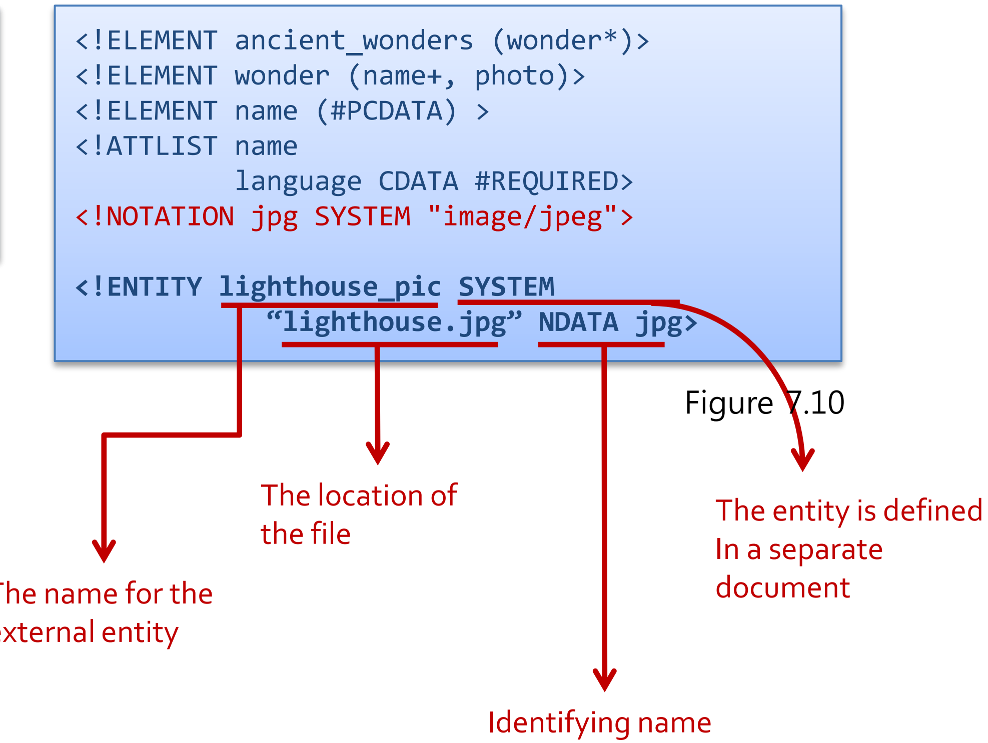
Figure 7.9

To identify the unparsed content

How to process the unparsed content

The entity is defined In a separate document

[dtd]



```
<!ELEMENT ancient_wonders (wonder*)>
<!ELEMENT wonder (name+, photo)>
<!ELEMENT name (#PCDATA) >
<!ATTLIST name
          language CDATA #REQUIRED>
<!NOTATION jpg SYSTEM "image/jpeg">

<!ENTITY lighthouse_pic SYSTEM
            “lighthouse.jpg” NDATA jpg>
```

Figure 7.10

The location of the file

The entity is defined In a separate document

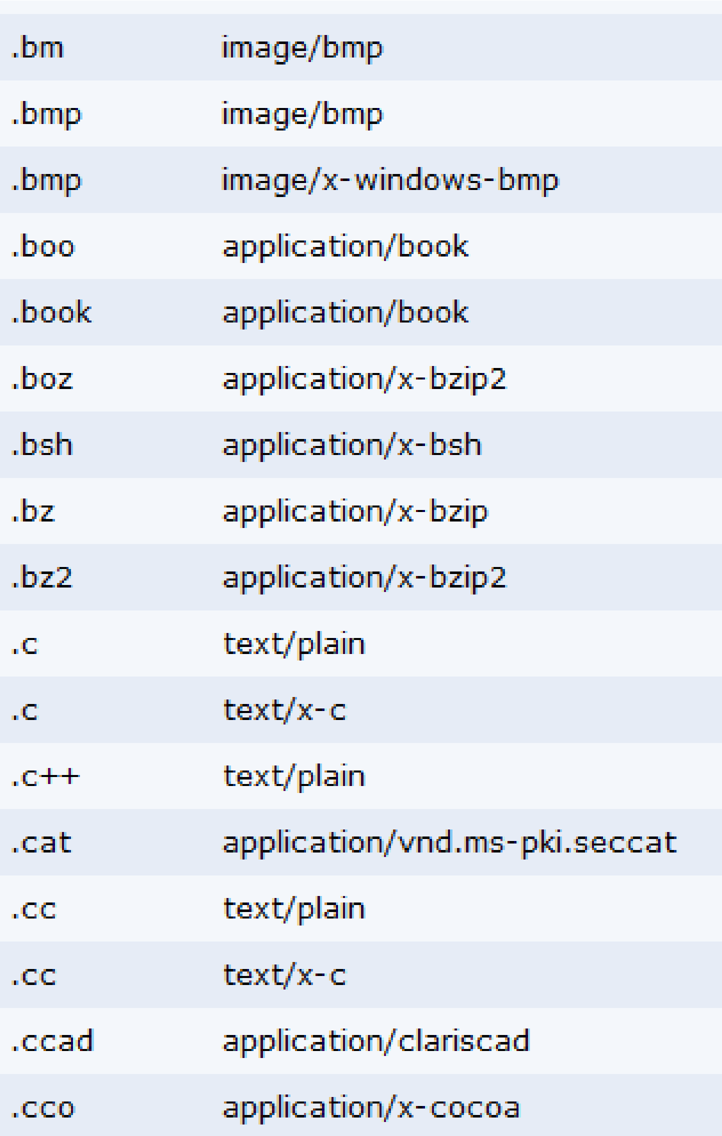
The name for the external entity

Identifying name

Creating Entities for Unparsed Content

- "image/jpeg" in `<!NOTATION jpg SYSTEM "image/jpeg">`
- can be
  - a MIME type
  - A URI
  - Anything else

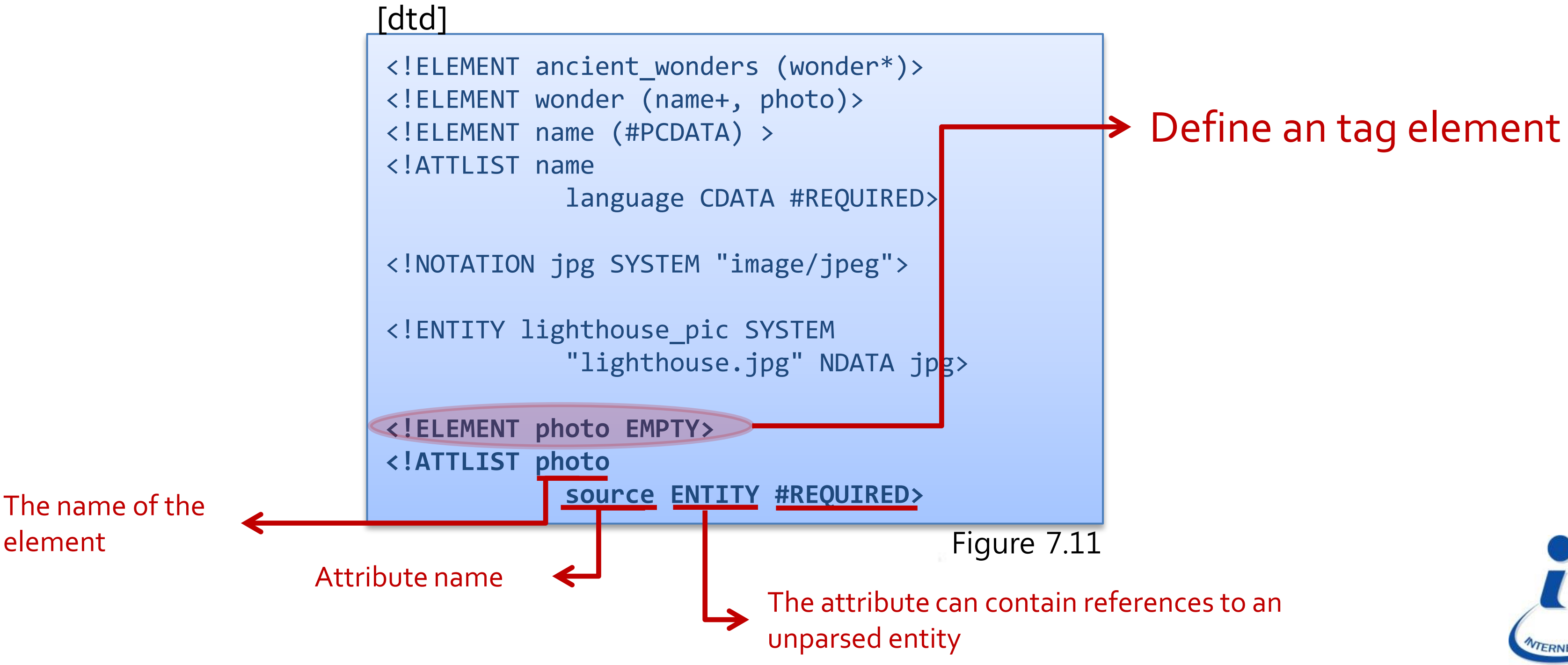
The sample list of MIME type



.bm	image/bmp
.bmp	image/bmp
.bmp	image/x-windows-bmp
.boo	application/book
.book	application/book
.boz	application/x-bzip2
.bsh	application/x-bsh
.bz	application/x-bzip
.bz2	application/x-bzip2
.c	text/plain
.c	text/x-c
.c++	text/plain
.cat	application/vnd.ms-pki.seccat
.cc	text/plain
.cc	text/x-c
.ccad	application/clariscad
.cco	application/x-cocoa

Embedding Unparsed Content using Tag Element

- Through a special ENTITY attribute type for unparsed content
- Declare the source attribute that will contain the reference to the unparsed entity



[dtd]

```
<!ELEMENT ancient_wonders (wonder*)>
<!ELEMENT wonder (name+, photo)>
<!ELEMENT name (#PCDATA) >
<!ATTLIST name
            language CDATA #REQUIRED>

<!NOTATION jpg SYSTEM "image/jpeg">

<!ENTITY lighthouse_pic SYSTEM
            "lighthouse.jpg" NDATA jpg>

<!ELEMENT photo EMPTY>
<!ATTLIST photo
            source ENTITY #REQUIRED>
```

Figure 7.11

Define an tag element

The name of the element

Attribute name

The attribute can contain references to an unparsed entity

Embedding Unparsed Content in XML

- In an XML document

[xml]

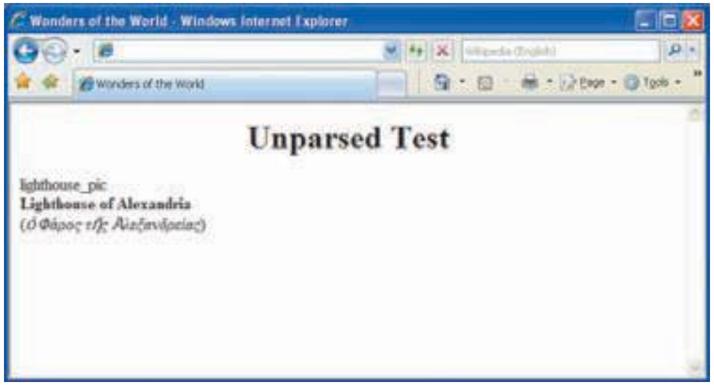
```
<?xml version="1.0" standalone="no"?>
...
<ancient_wonders>
  <wonder>
    <name language="English">
      Lighthouse of Alexandria</name>
    <name language="Greek">
      ʽο Φάρος τ˜ης ’Αλεξανδρείας</name>
    <photo source="lighthouse_pic" />
  </wonder>
</ancient_wonders>
```

The document will rely on an external file

Tag Element name

Attribute_name = “entity name”

Figure 7.12



Unparsed Test

lighthouse_pic
Lighthouse of Alexandria
(ὁ Φάρος τῆς Ἀλεξανδρείας)

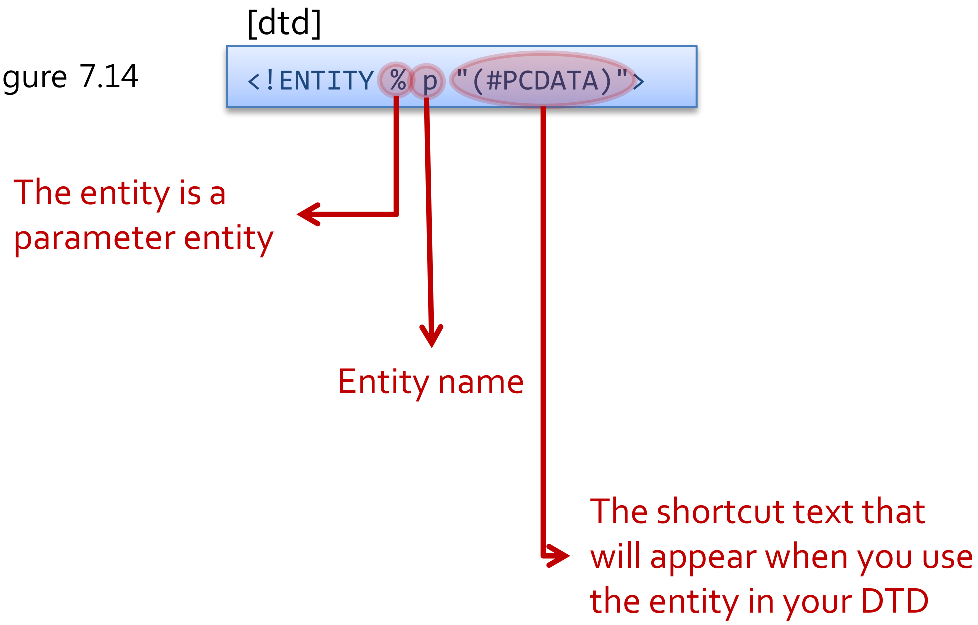
Figure 7.13

- Tips
  - Current browsers cannot view/play/display the unparsed entity
  - Instead of unparsed entities, you could set an element’s value to a URL

Creating and Using Parameter Entities

- Parameter entities are entities for the DTD
- To create a parameter entity

Figure 7.14



[dtd]

```
<!ENTITY % p "(#PCDATA)">
```

The entity is a parameter entity
Entity name
The shortcut text that will appear when you use the entity in your DTD

Figure 7.15

[dtd]

```
<!ENTITY % p "(#PCDATA)">
<!ELEMENT ancient_wonders (wonder+)>
<!ELEMENT wonder (name+, location,
    height, history, main_image,
    source*)>
<!ELEMENT name %p;>
<!ATTLIST name
            language CDATA #REQUIRED>
<!ELEMENT location %p;>
<!ELEMENT height %p;>
<!ATTLIST height
            units CDATA #REQUIRED>
<!ELEMENT history (year_built,
    year_destroyed?, how_destroyed?,
    story)>
<!ELEMENT year_built %p;>
<!ATTLIST year_built
            era (BC | AD) #REQUIRED>
<!ELEMENT year_destroyed %p;>
<!ATTLIST year_destroyed
            era (BC | AD) #REQUIRED>
<!ELEMENT how_destroyed %p;>
<!ELEMENT story (#PCDATA | para)*>
<!ELEMENT para EMPTY>
<!ELEMENT main_image EMPTY>
<!ATTLIST main_image
            file CDATA #REQUIRED
            h CDATA #REQUIRED
            w CDATA #REQUIRED>
<!ELEMENT source EMPTY>
<!ATTLIST source
            sectionid CDATA #REQUIRED
            newspaperid CDATA #REQUIRED>
```

'%' + entity name +';'

Creating an External Parameter Entity

- To create an external parameter entity
  - Create the entity’s content in an external file
  - Save it as text only sing an **.ent** extension

[dtd]

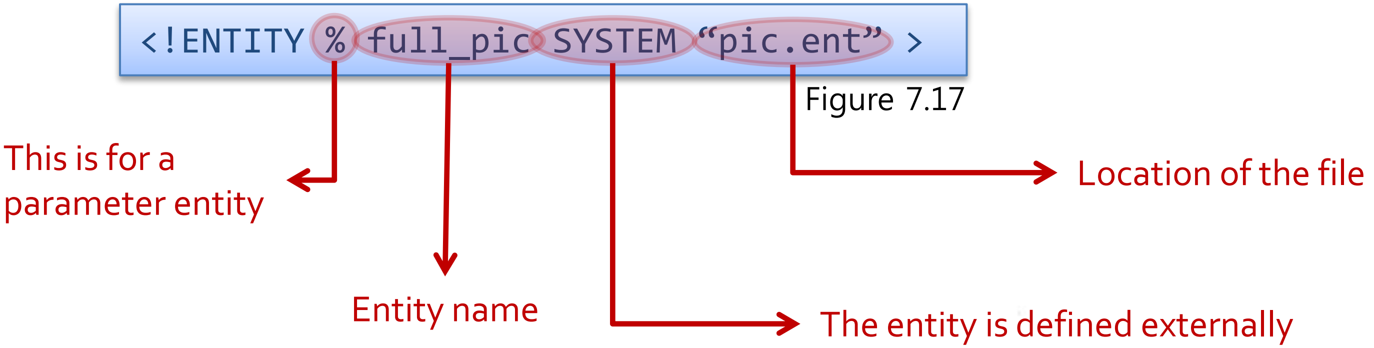
```
<!ELEMENT main_image EMPTY>
<!ATTLIST main_image
         file CDATA #REQUIRED
         w CDATA #REQUIRED
         h CDATA #REQUIRED>
```

pic.ent

Figure 7.16

- To define an external parameter entity

[dtd]



```
<!ENTITY % full_pic SYSTEM “pic.ent” >
```

Figure 7.17

This is for a parameter entity
Entity name
The entity is defined externally
Location of the file

Creating an External Parameter Entity

- To use external parameter entity [dtd]

```
<!ENTITY % full_pic SYSTEM "pic.ent">
<!ELEMENT ancient_wonders (wonder+)>
<!ELEMENT wonder (name+, location,
  height, history, main_image,
  source*)>
...
<!ELEMENT story (#PCDATA | para)*>
<!ELEMENT para EMPTY>
%full_pic;
<!ELEMENT source EMPTY>
<!ATTLIST source
         sectionid CDATA #REQUIRED
         newspaperid CDATA #REQUIRED>
```

Figure 7.18

Same way you would an internal parameter entity

```
<?xml version="1.0" standalone="no"?>
...
<ancient_wonders>
  <wonder>
    <name language="English">
      Lighthouse of Alexandria</name>
    <name language="Greek">
      ʽο Φάρος τ˜ης ’Αλεξανδρείας </name>
    <photo source="lighthouse_pic" />
  </wonder>
</ancient_wonders>
```

Figure 7.19

- If you are using an internal DTD
  - Then in the XML document containing the DTD
  - Add **standalone="no"**
    - This tells the XML parser that the document will rely on an external file