Ch. 8: Validation and Using DTDs

Seoul National University, Internet Database Laboratory

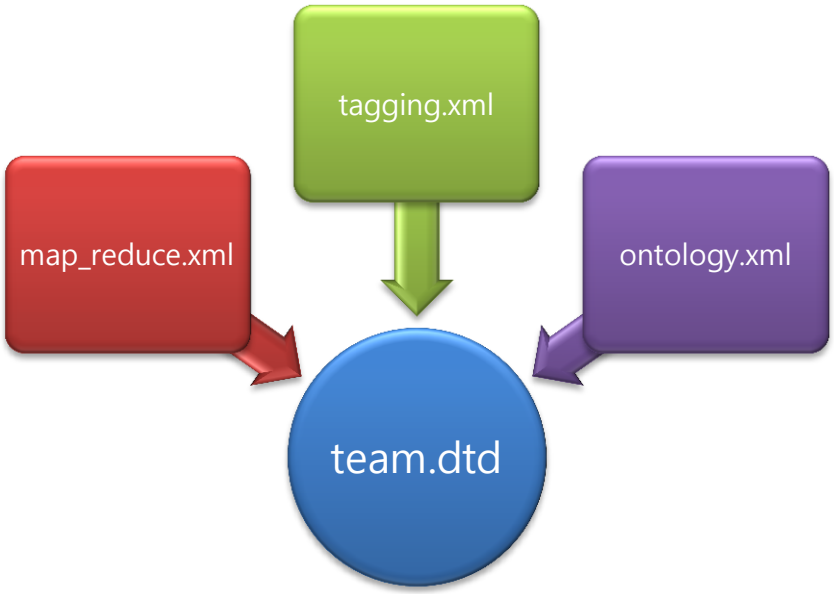
July, 2015

Contents

- Creating an External DTD
- Declaring an External DTD
- Declaring and Creating an Internal DTD
- Validating XML Documents Against a DTD
- Naming a Public External DTD
- Declaring a Public External DTD
- Pros and Cons of DTDs

Creating an External DTD

- You can create an external file that contains the DTD



[dtd] **08-01.dtd**

```
<!ELEMENT wonder (name, location, height)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT location (#PCDATA)>
<!ELEMENT height (#PCDATA)>
```

Figure 8.1

Declaring an External DTD

- You need to refer to the external DTD within your XML document

[08-01.dtd]

```
<!ELEMENT wonder (name, location, height)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT location (#PCDATA)>
<!ELEMENT height (#PCDATA)>
```

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE wonder SYSTEM "08-01.dtd">
<wonder>
  <name>Colossus of Rhodes</name>
  <location>Greece</location>
  <height>107</height>
</wonder>
```

This tells the XML parser that **the document will rely on an external file**

The location **URI** of the DTD content file

Corresponds to **the name of the root element** in this XML document

Figure 8.2-a

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE wonder SYSTEM "08-01.dtd">
<wonder>
  <name>Colossus of Rhodes</name>
  <location>Greece</location>
  <height units="feet">107</height>
</wonder>
```

not defined **attribute**

Figure 8.2-b

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE wonder SYSTEM "08-01.dtd">
<wonder>
  <name>Colossus of Rhodes</name>
  <name language="Greek">
    Κολοσσός της Ρόδου</name>
  <location>Greece</location>
  <height>107</height>
</wonder>
```

two name **elements**

Figure 8.2-c

Declaring and Creating an Internal DTD

- It is simplest to declare and create the DTD within the XML

Corresponds to **the name of the root element** in this XML document

```
<?xml version="1.0"?>
<!DOCTYPE wonder [
  <!ELEMENT wonder (name, location, height)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT location (#PCDATA)>
  <!ELEMENT height (#PCDATA)>
]>
<wonder>
  <name>Colossus of Rhodes</name>
  <location>Greece</location>
  <height>107</height>
</wonder>
```

Figure 8.3

Declaring and Creating an Internal DTD

- Combining both an internal and external DTD

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE wonder SYSTEM "08-01.dtd">
<wonder>
  <name>Colossus of Rhodes</name>
  <location>Greece</location>
  <height units="feet">107</height>
</wonder>
```

Figure 8.2-b

```
<?xml version="1.0"?>
<!DOCTYPE wonder SYSTEM "08-01.dtd"[
  <!ATTLIST height units CDATA #REQUIRED>
]>
<wonder>
  <name>Colossus of Rhodes</name>
  <location>Greece</location>
  <height units="feet">107</height>
</wonder>
```

define unit **attribute** for height **element**

Figure 8.4

Declaring and Creating an Internal DTD

- Combining both an internal and external DTD

```
<?xml version="1.0“ standalone="no"?>
<!DOCTYPE wonder SYSTEM "08-01.dtd">
<wonder>
  <name>Colossus of Rhodes</name>
  <name language="Greek">
    Κολοσσός της Ρόδου</name>
  <location>Greece</location>
  <height>107</height>
</wonder>
```

Figure 8.2-c

illegal to redefine wonder **element**

```
<?xml version="1.0"?>
<!DOCTYPE wonder SYSTEM "08-01.dtd"[
<!ELEMENT wonder (name*, location,height)>
<!ATTLIST name
          language CDATA #IMPLIED>
]>
<wonder>
  <name>Colossus of Rhodes</name>
  <name language="Greek">
    Κολοσσός της Ρόδου</name> ...
```

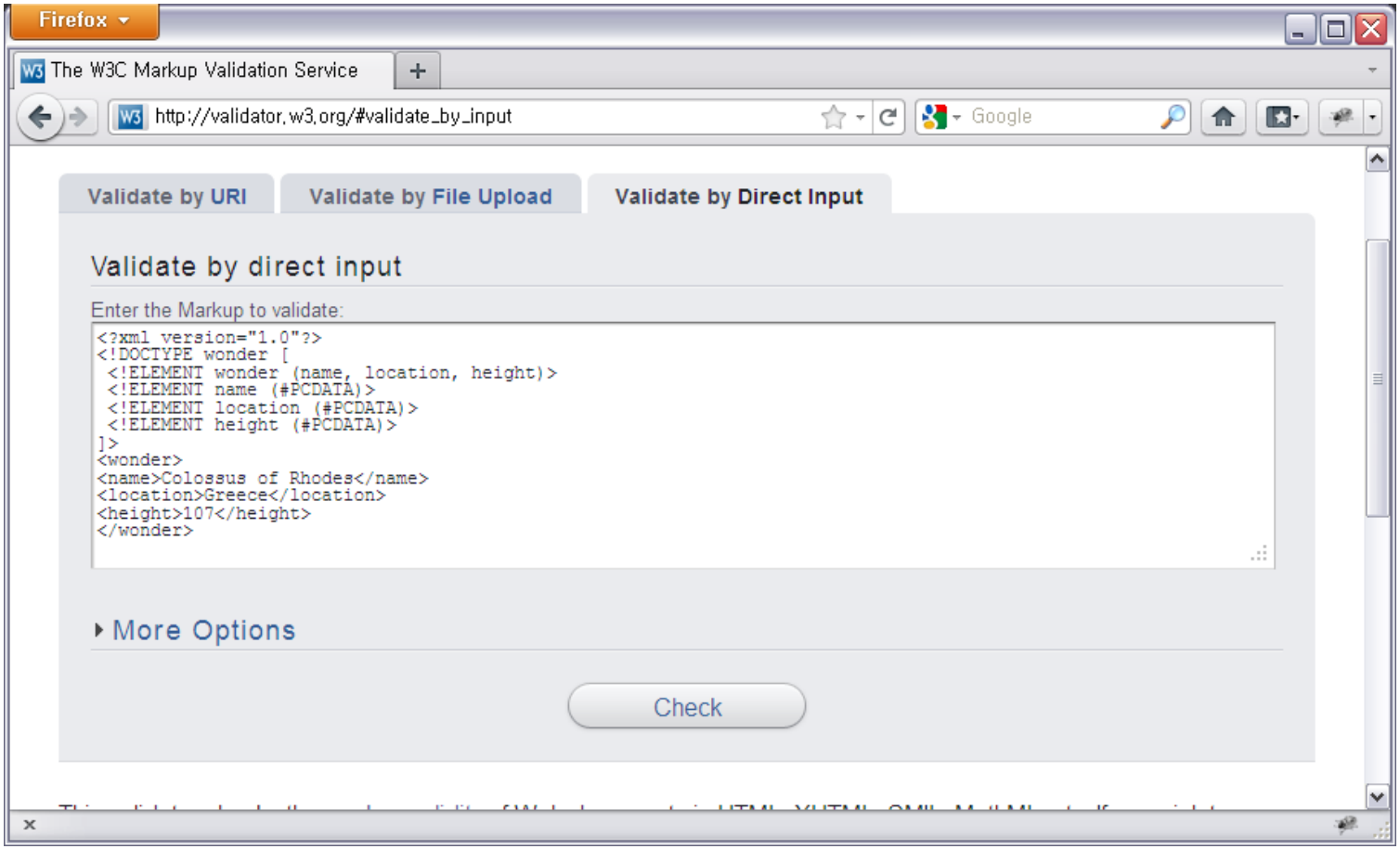
Figure 8.5

Validating XML Documents Against a DTD

- http://validator.w3.org/

Validating XML Documents Against a DTD

- http://validator.w3.org/



Validating XML Documents Against a DTD

- http://validator.w3.org/

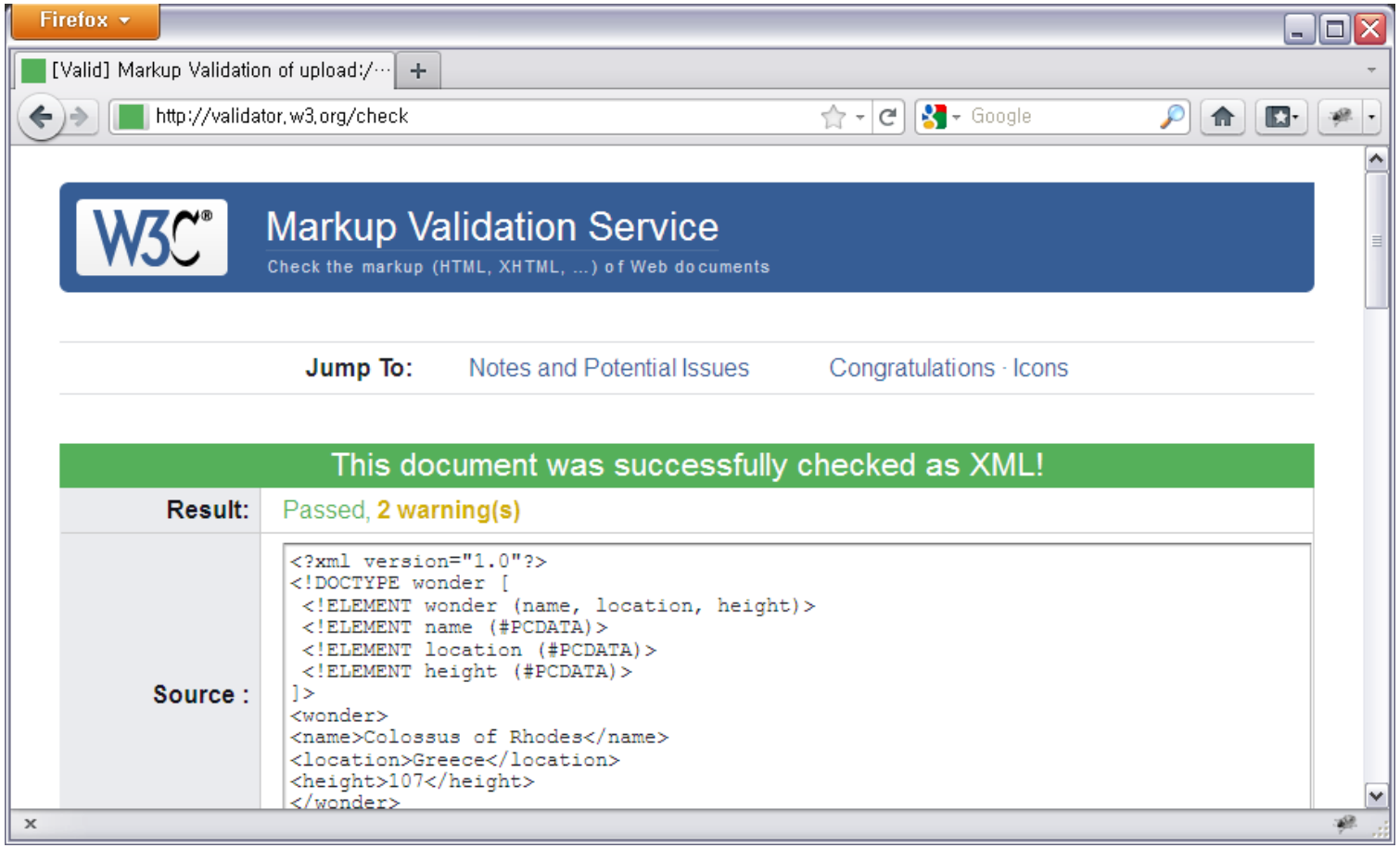
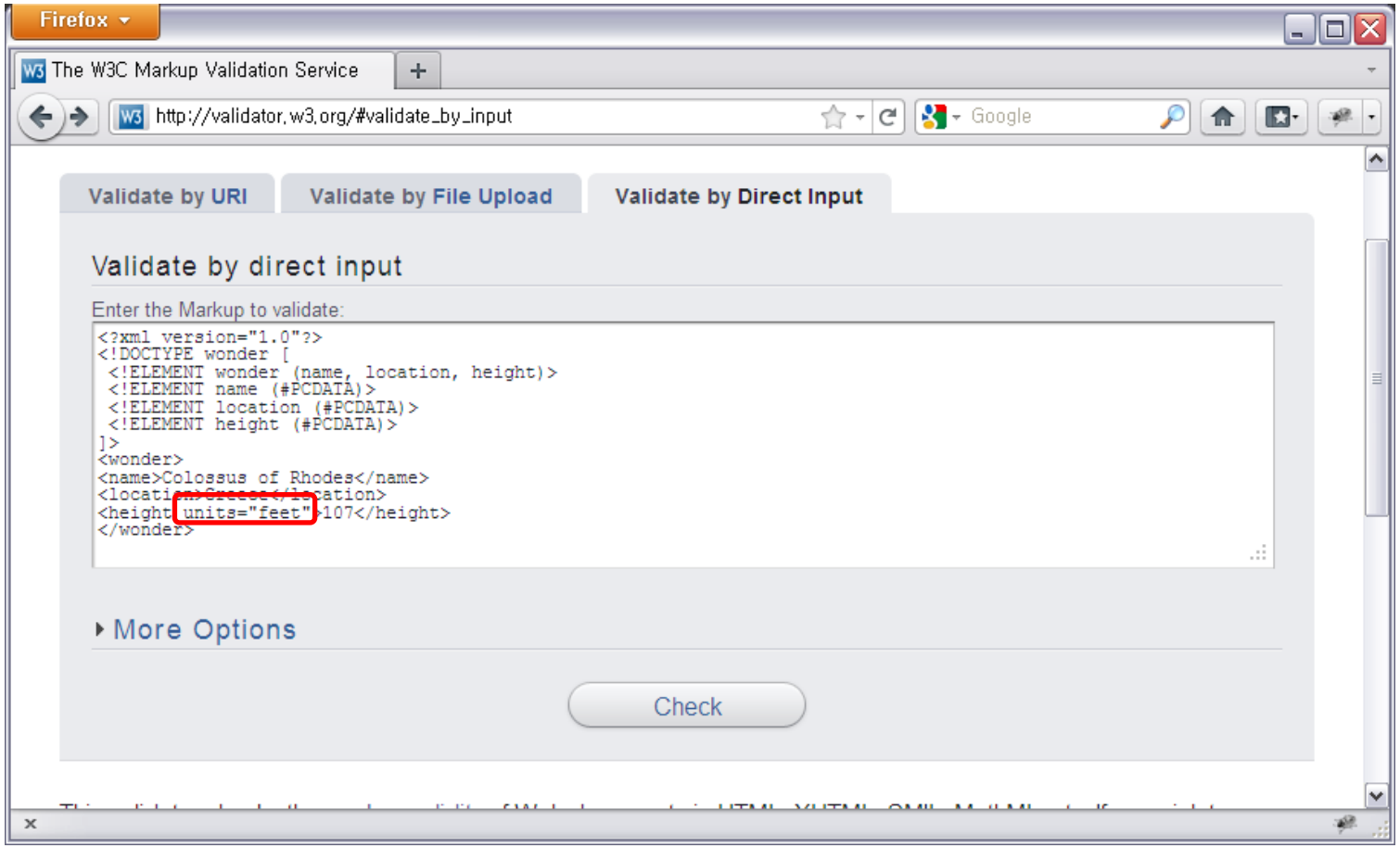


Figure 8.6

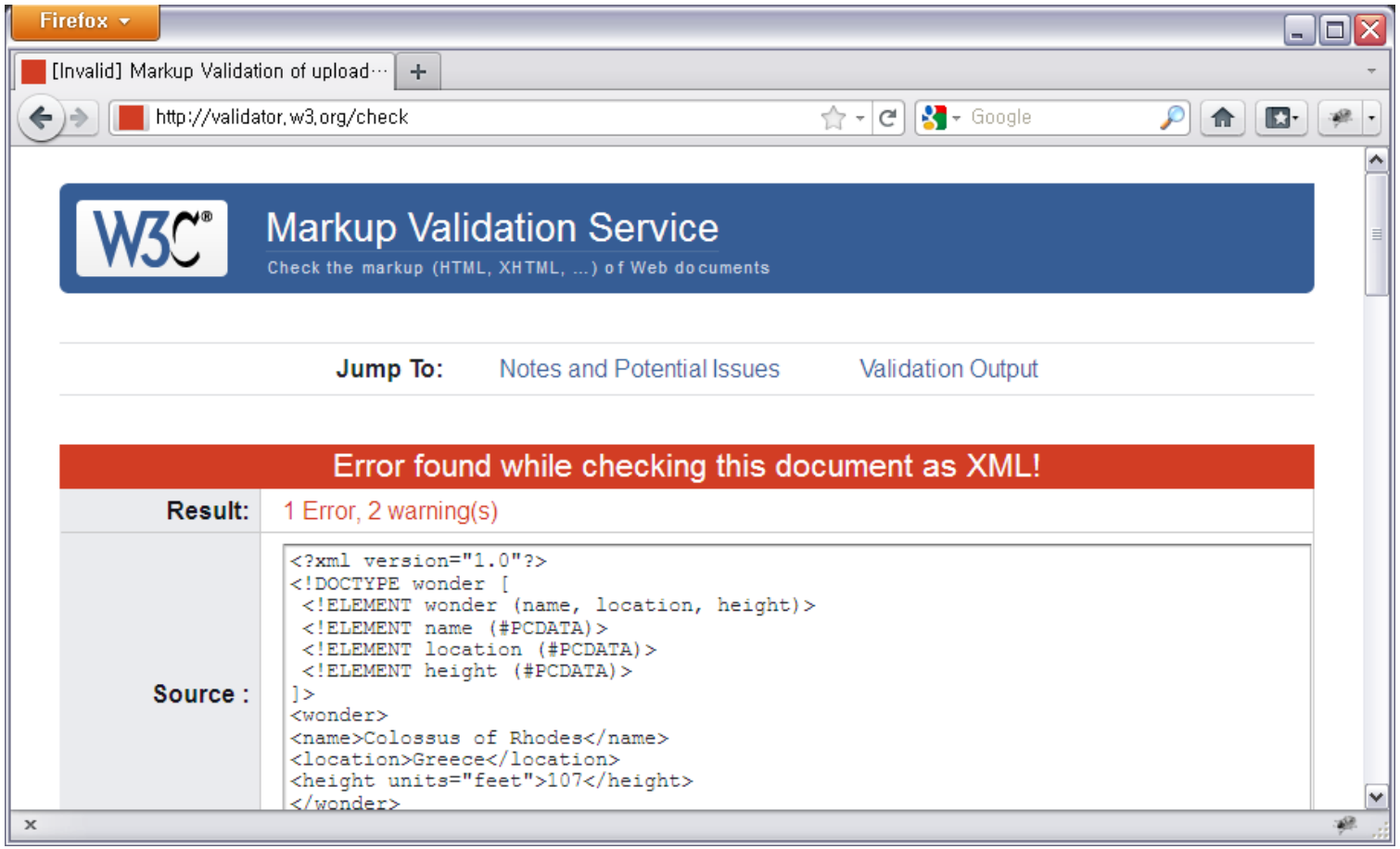
Validating XML Documents Against a DTD

- http://validator.w3.org/



Validating XML Documents Against a DTD

- http://validator.w3.org/



Validating XML Documents Against a DTD

- http://validator.w3.org/

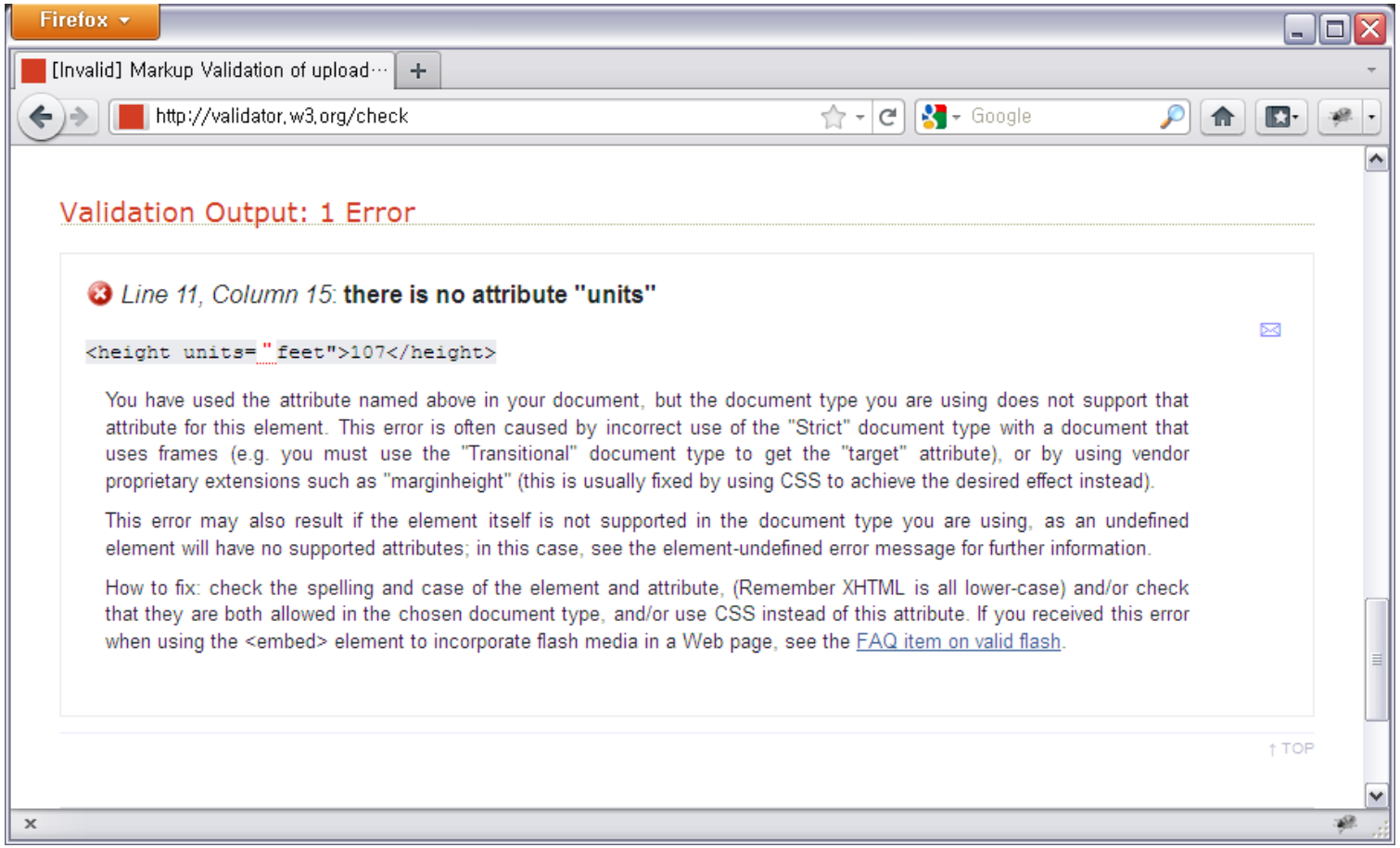


Figure 8.7

Naming a Public External DTD

- If your DTD will be used by others, you should name your DTDs in **STANDARD WAY** Using a ***formal public identifier*** (FPI)
- An XML parser could use the **FPI** to find **the latest version of the DTD** on a public Web server

-: if your DTD is **not a standard** (this is most common)
+: if your DTD is **an approved non-ISO standard**
ISO: if your DTD is **an approved ISO standard**

Language

```
-//kehogo//DTD WOWML 2.0//EN
```

Figure 8.8

Owner who wrote and will maintain the DTD

DTD Description (should contain a unique element)

Declaring a Public External DTD

- To declare a public external DTD:

Corresponds to **the name of the root element** in this XML document

Formal public identifier

```
<?xml version="1.0"?>
<!DOCTYPE ancient_wonders PUBLIC
"-//kehogo//DTD WowML 2.0//EN"
"http://www.kehogo.com/dtd/wonders-master.dtd">
<ancient_wonders>
 <wonder>
  <name language="English">
   Colossus of Rhodes
  </name>
...
 </wonder>
</ancient_wonders>
```

The location **URI** of the DTD content file

Figure 8.9

Pros and Cons of DTDs

- Some pros of using DTDs
  - They are compact and easily comprehended with a little direction
  - They can be defined inline (internal DTDs) for quick development
  - They can define entities (see Ch. 7)
  - They are the most widely accepted and are supported by most XML parsers

- Some cons of using DTDs
  - They are not written using XML syntax, and require parsers to support an additional language
  - They do not support Namespaces (see Ch. 12 and 13)
  - They do not have data type, thereby decreasing the strength of the validation
  - They have limited capacity to define how many child elements can nest within a given parent element