



SNS COLLEGE OF TECHNOLOGY

**Coimbatore-35
An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



DEPARTMENT OF INFORMATION TECHNOLOGY

WEB TECHNOLOGY

III YEAR - V SEM

UNIT 1 – Web Site Basics And HTML

TOPIC 6 - Markup Languages XHTML.

An Introduction to HTML History-Versions



HTML “Hello World!”



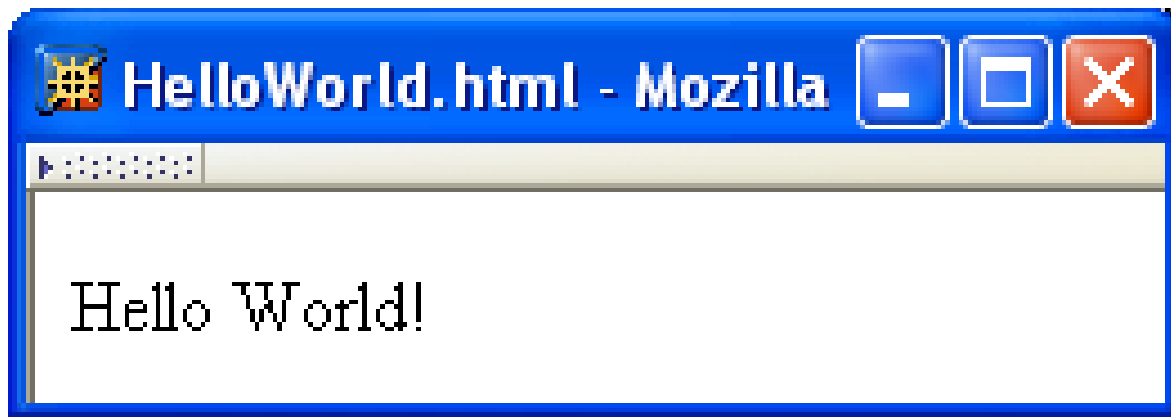
Document
Type
Declaration

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>
      HelloWorld.html
    </title>
  </head>
  <body>
    <p>
      Hello World!
    </p>
  </body>
</html>
```

Document
Instance



HTML “Hello World”





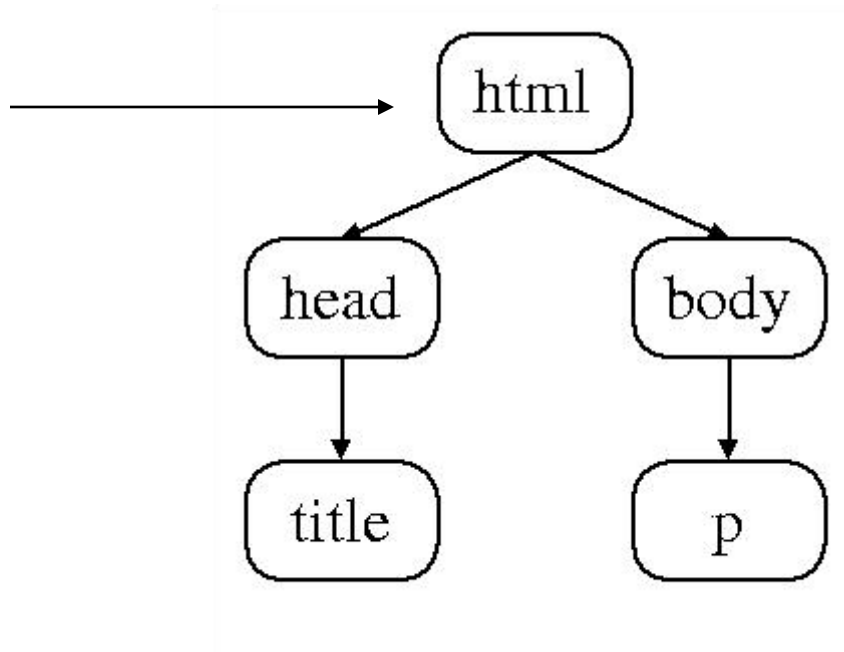
HTML Tags and Elements

- ▶ Any string of the form `< ... >` is a *tag*
- ▶ All tags in document instance of Hello World are either **end tags** (begin with `</`) or **start tags** (all others)
 - ▶ Tags are an example of **markup**, that is, text treated specially by the browser
 - ▶ Non-markup text is called **character data** and is normally displayed by the browser
- ▶ String at beginning of start/end tag is an **element name**
- ▶ Everything from start tag to matching end tag, including tags, is an **element**
 - ▶ **Content** of element excludes its start and end tags



HTML Element Tree

Root
Element





HTML Root Element

- ▶ Document type declaration specifies name of root element:
`<!DOCTYPE html`
- ▶ Root of HTML document must be `html`
- ▶ XHTML 1.0 (standard we will follow) requires that this element contain the xml namespace `xmlns` **attribute specification** (name/value pair)

```
<html xmlns="http://www.w3.org/1999/xhtml" >
```



HTML head and body Elements



- ▶ The **body** element contains information displayed in the browser client area
- ▶ The **head** element contains information used for other purposes by the browser:
 - ▶ title (shown in title bar of browser window)
 - ▶ scripts (client-side programs)
 - ▶ style (display) information
 - ▶ etc.



HTML History



- ▶ 1990: HTML invented by Tim Berners-Lee
- ▶ 1993: Mosaic browser adds support for images, sound, video to HTML
- ▶ 1994-~1997: “**Browser wars**” between Netscape and Microsoft, HTML defined operationally by browser support
- ▶ ~1997-present: Increasingly, World-Wide Web Consortium ([W3C](http://www.w3.org)) recommendations define HTML

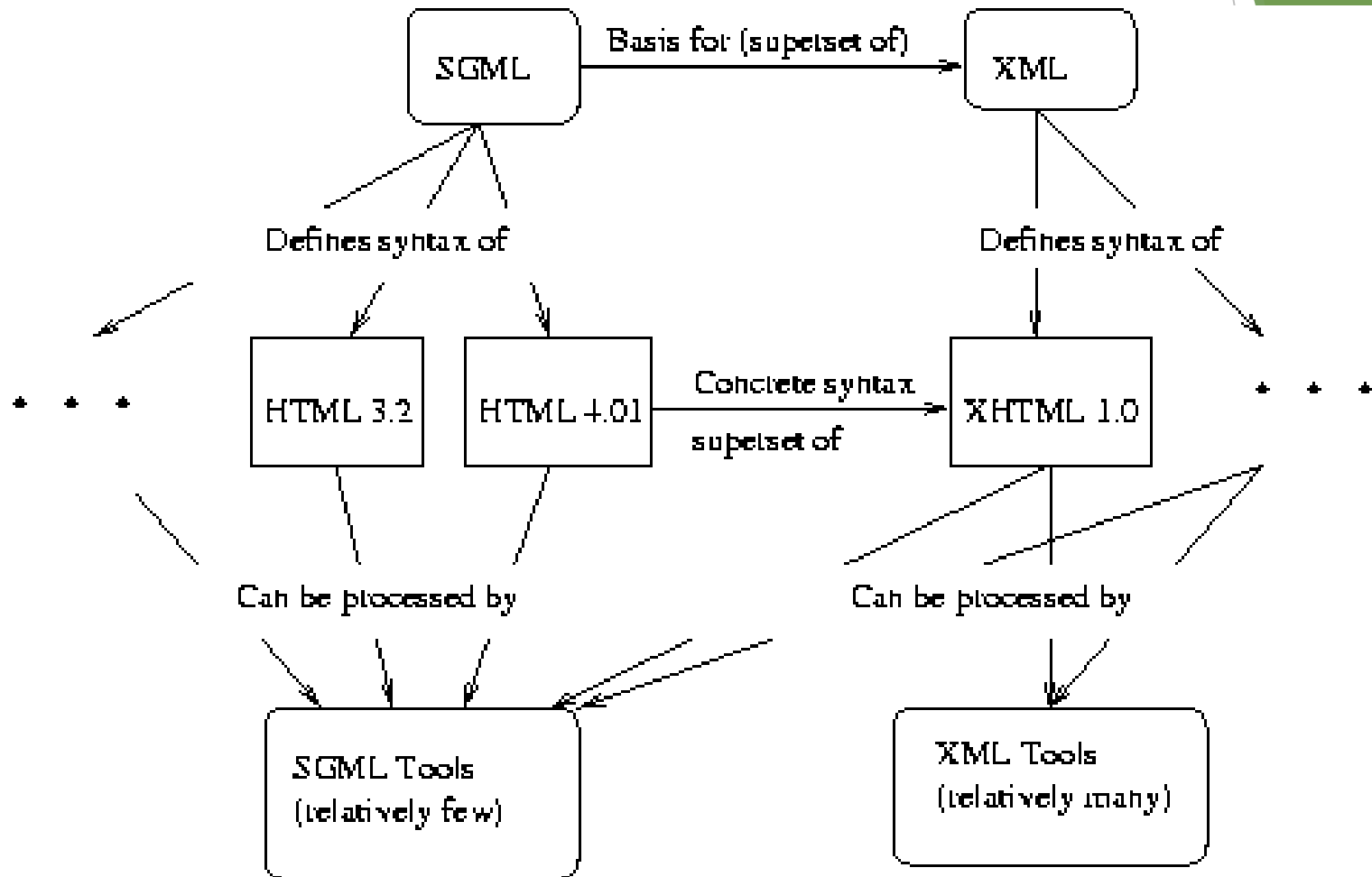


HTML Versions

- ▶ HTML 4.01 (Dec 1999) syntax defined using **Standard Generalized Markup Language (SGML)**
- ▶ XHTML 1.0 (Jan 2000) syntax defined using **Extensible Markup Language (XML)**
- ▶ Primary differences:
 - ▶ HTML allows some **tag omissions** (e.g., end tags)
 - ▶ XHTML element and attribute names are **lower case** (HTML names are case-insensitive)
 - ▶ XHTML requires that attribute **values** be **quoted**



SGML and XML





HTML “Flavors”

- ▶ For HTML 4.01 and XHTML 1.0, the document type declaration can be used to select one of three “flavors”:
 - ▶ **Strict**: W3C ideal
 - ▶ **Transitional**: Includes deprecated elements and attributes (W3C recommends use of *style sheets* instead)
 - ▶ **Frameset**: Supports frames (subwindows within the client area)



HTML Frameset

The screenshot shows a Mozilla browser window titled "Applet (Java 2 Platform SE v1.4.2) - Mozilla". The page content includes a navigation menu with "Overview", "Package", "Class", "Use", "Tree", "Deprecated", "Index", and "Help". The "Class" tab is selected, showing the "Class Applet" under the "java.applet" package. A class hierarchy tree is displayed, starting from "java.lang.Object" and descending through "java.awt.Component", "java.awt.Container", and "java.awt.Panel" to "java.applet.Applet". Below the hierarchy, it lists "All Implemented Interfaces: Accessible, ImageObserver, MenuContainer, Serializable".

Screen shots are reproduced by permission of Sun Microsystems Inc. All rights reserved.



HTML Document Type Declarations

- ▶ XHTML 1.0 Strict:

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```
- ▶ XHTML 1.0 Frameset:

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```
- ▶ HTML 4.01 Transitional:

```
<!DOCTYPE HTML  
PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```



XHTML White Space

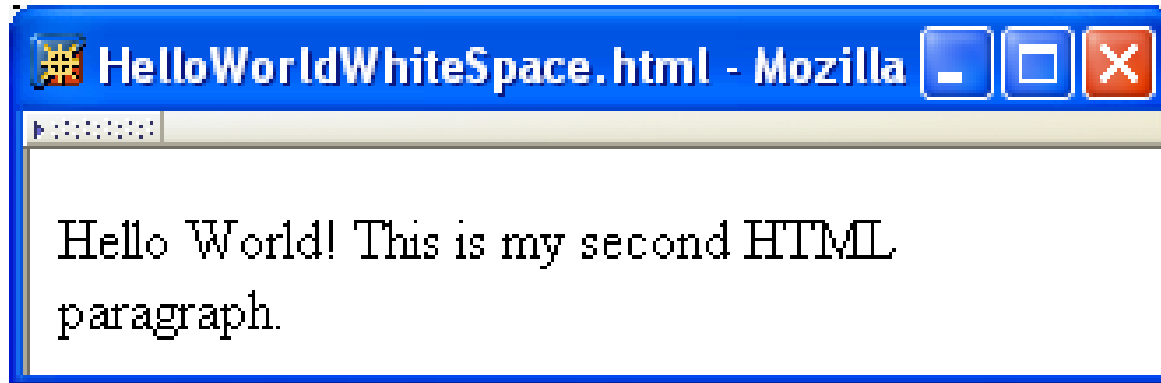


- ▶ Four white space characters: carriage return, line feed, space, horizontal tab
- ▶ Normally, character data is **normalized**:
 - ▶ All white space is converted to space characters
 - ▶ Leading and trailing spaces are trimmed
 - ▶ Multiple consecutive space characters are replaced by a single space character



XHTML White Space

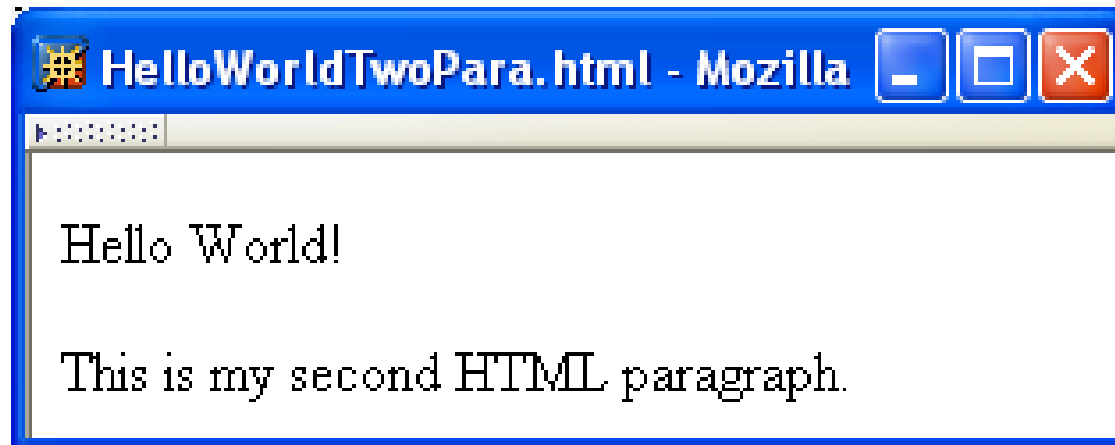
```
<body>  
  <p>  
    Hello World!  
  
    This is my second HTML paragraph.  
  </p>  
</body>
```





XHTML White Space

```
<p>  
    Hello World!  
</p>  
<p>  
    This is my second HTML paragraph.  
</p>
```





Unrecognized HTML Elements



```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <titl>
      HelloWorldBadElt.html
    </title>
  </head>
  <body>
    <p>
      Hello World!
    </p>
  </body>
</html>
```

Misspelled
element name →

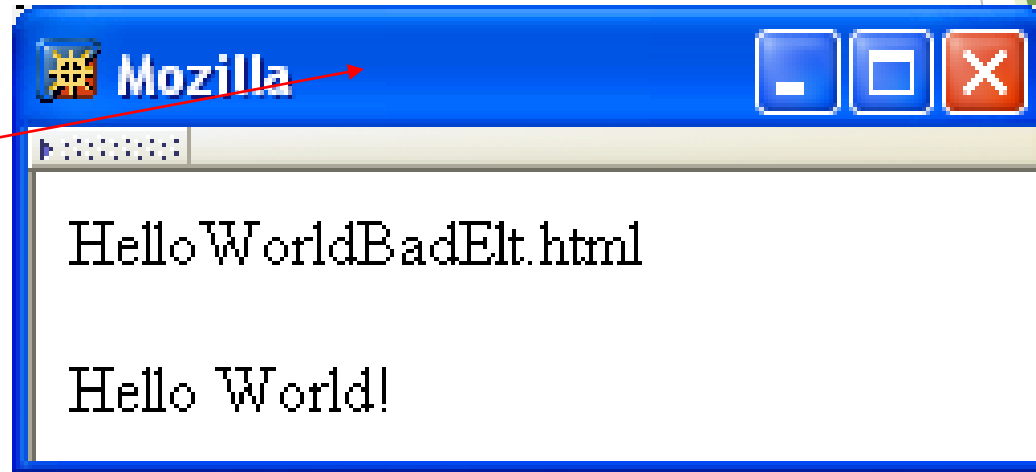


Unrecognized HTML Elements



Belongs here

title character data





Unrecognized HTML Elements



title character
data

Displayed
here





Unrecognized HTML Elements



- ▶ Browsers ignore tags with unrecognized element names, attribute specifications with unrecognized attribute names
 - ▶ Allows evolution of HTML while older browsers are still in use
- ▶ Implication: an HTML document may have errors even if it displays properly
- ▶ Should use an [HTML validator](#) to check syntax



Unrecognized HTML Elements



Example for non-frame browsers (old)

```
<HTML>
  <HEAD>
    <TITLE>A simple frameset document</TITLE>
  </HEAD>
  <FRAMESET cols="20%, 80%">
    <FRAME src="contents_of_frame1.html" />
    <FRAME src="contents_of_frame2.html" />
    <NOFRAMES>
      <P>This doc contains frames</P>
    </NOFRAMES>
  </FRAMESET>
</HTML>
```



HTML References

- ▶ Since < marks the beginning of a tag, how do you include a < in an HTML document?
- ▶ Use markup known as a **reference**
- ▶ Two types:
 - ▶ **Character reference** specifies a character by its Unicode code point
 - ▶ For <, use `<`; or `<`; or `<`;
 - ▶ **Entity reference** specifies a character by an HTML-defined name
 - ▶ For <, use `<`;



HTML References

TABLE 2.2: Example entity and character references.

Character	Entity Reference	Character Reference (decimal)
<	<	<
>	>	>
&	&	&
"	"	"
'	'	'
©	©	©
ñ	ñ	ñ
α	α	α
∀	∀	∀



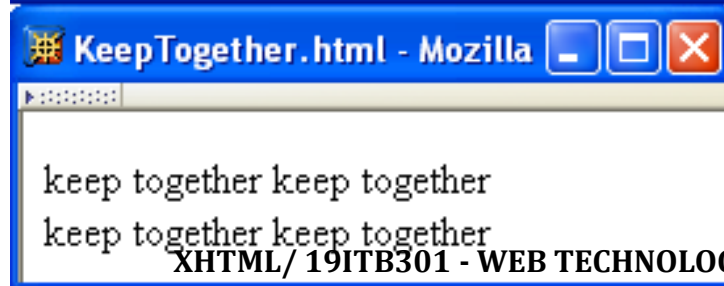
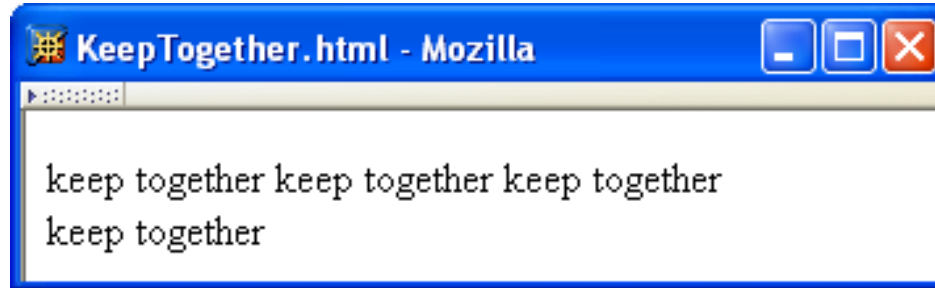
HTML References

- ▶ Since `<` and `&` begin markup, within character data or attribute values these characters must *always* be represented by references (normally `<` and `&`;))
- ▶ Good idea to represent `>` using reference (normally `>`;))
 - ▶ Provides consistency with treatment of `<`
 - ▶ Avoids accidental use of the reserved string `]]>`



HTML References

- ▶ **Non-breaking space ()** produces space but counts as part of a word
 - ▶ Ex: keep together keep together ...





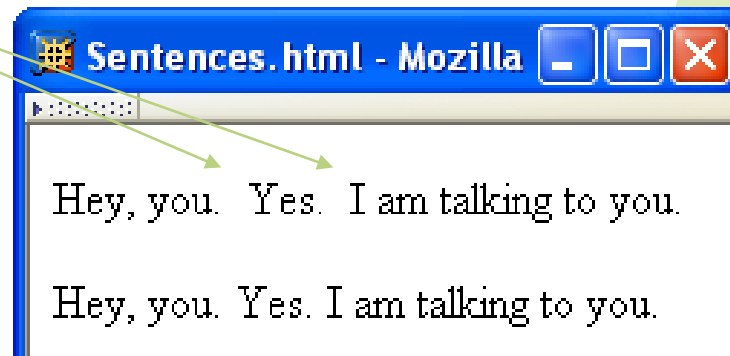
HTML References

- ▶ Non-breaking space often used to create multiple spaces (not removed by normalization)

```
<p>  
  Hey, you.&nbsp; Yes.&nbsp; I am talking to you.  
</p>
```

```
<p>  
  Hey, you.  Yes.  I am talking to you.  
</p>
```

 + space
displays as two
spaces





XHTML Attribute Specifications

▶ Example:

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
```

- ▶ Valid attribute names specified by HTML recommendation (or XML, as in xml:lang)
- ▶ Attribute values must be quoted (matching single or double quotes)
- ▶ Multiple attribute specifications are space-separated, order-independent



XHTML Attribute Values

- ▶ Can contain embedded quotes or references to quotes
- ▶ May be ~~value~~ value = "Ain't this grand!"
 - ▶ Best to ~~value~~ value = "He said, "She said"; then sighed."
browse ~~value~~ value = "He said, "She said", then sighed."