



# **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 8 - Relative URLs-Lists-tables-Frames- Forms-HTML 5.0**



# Relative URL's

- ▶ Consider an `<img>` start tag containing attribute specification

- ▶ This is an example of a `src="valid-xhtml10.png"` **relative URL**: it is interpreted relative to the URL of the document that contains the `img` tag

- ▶ If document URL is <http://localhost:8080/MultiFile.html> then relative URL above represents **absolute URL** <http://localhost:8080/valid-xhtml10.png>



# Relative URL's

TABLE 2.4: Absolute URL's corresponding to relative URL's when the base URL is `http://www.example.org/a/b/c.html`.

Relative URL	Absolute URL
<code>d/e.html</code>	<code>http://www.example.org/a/b/d/e.html</code>
<code>../f.html</code>	<code>http://www.example.org/a/f.html</code>
<code>../../g.html</code>	<code>http://www.example.org/g.html</code>
<code>../h/i.html</code>	<code>http://www.example.org/a/h/i.html</code>
<code>/j.html</code>	<code>http://www.example.org/j.html</code>
<code>/k/l.html</code>	<code>http://www.example.org/k/l.html</code>



# Relative URL's

- ▶ Query and fragment portions of a relative URL are appended to the resulting absolute URL
  - ▶ Example: If document URL is <http://localhost:8080/PageAnch.html> and it contains the anchor element

then the corresponding absolute URL is <http://localhost:8080/PageAnch.html#section1>

```
<a href="#section1">...
```



# Relative URL's



- ▶ Advantages:
  - ▶ Shorter than absolute URL's
  - ▶ Primary: can **change the URL of a document** (e.g., move document to a different directory or rename the server host) without needing to change URL's within the document
- ▶ Should use relative URL's whenever possible



# Lists

The screenshot shows a browser window titled "Lists.html - Mozilla...". The content area displays the following HTML lists:

- Bulleted list item
- Bulleted list item 2

1. Numbered list item
2. Numbered list item 2

Term  
    Definition of term

Term 2  
    Definition of term 2



# Lists

Unordered List

```
<ul>
  <li>Bulleted list item</li>
  <li>Bulleted list item 2</li>
</ul>
```

Ordered List

```
<ol>
  <li>Numbered list item</li>
  <li>Numbered list item 2</li>
</ol>
```

Definition List

```
<dl>
  <dt>Term</dt>
  <dd>Definition of term</dd>
  <dt>Term 2</dt>
  <dd>Definition of term 2</dd>
</dl>
```

List Items



# Lists



```
<ul>  
  <li>Bulleted list item  
    <ul>  
      <li>Nested list item</li>  
      <li>Nested list item 2</li>  
    </ul>  
  </li>  
  <li>Bulleted list item 2</li>  
</ul>
```





# Tables

Kim	100	89
Sandy	78	92
Taylor	83	73

Rules

Borders

Rules



# Tables

Border 5 pixels, rules 1 pixel

```
<table border="5">  
  <tr>  
    <td>Kim</td><td>100</td><td>89</td>  
  </tr>  
  <tr>  
    <td>Sandy</td><td>78</td><td>92</td>  
  </tr>  
  <tr>  
    <td>Taylor</td><td>83</td><td>73</td>  
  </tr>  
</table>
```

Table Row

Table Data



# Tables

GradeTableHdr.html - Mozilla

COSC 400 Student Grades

		Grades	
	Student	Exam 1	Exam 2
Undergraduates	Kim	100	89
	Sandy	78	92
Graduates	Taylor	83	73



# Tables

```
<table border="5">
  <caption>
    COSC 400 Student Grades
  </caption>
  <tr>
    <td>&nbsp;</td><td>&nbsp;</td><th colspan="2">Grades</th>
  </tr>
  <tr>
    <td>&nbsp;</td><th>Student</th><th>Exam 1</th><th>Exam 2</th>
  </tr>
  <tr>
    <th rowspan="2">Undergraduates</th><td>Kim</td><td>100</td><td>89</td>
  </tr>
  <tr>
    <td>Sandy</td><td>78</td><td>92</td>
  </tr>
  <tr>
    <th>Graduates</th><td>Taylor</td><td>83</td><td>73</td>
  </tr>
</table>
```

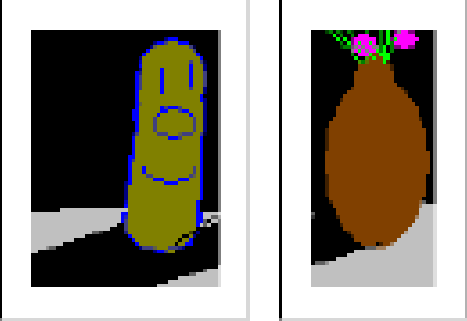
Table Header





# Tables

TableSpacePad.html - Mozilla

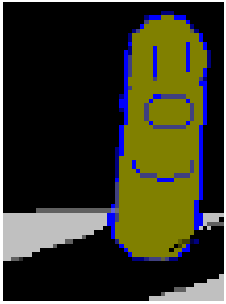
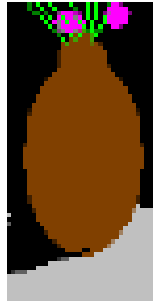
cellspacing	cellpadding	Example
10	10	

```
<table border="1" cellspacing="10" cellpadding="10">
```



# Tables

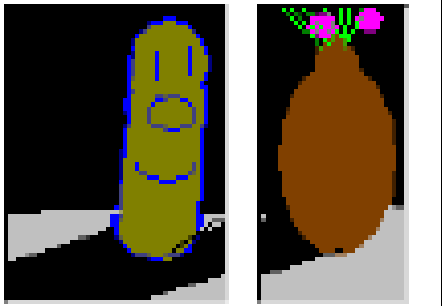
cellspacing cellpadding

0	10		
---	----	---	--



# Tables

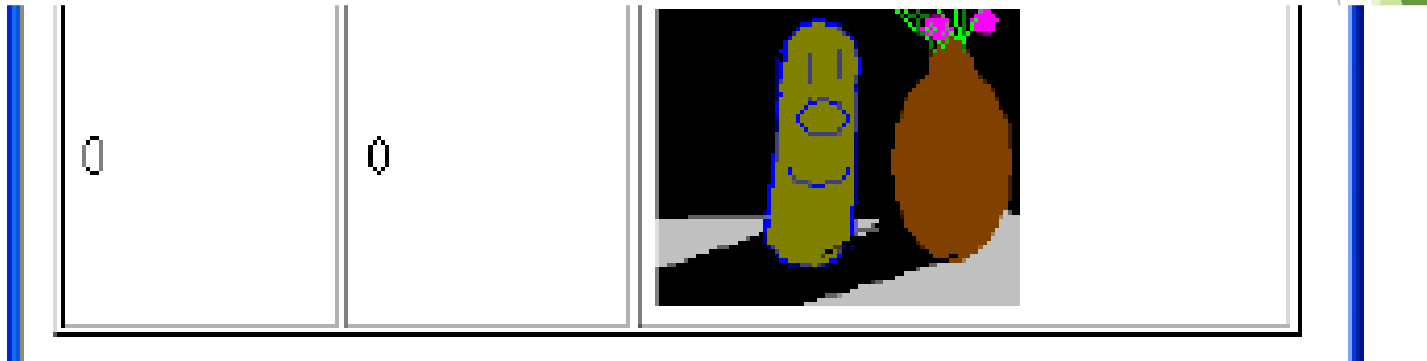
cellspacing cellpadding

10	0	
----	---	---



# Tables

cellspacing cellpadding







# Frames



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. Below the navigation, there are links for "PREV CLASS", "NEXT CLASS", "FRAMES", and "NO FRAMES". The main content area displays the class hierarchy for `java.applet.Applet`, showing its inheritance from `java.lang.Object`, `java.awt.Component`, `java.awt.Container`, and `java.awt.Panel`. Below the hierarchy, it lists "All Implemented Interfaces" as `Accessible`, `ImageObserver`, `MenuContainer`, and `Serializable`. The left sidebar shows the "All Classes" section with "Applet" selected.

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



# Frames

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Java 2 Platform SE v1.4.2</title>
  </head>
  <frameset cols="20%,80%">
    <frameset rows="1*,2*">
      <frame src="overview-frame.html"
        id="upperLeftFrame" name="upperLeftFrame"></frame>
      <frame src="allclasses-frame.html"
        id="lowerLeftFrame" name="lowerLeftFrame"></frame>
    </frameset>
    <frame src="overview-summary.html"
      id="rightFrame" name="rightFrame"></frame>
  </frameset>
</html>
```

1/3,2/3 split



# Frames

- ▶ Hyperlink in one frame can load document in another:
- ▶ Value of target attribute specification is id/name of a frame

```
<a href="java/applet/package-frame.html" target="lowerLeftFrame">
```



# Frames

- ▶ User interface issues:
  - ▶ What happens when the page is **printed**?
  - ▶ What happens when the **Back button** is clicked?
  - ▶ How should **assistive technology** “read” the page?
  - ▶ How should the information be displayed on a **small display**?
- ▶ Recommendation: avoid frames except for applications aimed at “power users”



# Forms



LifeStory.html - Mozilla

Enter your name:

Give your life's story in 100 words or less:

Check all that apply to you:  tall  funny  smart



# Forms



Each form is content of a form element

```
<form action="http://www.example.org" method="get">
  <div>
    <label>
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
  </div>
</form>
```



# Forms



action specifies URL where form data is sent in an HTTP request

```
<form action="http://www.example.org" method="get">
  <div>
    <label>
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
  </div>
</form>
```



# Forms

HTTP request method (lower case)

```
<form action="http://www.example.org" method="get" >
  <div>
    <label>
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
  </div>
</form>
```





# Forms



- ▶ The XHTML grammar require any child of the *form* element to be a block
- ▶ Many form elements are actually *inline*, so including a block element on top such a *div* or a table is a simple way to be compliant with the grammar



# Forms



```
<form action="http://www.example.org" method="get">  
<div> div is the block element analog of span (no-style block element)  
<label>  
  Enter your name: <input type="text" name="username" size="40" />  
</label>  
<br />  
<label>  
  Give your life's story in 100 words or less:  
<br />  
  <textarea name="lifestory" rows="5" cols="60"></textarea>  
</label>  
<br />
```



# Forms



```
<form action="http://www.example.org" method="get">
```

**<div>** Form control elements must be content of a block element

```
<label>
```

```
  Enter your name: <input type="text" name="username" size="40" />
```

```
</label>
```

```
<br />
```

```
<label>
```

```
  Give your life's story in 100 words or less:
```

```
<br />
```

```
  <textarea name="lifestory" rows="5" cols="60"></textarea>
```

```
</label>
```

```
<br />
```



# Forms



```
<form action="http://www.example.org" method="get">
  <div>
    <label>
      Text field control (form user-interface element)
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
  </div>
</form>
```



# Forms



```
<form action="http://www.example.org" method="get">
  <div>
    <label>
      Text field used for one-line inputs
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
  </div>
</form>
```



# Forms



LifeStory.html - Mozilla

Enter your name:

Give your life's story in 100 words or less:

Check all that apply to you:  tall  funny  smart



# Forms

```
<form action="http://www.example.org" method="get">
  <div>
    <label> Name associated with this control's data in HTTP request
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
  </div>
</form>
```



# Forms

```
<form action="http://www.example.org" method="get">
  <div>
    <label>
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
  </div>
</form>
```

Width (number of characters) of text field

size="40"





# Forms



```
<form action="http://www.example.org" method="get">
```

```
<div>
```

```
<label>
```

```
Enter your name: <input type="text" name="username" size="40" />
```

```
</label>
```

```
<br />
```

```
<label>
```

```
Give your life's story in 100 words or less:
```

```
<br />
```

```
<textarea name="lifestory" rows="5" cols="60"></textarea>
```

```
</label>
```

```
<br />
```

input is an empty element



# Forms

```
<form action="http://www.example.org" method="get">
  <div>
    <label> Use label to associate text with a control
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
  </div>
</form>
```

Only one control inside a label element!



# Forms

```
<form action="http://www.example.org" method="get">
  <div>
    <label>
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br /> Form controls are inline elements
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
```



# Forms

```
<form action="http://www.example.org" method="get">
  <div>
    <label>
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br /> textarea control used for multi-line input
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
  </div>
</form>
```



# Forms

```
<form action="http://www.example.org" method="get">
  <div>
    <label>
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
```

Height and width in characters



# Forms

```
<form action="http://www.example.org" method="get">
  <div>
    <label>
      Enter your name: <input type="text" name="username" size="40" />
    </label>
    <br />
    <label>
      Give your life's story in 100 words or less:
      <br />
      <textarea name="lifestory" rows="5" cols="60"></textarea>
    </label>
    <br />
```

textarea is not an empty element; any content is displayed



# Forms



LifeStory.html - Mozilla

Enter your name:

Give your life's story in 100 words or less:

Check all that apply to you:  tall  funny  smart



# Forms

Check all that apply to you:

<label> Checkbox control

```
<input type="checkbox" name="boxgroup1" value="tall" />tall
```

</label>

<label>

```
<input type="checkbox" name="boxgroup1" value="funny" />funny
```

</label>

<label>

```
<input type="checkbox" name="boxgroup1" value="smart" />smart
```

</label>

<br /><br />

```
<input type="submit" name="doit" value="Publish My Life's Story" />
```

</div>

</form>





# Forms

Check all that apply to you:

Value sent in HTTP request if box is checked

```
<label>
  <input type="checkbox" name="boxgroup1" value="tall" />tall
</label>
<label>
  <input type="checkbox" name="boxgroup1" value="funny" />funny
</label>
<label>
  <input type="checkbox" name="boxgroup1" value="smart" />smart
</label>
<br /><br />
<input type="submit" name="doit" value="Publish My Life's Story" />
</div>
</form>
```



# Forms

Controls can share a common name

Check all that apply to you:

```
<label>
  <input type="checkbox" name="boxgroup1" value="tall" />tall
</label>
<label>
  <input type="checkbox" name="boxgroup1" value="funny" />funny
</label>
<label>
  <input type="checkbox" name="boxgroup1" value="smart" />smart
</label>
<br /><br />
<input type="submit" name="doit" value="Publish My Life's Story" />
</div>
</form>
```



# Forms

Check all that apply to you:

```
<label>
```

```
  <input type="checkbox" name="boxgroup1" value="tall" />tall
```

```
</label>
```

```
<label>
```

```
  <input type="checkbox" name="boxgroup1" value="funny" />funny
```

```
</label>
```

```
<label>
```

```
  <input type="checkbox" name="boxgroup1" value="smart" />smart
```

```
</label>
```

```
<br /><br />
```

```
<input type="submit" name="doit" value="Publish My Life's Story" />
```

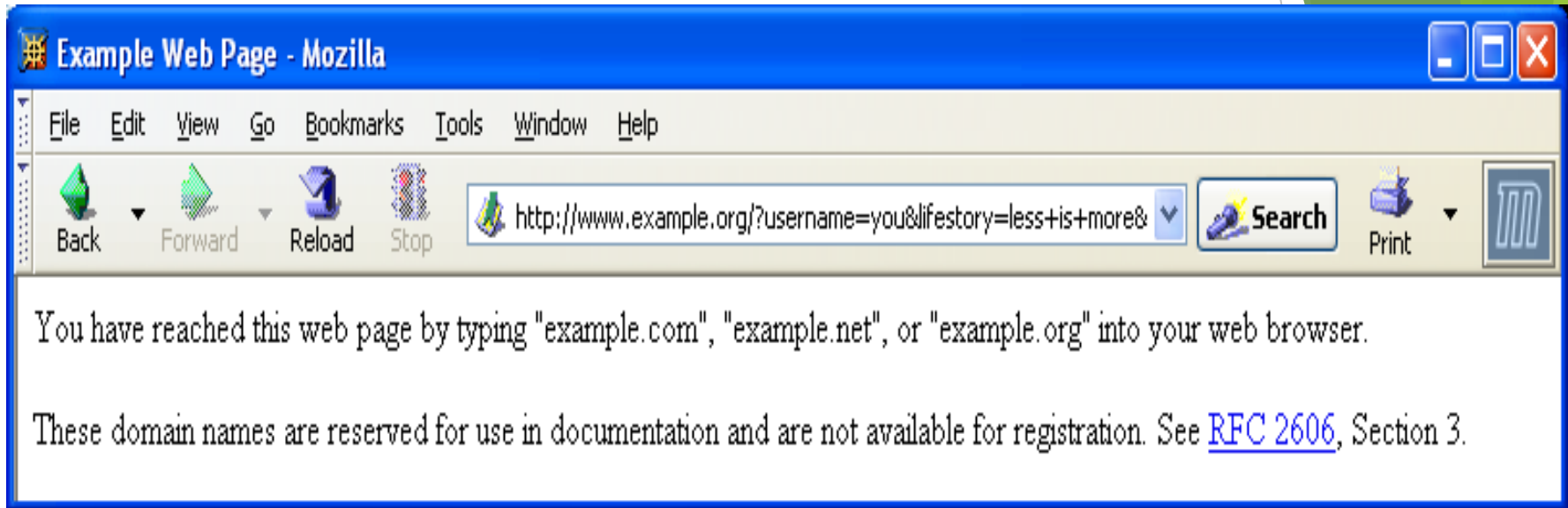
```
</div>
```

Submit button: form data sent to action URL if button is clicked

```
</form>
```



# Forms





# Forms



The screenshot shows a Mozilla browser window titled "Example Web Page - Mozilla". The address bar contains the URL `http://www.example.org/?username=you&lifestory=less+is+more&`. The text "Form data (in GET request)" is overlaid in green above the address bar, and the query string portion of the URL is circled in green. The browser's menu bar includes File, Edit, View, Go, Bookmarks, Tools, Window, and Help. Below the address bar are navigation buttons for Back, Forward, Reload, and Stop, along with Search and Print buttons. The main content area of the browser displays the following text:

You have reached this web page by typing "example.com", "example.net", or "example.org" into your web browser.

These domain names are reserved for use in documentation and are not available for registration. See [RFC 2606](#), Section 3.



# Forms

Check all that apply to you:

```
<label>
```

```
  <input type="checkbox" name="boxgroup1" value="tall" />tall
```

```
</label>
```

```
<label>
```

```
  <input type="checkbox" name="boxgroup1" value="funny" />funny
```

```
</label>
```

```
<label>
```

```
  <input type="checkbox" name="boxgroup1" value="smart" />smart
```

```
</label>
```

```
<br /><br />
```

```
<input type="submit" name="doit" value="Publish My Life's Story" />
```

```
</div>
```

```
</form>
```

Displayed on button and sent to server if button clicked



# Forms

Radio buttons: at most one can be selected at a time.



IncomeRadio.html - Mozilla

Your annual income is (select one):

- Less than \$10,000
- Between \$10,000 and \$50,000
- Over \$50,000



# Forms

Your annual income is (select one):<br />

<label> Radio button control

<input type="radio" name="radgroup1" value="0-10" />

Less than \$10,000

</label><br />

<label>

<input type="radio" name="radgroup1" value="10-50"  
checked="checked" />

Between \$10,000 and \$50,000

</label><br />

<label>

<input type="radio" name="radgroup1" value=">50" />

Over \$50,000

</label>





# Forms

Your annual income is (select one):<br />

<label>

```
<input type="radio" name="radgroup1" value="0-10" />
```

Less than \$10,000

</label><br />

<label>

```
<input type="radio" name="radgroup1" value="10-50"
      checked="checked" />
```

Between \$10,000 and \$50,000

</label><br />

<label>

```
<input type="radio" name="radgroup1" value=">50" />
```

Over \$50,000

</label>

All radio buttons with the same name form a *button set*



# Forms

Your annual income is (select one):<br />

<label>

```
<input type="radio" name="radgroup1" value="0-10" />
```

Less than \$10,000

</label><br />

<label>

```
<input type="radio" name="radgroup1" value="10-50"
checked="checked" />
```

Between \$10,000 and \$50,000

</label><br />

<label>

```
<input type="radio" name="radgroup1" value=">50" />
```

Over \$50,000

</label>

Only one button of a set can be selected at a time



# Forms

Your annual income is (select one):<br />

<label>

```
<input type="radio" name="radgroup1" value="0-10" />
```

Less than \$10,000

</label><br />

<label>

```
<input type="radio" name="radgroup1" value="10-50"
```

```
checked="checked" /> This button is initially selected
```

Between \$10,000 and \$50,000 (checked attribute also applies  
to check boxes)

</label><br />

<label>

```
<input type="radio" name="radgroup1" value=">50" />
```

Over \$50,000

</label>



# Forms

Your annual income is (select one):<br />

<label>

```
<input type="radio" name="radgroup1" value="0-10" />
```

Less than \$10,000

</label><br />

<label>

```
<input type="radio" name="radgroup1" value="10-50"
```

**checked="checked" />** *Boolean attribute: default false,*

Between \$10,000 and \$50,000 *set true by specifying name as value*

</label><br />

<label>

```
<input type="radio" name="radgroup1" value=">50" />
```

Over \$50,000

</label>



# Forms

```
Your annual income is (select one):<br />
<label>
  <input type="radio" name="radgroup1" value="0-10" />
  Less than $10,000
</label><br />
<label>
  <input type="radio" name="radgroup1" value="10-50"
    checked="checked" />
  Between $10,000 and $50,000
</label><br />
<label>
  <input type="radio" name="radgroup1" value=">50" />
  Over $50,000
</label>
```

Represents string: >50



# Forms

IncomeSelect.html - Mozilla

Your annual income is (select one):

Between \$10,000 and \$50,000	▼
Less than \$10,000	
Between \$10,000 and \$50,000	
Over \$50,000	

} Menu



# Forms



Your annual income is (select one):

```
<select name="income"> Menu control; name given once
  <option value="0-10">Less than $10,000</option>
  <option value="10-50" selected="selected">
    Between $10,000 and $50,000
  </option>
  <option value=">50">Over $50,000</option>
</select>
```



# Forms



Your annual income is (select one):

```
<select name="income">Each menu item has its own value
  <option value="0-10">Less than $10,000</option>
  <option value="10-50" selected="selected">
    Between $10,000 and $50,000
  </option>
  <option value=">50">Over $50,000</option>
</select>
```





# Forms



Your annual income is (select one):

```
<select name="income">
```

```
<option value="0-10">Less than $10,000</option>
```

```
<option value="10-50" selected="selected">
```

```
  Between $10,000 and $50,000
```

Item initially displayed in menu control

```
</option>
```

```
<option value=">50">Over $50,000</option>
```

```
</select>
```



# Forms

- ▶ Other form controls:
  - ▶ Fieldset (grouping)
  - ▶ Password
  - ▶ Clickable image
  - ▶ Non-submit buttons
  - ▶ Hidden (embed data)
  - ▶ File upload
  - ▶ Hierarchical menus

MoreControls.html - Mozilla

Example of a fieldset

input type=password:

input type=image:

input type=button:

input type=hidden:

input type=file:

button type=button:

Hierarchical menu

select with optgroup:    
Make a selection  
**Group1**  
1.1  
1.2  
**Group2**  
2.1  
2.2



# Forms

TABLE 2.5: HTML 4.01/XHTML 1.0 non-deprecated form controls.

Element	type Attribute	Control
input	text	Text input
input	password	Password input
input	checkbox	Checkbox
input	radio	Radio button
input	submit	Submit button
input	image	Graphical submit button
input	reset	Reset button (form clear)
input	button	Push button (for use with scripts)
input	hidden	Non-displayed control (stores server-supplied information)
input	file	File select
button	submit	Submit button with content (not an empty element)
button	reset	Cancel button with content (not an empty element)
button	button	Button with content but no predefined action
select	N/A	Menu
option	N/A	Menu item
optgroup	N/A	Heading in a hierarchical menu
textarea	N/A	Multi-line text input
label	N/A	Associate label with control(s)
fieldset	N/A	Groups controls
legend	N/A	Add caption to a fieldset



# XML DTD



- ▶ Recall that XML is used to define the syntax of XHTML
- ▶ Set of XML files that define a language are known as the **document type definition (DTD)**
- ▶ DTD primarily consists of **declarations**:
  - ▶ **Element type**: name and content of elements
  - ▶ **Attribute list**: attributes of an element
  - ▶ **Entity**: define meaning of, e.g., &gt;



# XML DTD

► Example from

<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>

```
<!ELEMENT html (head, body)>
```

```
<!ATTLIST html
```

```
  %i18n;
```

```
  id          ID          #IMPLIED
```

```
  xmlns      %URI;      #FIXED 'http://www.w3.org/1999/xhtml' >
```

```
<!ENTITY % i18n
```

```
  "lang      %LanguageCode;  #IMPLIED
```

```
  xml:lang  %LanguageCode;  #IMPLIED
```

```
  dir       (ltr|rtl)      #IMPLIED" >
```



# XML Element Type Declaration

```
<!ELEMENT html (head, body)>
```

Element type name



# XML Element Type Declaration

```
<!ELEMENT html (head, body)>
```

Element type *content specification* (or *content model*)



# XML Element Type Declaration

```
<!ELEMENT html (head, body)>
```

Element type *content specification (or content model)*

TABLE 2.6: Basic XML content specifications.

Specification Type	Syntax	Content Allowed
Empty	EMPTY	None
Arbitrary	ANY	Any content (no restrictions)
Sequence	(elt1, elt2, ...)	Sequence of elements that must appear in order specified
Choice	(elt1   elt2   ...)	Exactly one of the specified elements must appear
Character data	(#PCDATA)	Arbitrary character data, but no elements
Mixed	(#PCDATA   elt1   elt2  ... )*	Any mixture of character data and the specified elements in any order





# XML Element Type Declaration

```
<!ELEMENT html (head, body)>
```

Element type *content specification (or content model)*

TABLE 2.6: Basic XML content specifications.

Specification Type	Syntax	Content Allowed
Empty	EMPTY	None
Arbitrary	ANY	Any content (no restrictions)
Sequence	(elt1, elt2, ...)	Sequence of elements that must appear in order specified
Choice	(elt1   elt2   ...)	Exactly one of the specified elements must appear
Character data	(#PCDATA)	Arbitrary character data, but no elements
Mixed	(#PCDATA   elt1   elt2  ... )*	Any mixture of character data and the specified elements in any order



# XML Element Type Declaration

<!ELEMENT br **EMPTY**>

Element type *content specification (or content model)*

TABLE 2.6: Basic XML content specifications.

Specification Type	Syntax	Content Allowed
Empty	EMPTY	None
Arbitrary	ANY	Any content (no restrictions)
Sequence	(elt1, elt2, ...)	Sequence of elements that must appear in order specified
Choice	(elt1   elt2   ...)	Exactly one of the specified elements must appear
Character data	(#PCDATA)	Arbitrary character data, but no elements
Mixed	(#PCDATA   elt1   elt2  ... )*	Any mixture of character data and the specified elements in any order



# XML Element Type Declaration

`<!ELEMENT br EMPTY>`

Element type *content specification (or content model)*

TABLE 2.6: Basic XML content specifications.

Specification Type	Syntax	Content Allowed
Empty	EMPTY	None
Arbitrary	ANY	Any content (no restrictions)
Sequence	(elt1, elt2, ...)	Sequence of elements that must appear in order specified
Choice	(elt1   elt2   ...)	Exactly one of the specified elements must appear
Character data	(#PCDATA)	Arbitrary character data, but no elements
Mixed	(#PCDATA   elt1   elt2  ... )*	Any mixture of character data and the specified elements in any order



# XML Element Type Declaration

`<!ELEMENT select (optgroup|option)+>`

Element type *content specification (or content model)*

TABLE 2.6: Basic XML content specifications.

Specification Type	Syntax	Content Allowed
Empty	EMPTY	None
Arbitrary	ANY	Any content (no restrictions)
Sequence	(elt1, elt2, ...)	Sequence of elements that must appear in order specified
Choice	(elt1   elt2   ...)	Exactly one of the specified elements must appear
Character data	(#PCDATA)	Arbitrary character data, but no elements
Mixed	(#PCDATA   elt1   elt2  ... )*	Any mixture of character data and the specified elements in any order



# XML Element Type Declaration

`<!ELEMENT select (optgroup|option)+>`

Element type *content specification (or content model)*

TABLE 2.6: Basic XML content specifications.

Specification Type	Syntax	Content Allowed
Empty	EMPTY	None
Arbitrary	ANY	Any content (no restrictions)
Sequence	(elt1, elt2, ...)	Sequence of elements that must appear in order specified
Choice	(elt1   elt2   ...)	Exactly one of the specified elements must appear
Character data	(#PCDATA)	Arbitrary character data, but no elements
Mixed	(#PCDATA   elt1   elt2  ... )*	Any mixture of character data and the specified elements in any order



# XML Element Type Declaration

**<!ELEMENT textarea (#PCDATA)>**

Element type *content specification* (or *content model*)

TABLE 2.6: Basic XML content specifications.

Specification Type	Syntax	Content Allowed
Empty	EMPTY	None
Arbitrary	ANY	Any content (no restrictions)
Sequence	(elt1, elt2, ...)	Sequence of elements that must appear in order specified
Choice	(elt1   elt2   ...)	Exactly one of the specified elements must appear
Character data	(#PCDATA)	Arbitrary character data, but no elements
Mixed	(#PCDATA   elt1   elt2  ... )*	Any mixture of character data and the specified elements in any order





# XML Element Type Declaration

**<!ELEMENT textarea (#PCDATA)>**

Element type *content specification* (or *content model*)

TABLE 2.6: Basic XML content specifications.

Specification Type	Syntax	Content Allowed
Empty	EMPTY	None
Arbitrary	ANY	Any content (no restrictions)
Sequence	(elt1, elt2, ...)	Sequence of elements that must appear in order specified
Choice	(elt1   elt2   ...)	Exactly one of the specified elements must appear
Character data	(#PCDATA)	Arbitrary character data, but no elements
Mixed	(#PCDATA   elt1   elt2  ... )*	Any mixture of character data and the specified elements in any order



# XML Element Type Declaration

```
<!ELEMENT select (optgroup|option)+>
```

Element type *content specification* (or *content model*)





# XML Element Type Declaration

`<!ELEMENT select (optgroup|option)+>`

Element type *content specification* (or *content model*)

TABLE 2.7: XML content specification iterator characters.

Character	Meaning
?	Sequence/choice is optional (appears zero or one times)
*	Sequence/choice may be repeated an arbitrary number of times, including none
+	Sequence/choice may appear one or more times



# XML Element Type Declaration

`<!ELEMENT select (optgroup|option)+>`

Element type *content specification (or content model)*

TABLE 2.7: XML content specification iterator characters.

Character	Meaning
?	Sequence/choice is optional (appears zero or one times)
*	Sequence/choice may be repeated an arbitrary number of times, including none
+	Sequence/choice may appear one or more times



# XML Element Type Declaration

```
<!ELEMENT table  
  (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:



# XML Element Type Declaration

```
<!ELEMENT table  
  (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption



# XML Element Type Declaration



```
<!ELEMENT table  
  (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by



# XML Element Type Declaration



```
<!ELEMENT table  
  (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements



# XML Element Type Declaration

```
<!ELEMENT table  
  (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements or



# XML Element Type Declaration

```
<!ELEMENT table  
  (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements or any number of colgroup elements





# XML Element Type Declaration

```
<!ELEMENT table  
  (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements or any number of colgroup elements then



# XML Element Type Declaration

```
<!ELEMENT table  
    (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements or any number of colgroup elements then
  - ▶ **Optional header**



# XML Element Type Declaration

```
<!ELEMENT table  
  (caption?, (col*|colgroup*), tthead?, ttfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements or any number of colgroup elements then
  - ▶ Optional header followed by



# XML Element Type Declaration

```
<!ELEMENT table  
  (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements or any number of colgroup elements then
  - ▶ Optional header followed by optional footer



# XML Element Type Declaration

```
<!ELEMENT table  
  (caption?, (col*|colgroup*), thead?, tfoot? (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements or any number of colgroup elements then
  - ▶ Optional header followed by optional footer then



# XML Element Type Declaration

```
<!ELEMENT table  
  (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements or any number of colgroup elements then
  - ▶ Optional header followed by optional footer then
  - ▶ One or more tbody elements



# XML Element Type Declaration

```
<!ELEMENT table  
    (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements or any number of colgroup elements then
  - ▶ Optional header followed by optional footer then
  - ▶ One or more tbody elements or



# XML Element Type Declaration

```
<!ELEMENT table  
    (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+tr+))>
```

- ▶ Child elements of table are:
  - ▶ Optional caption followed by
  - ▶ Any number of col elements or any number of colgroup elements then
  - ▶ Optional header followed by optional footer then
  - ▶ One or more tbody elements or **one or more tr elements**





# XML Attribute List Declaration

Element type name

```
<!ATTLIST html  
  lang          NMTOKEN      #IMPLIED  
  xml:lang      NMTOKEN      #IMPLIED  
  dir           (ltr|rtl)     #IMPLIED  
  id            ID             #IMPLIED  
  xmlns         CDATA        #FIXED 'http://www.w3.org/1999/xhtml'>
```



# XML Attribute List Declaration



```
<!ATTLIST html
```

```
  lang          NMTOKEN      #IMPLIED  
  xml:lang     NMTOKEN      #IMPLIED  
  dir          (ltr|rtl)    #IMPLIED  
  id           ID           #IMPLIED  
  xmlns       CDATA        #FIXED 'http://www.w3.org/1999/xhtml'>
```

Recognized  
attribute names



# XML Attribute List Declaration



```
<!ATTLIST html
```

```
  lang
```

```
NMTOKEN
```

```
#IMPLIED
```

```
  xml:lang
```

```
NMTOKEN
```

```
#IMPLIED
```

```
  dir
```

```
(ltr|rtl)
```

```
#IMPLIED
```

```
  id
```

```
ID
```

```
#IMPLIED
```

```
  xmlns
```

```
CDATA
```

```
#FIXED 'http://www.w3.org/1999/xhtml'>
```

Attribute types

(data types allowed as attribute values)



# XML Attribute List Declaration



ASCII characters: letter, digit, or . - \_ :

```
<!ATTLIST html
```

```
  lang          NMTOKEN #IMPLIED
```

```
  xml:lang      NMTOKEN #IMPLIED
```

```
  dir           (ltr|rtl) #IMPLIED
```

```
  id            ID #IMPLIED
```

```
  xmlns         CDATA #FIXED 'http://www.w3.org/1999/xhtml'>
```



# XML Attribute List Declaration



```
<!ATTLIST html
  lang          NMTOKEN          #IMPLIED
  xml:lang      NMTOKEN          #IMPLIED
  dir           (ltr|rtl)        #IMPLIED
  id            ID                #IMPLIED
  xmlns        CDATA             #FIXED 'http://www.w3.org/1999/xhtml'>
```

Attribute value must be ltr or rtl



# XML Attribute List Declaration



```
<!ATTLIST html
  lang          NMTOKEN      #IMPLIED
  xml:lang      NMTOKEN      #IMPLIED
  dir           (ltr|rtl)    #IMPLIED
  id            ID           #IMPLIED
  xmlns        CDATA        #FIXED 'http://www.w3.org/1999/xhtml'>
```

ID

Like NMTOKEN but must begin with letter or \_ :  
Attribute value must be unique



# XML Attribute List Declaration



```
<!ATTLIST html
```

```
  lang          NMTOKEN      #IMPLIED
```

```
  xml:lang      NMTOKEN      #IMPLIED
```

```
  dir           (ltr|rtl)    #IMPLIED
```

```
  id            ID           #IMPLIED
```

```
  xmlns         CDATA        #FIXED 'http://www.w3.org/1999/xhtml'>
```

Any character except XML special characters < and & or the quote character enclosing the attribute value



# XML Attribute List Declaration



TABLE 2.8: Key attribute types used in XHTML 1.0 Strict DTD.

Attribute type	Syntax	Usage
Name token	NMTOKEN	Name (word)
Enumerated	( string1   string2   ... )	List of all possible attribute values
Identifier	ID	Type for id attribute
Identifier reference	IDREF	Reference to an id attribute value
Identifier reference list	IDREFS	List of references to id attribute values
Character data	CDATA	Arbitrary character data (except < and &)





# XML Attribute List Declaration



```
<!ATTLIST html
```

```
  lang      NMTOKEN
```

```
  xml:lang  NMTOKEN
```

```
  dir       (ltr|rtl)
```

```
  id        ID
```

```
  xmlns     CDATA
```

```
#IMPLIED
```

```
#IMPLIED
```

```
#IMPLIED
```

```
#IMPLIED
```

```
#FIXED 'http://www.w3.org/1999/xhtml'>
```

Attribute default declarations



# XML Attribute List Declaration



TABLE 2.9: XML attribute default-value declarations.

Default type	Syntax
No default value provided by DTD, attribute optional	#IMPLIED
Default provided by DTD, may not be changed	#FIXED followed by any valid value (quoted)
Default provided by DTD, may be overridden by user	Any valid value (quoted)
No default value provided by DTD, attribute required	#REQUIRED



# XML Entity Declaration



- ▶ Entity declaration is essentially a macro
- ▶ Two types of entity:
  - ▶ **General**: referenced from HTML document using &

```
<!ENTITY gt "&#62;">
```

Entity name



# XML Entity Declaration



- ▶ Entity declaration is essentially a macro
- ▶ Two types of entity:
  - ▶ **General**: referenced from HTML document using &

```
<!ENTITY gt
```

```
"&#62;">
```

Replacement text;  
recursively replaced if it is a reference



# XML Entity Declaration



- ▶ Entity declaration is essentially a macro
- ▶ Two types of entity:
  - ▶ **General**: referenced from HTML document using &
  - ▶ **Parameter**: reference from DTD using %

```
<!ENTITY gt          "&#62;">
```

```
<!ENTITY % LanguageCode "NMTOKEN">
```

```
<!ATTLIST html  
  lang          NMTOKEN    #IMPLIED  
  xml:lang      %LanguageCode; #IMPLIED
```

101



# XML Entity Declaration



- ▶ Entity declaration is essentially a macro
- ▶ Two types of entity:
  - ▶ **General**: referenced from HTML document using &
  - ▶ **Parameter**: reference from DTD using %

```
<!ENTITY gt          "&#62;">
```

```
<!ENTITY % LanguageCode "NMTOKEN">
```

```
<!ATTLIST html  
  lang          NMTOKEN          #IMPLIED  
  xml:lang      %LanguageCode;  #IMPLIED
```



# DTD Files

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

*System Identifier*: URL for primary DTD document

- ▶ DTD document contains element type, attribute list, and entity declarations
- ▶ May also contain declaration of **external entities**: identifiers for secondary DTD documents



# DTD Files

## External entity name

```
<!ENTITY % HTMLlat1 PUBLIC  
    "-//W3C//ENTITIES Latin 1 for XHTML//EN"  
    "xhtml-lat1.ent">  
%HTMLlat1;
```





# DTD Files

```
<!ENTITY % HTMLlat1 PUBLIC  
    "-//W3C//ENTITIES Latin 1 for XHTML//EN"  
    "xhtml-lat1.ent">  
%HTMLlat1;
```

System identifier (relative URL)



# DTD Files

```
<!ENTITY % HTMLlat1 PUBLIC  
    "-//W3C//ENTITIES Latin 1 for XHTML//EN"  
    "xhtml-lat1.ent">  
%HTMLlat1;
```

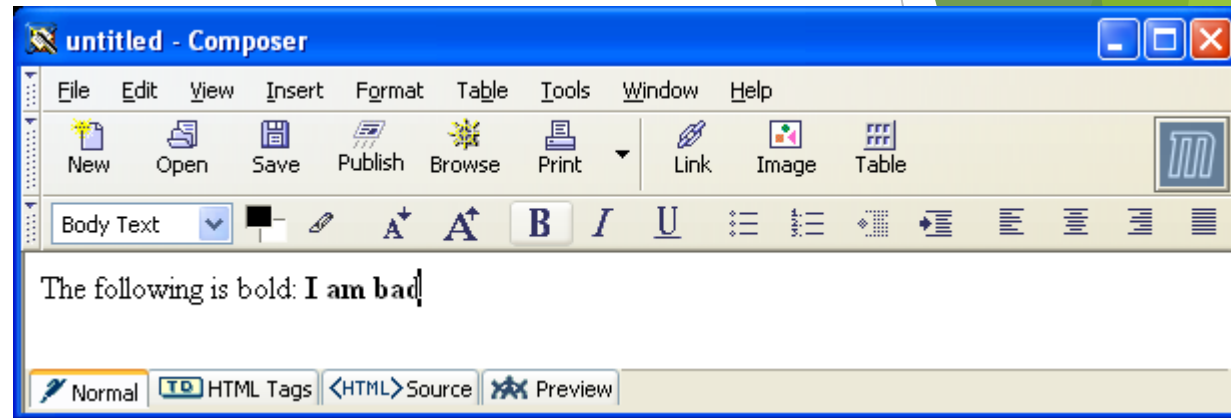
Entity reference; imports content (entity declarations, called *entity set*) of external entity at this point in the primary DTD



# HTML Creation Tools



- ▶ Mozilla Composer



- ▶ Microsoft FrontPage
- ▶ Macromedia Dreamweaver
- ▶ Etc.