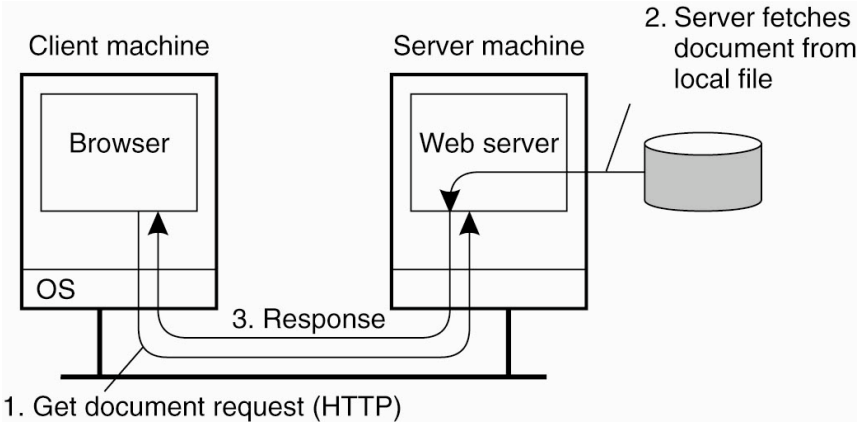
HTML

Dominic Duggan
Stevens Institute of Technology
Based on material by D. Matuszek and S. Mitchell

1

World Wide Web



2

How does the browser display pages?

- Web pages are text files
- Display instructions
- HTML tags
  This is a Paragraph

3

Web standards

- World Wide Web Consortium (www.w3c.org)
  - HTML
  - XHTML
  - CSS
  - Many others

4

What is an HTML File?

- HTML stands for Hypertext Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- An HTML file must have an htm or html file extension

5

HTML Tags

- HTML tags are used to mark up HTML elements
- Angle brackets, < and >
- Most HTML tags come in pairs, like and
- The text between the start and end tags is the element content
- Tags should be properly nested
  - Bad: this is a bold paragraph
  - Good: this is a bold paragraph
- HTML tags are not case sensitive
  - means the same as <B>
- XHTML tags *are* case sensitive and must be *lower case*

6

Structure of an HTML document

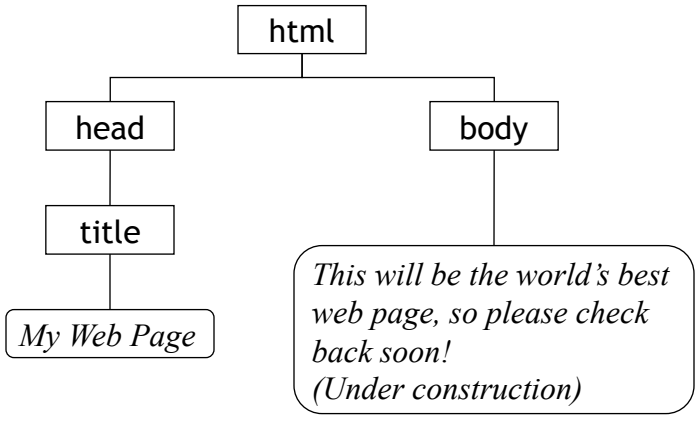
- An HTML document is contained within <html> tags
  - <head> and <body> elements
  - <head> contains a <title>, (for browser window)
  - Almost all other content goes in the <body>

- Example:

```
<html>
  <head>
    <title>My Title</title>
  </head>
  <body>
     Hello, World!
  </body>
</html>
```

7

HTML documents are trees



8

Text in HTML

- Unless marked otherwise, content is text
- Emphasis with and tags
  - Browsers usually display emphasis with *italics*
- Strong emphasis with and tags
  - Browsers usually display strong emphasis with **boldface**
- Headers with <h1>, <h2>, <h3>, <h4>, <h5>, or <h6> tags (and the corresponding end tag, </h1> through </h6>)
  - <h1> is quite large; <h6> is very small
  - Each header goes on a line by itself

9

Whitespace

- Any non-printing characters (space, tab, newline, and a few others)
- Treated as word separators
  - Never line separators
  - Line separator:
 (
 in XHTML)
- Paragraphs: and tags
  - HTML allows to be omitted
- To force HTML to use whitespace exactly as you wrote it, enclose your text in <pre> and </pre> tags
  - <pre> also uses a monospace font
  - <pre> is handy for displaying programs

10

Lists

- Ordered, <ol> to
- Unordered, <ul> to
- Ordered lists typically use numbers: 1, 2, 3, ...
- Unordered lists typically use bullets (•)
- The elements of a list (either kind) are surrounded by <li> and
- Example:

```
The four main food groups are:
<ul>
    <li>Sugar</li>
    <li>Chips</li>
    <li>Caffeine</li>
    <li>Chocolate</li>
</ul>
```

11

Attributes

- Attributes of the form *name*="*value*" add additional information to elements
- Example: To have an ordered list with letters A, B, C, ... instead of numbers, use <ol type="A"> to
  - For lowercase letters, use type="a"
  - For Roman numerals, use type="I"
  - For lowercase Roman numerals, use type="i"
  - In this example, type is an attribute

12

Links

- To link to another document: to
  - Example: You can find more information about my research here.
  - Link text will automatically be underlined and blue (or purple if recently visited)
- To link to another part of the same page,
  - Insert a named anchor: References
  - And link to it with: My references
- To link to a named anchor from a different page, use
  My references

13

Images

- Images are not part of an HTML page; the HTML just tells where to find the image
- To add an image to a page, use:
  <img src="*URL*" alt="*text description*" width="*150*" height="*100*">
  - The src attribute is required; the others are optional
  - The *URL* may refer to any .gif, .jpg, or .png file
  - The alt attribute provides a text alternative for the image
  - The height and width attributes are optional
  - There is no </img> end tag, because <img> is *not* a container

14

Tables

- A <table> contains one or more table rows, <tr>
- Each table row contains one or more table data cells, <td>, or table header cells, <th>
  - Text in <th> cells is boldface and centered
- Each table row should contain the same number of table cells
- To put borders around every cell, add the attribute border="1" to the <table> start tag

15

Example table

```
<table border="1">
 <tr>
  <th>Name</th>  <th>Phone</th>
 </tr>
 <tr>
  <td>Dick</td>  <td>555-1234</td>
 </tr>
 <tr>
  <td>Jane</td>  <td>555-2345</td>
 </tr>
 <tr>
  <td>Sally</td>  <td>555-3456</td>
 </tr>
</table>
```

Name	Phone
Dick	555-1234
Jane	555-2345
Sally	555-3456

16

More about tables

- Excellent for arranging things in rows and columns
  - Wider borders can be set with border="*n*"
  - Text in cells is less crowded if you add the attribute cellpadding="*n*" to the <table> start tag
- Tables can be nested within tables
- Tables, rows, or individual cells may be set to any background color (with bgcolor="*color*")
  - Columns have to be colored one cell at a time
  - (You can also add bgcolor="*color*" to the <body> start tag)

17

A Caution About Use of Tables

- (Amateur) Page Designers sometimes use tables to format the layout of a Web page
- Do not do this!
  - Use <DIV> element and cascading style sheets instead

18

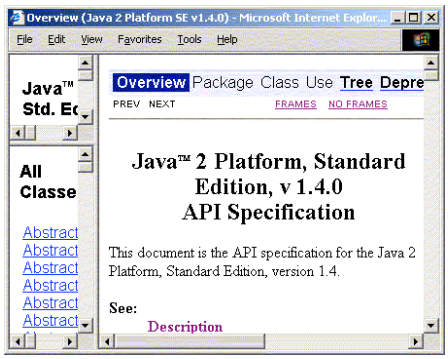
Entities

- Certain characters, such as <, have special meaning in HTML
  - X < 3 and Y > 4 displays as X 4
- Entities encode special characters
- Example:
  - < represents <
  - > represents >
  - & represents &
  - ' represents '
  - " represents "
  -   represents a “non-breaking space”

19

Frames

- Break a browser window up into “panes,” and put a separate HTML page into each pane
  - Example: Java API



20

Framesets

- Frames are enclosed within a frameset
- Replace <body>…</body> with <frameset>…</frameset>
  - Within the <frameset> start tag, use the attributes:
    - rows=*row_height_value_list*
    - cols=*col_width_value_list*
  - The value lists are comma-separated lists of values, where a value is any of:
    - *value*% – that percent of the height or width
    - *value* – that height or width in pixels (usually a bad idea)
    - * – everything left over (use only once)
- Example: <frameset cols="20%,80%">

21

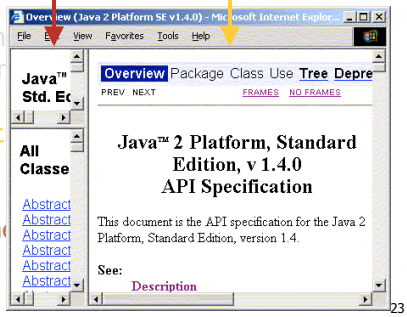
Adding frames to a frameset

- Put as many <frame> tags within a <frameset> as there are rows or columns
  - <frame> is not a container, so there is no </frame> end tag
- Each <frame> should have this attribute:
  - src=*URL* – tells what page to load
- Some optional tags include:
  - scrolling="yes|no|auto" (default is "auto")
  - noresize
- Within a <frameset> you can also put <noframes>*Text to display if no frames*</noframes>

22

Example: The Java API

```
<HTML>
  <HEAD>
    <TITLE>Java 2 Platform SE v1.4.0</TITLE>
  </HEAD>
  <FRAMESET cols="20%,80%">
    <FRAMESET rows="30%,70%">
      <FRAME src="overview-frame.htm
      <FRAME src="allclasses-frame.htm
    </FRAMESET>
    <FRAME src="overview-summary.ht
  </FRAMESET>
  <NOFRAMES>
    <H2>If you see this, you have frame
  </NOFRAMES>
</HTML>
```



23

Example: The Java API

```
<HTML>
  <HEAD>
    <TITLE>Java 2 Platform SE v1.4.0</TITLE>
  </HEAD>
  <FRAMESET cols="20%,80%">
    <FRAMESET rows="30%,70%">
      <FRAME src="overview-frame.htm
      <FRAME src="allclasses-frame.htm
    </FRAMESET>
    <FRAME src="overview-summary.ht
  </FRAMESET>
  <NOFRAMES>
    <H2>If you see this, you have frame
  </NOFRAMES>
</HTML>
```

24

The rest of HTML

- HTML is a large markup language, with a lot of options
  - Read on-line tutorials
    - The w3schools tutorial is among the best
  - Browser View -> Source

25

XHTML

26

The problem with HTML

- HTML started out as a way of way of describing the structure of documents
  - Headers
  - Paragraphs
  - Etc
- HTML pages focused on presentation rather than structure of the underlying data
  - Ultimate fail: embedded graphics for separators
- Consequences:
  - Difficult to maintain consistent look and feel
  - Difficult for search engines

27

HTML Example

```
<html> <head> <title>Your books</title> </head>
<body>
<table border="1">
  <tr>
    <td>Fiction</td>
    <td>The World is Flat</td>
    <td>T Friedman</td>
  </tr>
  <tr>
    <td>Non-Fiction</td>
    <td>Moustache of Freedom</td>
    <td>D Black</td>
  </tr>
</table>
</body>
```

28

XML Example

```
<?xml version="1.0"?>
<inventory>
  <book category="Fiction">
    <title>The World is Flat</title>
    <author>T Friedman</author>
  </book>
  <book category="Non-Fiction">
    <title>Moustache of Freedom</title>
    <author>D Black</author>
  </book>
</inventory>
```

29

XSLT Example

```
<?xml version="1.0"?>
<xsl:transform
      xmlns:xls="http://www/w3/org/…"
      version="1.0">
      <xsl:template match="/">
            <xsl:apply-templates/>
      </xsl:template>
      <xsl:template match="inventory">
        <table border="1">
          <xsl:for-each select="book">
            <tr>
              <td><xsl:value-of select="@category"/></td>
              <td><xsl:value-of select="title"/></td>
              <td><xsl:value-of selct="author"/></td>
            </tr>
          </xsl:for-each>
        </table>
      </xsl:template>
</xsl:transform>
```

```
<?xml version="1.0"?>
<inventory>
  <book category="Fiction">
    <title>The World is Flat</title>
    <author>T Friedman</author>
  </book>
  <book category="Non-Fiction">
    <title>Moustache of Freedom</title>
    <author>D Black</author>
  </book>
</inventory>
```

```
<table border="1">
  <tr>
    <td>Fiction</td>
    <td>The World is Flat</td>
    <td>T Friedman</td>
  </tr>
  <tr>
    <td>Non-Fiction</td>
    <td>Moustache of Freedom</td>
    <td>D Black</td>
  </tr>
</table>
```

30

What is XHTML?

- XHTML stands for Extensible Hypertext Markup Language
  - Stricter and cleaner version of HTML
- XML (Extensible Markup Language) is a markup language designed for describing data
  - XHTML is HTML redefined as an XML application
  - XHTML is a "bridge" between HTML and XML

31

From HTML to XHTML (1)

- XHTML elements must be properly nested
  - bold and italic is wrong
- XHTML documents must be well-formed
  - <html>
    <head> ... </head>
    <body> ... </body>
    </html>
- Tag names must be in lowercase
- All XHTML elements must be closed
  - If an HTML tag is not a container, close it like this:

, <hr />,

32

From HTML to XHTML (2)

- Attribute names must also be in lower case
  - Example: <table width="100%">
- Attribute values must be quoted
  - Example: <table width="100%">
- Attribute minimization is forbidden
  - Example: <frame noresize="noresize">, cannot be abbreviated to <frame noresize>
- The id attribute replaces the name attribute
  - Wrong:
  - Right:
  - Best:

33

SGML and DTDs

- A DTD, or "Document Type Definition" describes the syntax to use for the current document
- There are three different DTDs for XHTML
  - Pick the one you want
  - You must start your XHTML document with a reference to one of these DTDs

34

DOCTYPE declaration (1)

- Every XHTML document must begin with one of the DOCTYPE declarations (DTDs):
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
    - (Essentially the same as XHTML 1.0 Strict)

35

DOCTYPE declaration (2)

- 1.0 Strict
  - Use for really clean markup, with no display information (no font, color, or size information)
  - Use with CSS (Cascading Style Sheets) if you want to define how the document should look
- 1.0 Transitional
  - Use with standard HTML and/or with CSS
  - Allows deprecated HTML elements
- 1.0 Frameset
  - Use if your document uses HTML frames
- 1.1
  - Like 1.0 Strict, but with added support for Chinese

36

XHTML Example

```
<!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"
                xml:lang="en" lang="en">
   <head>
      <title>A simple document</title>
   </head>
   <body>
      <p>A simple paragraph.</p>
   </body>
</html>
```

37

Extension

- A file containing an HTML page should have the extension .html
- According to W3C, an XHTML page should have the extension .xhtml
- …but various tools prefer .html

38

Cascading Style Sheets

39

Another problem with HTML

- Originally intended to describe the *content* of a document
- Browser chooses *presentation* based on HTML tags
  - Document authors want more control
- Enrich HTML with tags and attributes to control appearance
  - Maintenance disaster
- Cascading Style Sheets (CSS)
  - Presentation based on HTML tags can be specialized
  - Instructions for doing this factored out of the document itself

40

Cascading Style Sheets

- A Cascading Style Sheet (CSS) describes the appearance of an HTML page in a separate document
- Advantages:
  - Separate content from presentation
  - Define appearance and layout of all the pages in your web site in a single place
  - It can be used for HTML, XHTML, and XML pages
- Disadvantage:
  - Browser incompatibilities
  - Dreamweaver works around browser bugs

41

CSS syntax (1)

- A file containing a list of
  - selectors (choose tags) and
  - descriptors (what to do with them):
- Example:

```
h1 {color: green; font-family: Verdana}
```

42

CSS syntax (2)

- The general syntax is:
  selector { ***property***: ***value***; }
  - or
    selector, . . . , ***selector*** {
    property: ***value***;
    . . .
    property: ***value***;
    }
  - where
    - *selector* is the tag to be affected
    - *property* and *value* describe the appearance of that tag
    - Spaces after colons and semicolons are optional
    - A semicolon must be used *between* property:value pairs
    - A semicolon after the last pair is recommended but optional

43

Example of CSS

```
/* This is a comment */
h1,h2,h3 {font-family: Arial, sans-serif;}   /* use 1st available font */
p, table, li, address {                 /* apply to all these tags */
  font-family: "Courier New";           /* quote values containing spaces */
  margin-left: 15pt;                    /* specify indentation */
}
p, li, th, td {font-size: 80%;}         /* 80% of size in containing element */
th {background-color:#FAEBD7}  /* colors can be specified in hex */
body { background-color: #ffffff;}
h1,h2,h3,hr {color:saddlebrown;} /* adds to what we said before */
a:link    {color:darkred}               /* an unvisited link */
a:visited {color:darkred}               /* a link that has been visited */
a:active  {color:red}                   /* a link now being visited */
a:hover   {color:red}                   /* when the mouse hovers over it */
```

44

Adapted from: http://www.w3schools.com/css/demo_default.htm

More about selectors (1)

- An XML or HTML tag can be used as a simple element selector:

  ```
  body { background-color: #ffffff }
  ```

- You can use multiple selectors:

  ```
  em, i {color: red}
  ```

  You can repeat selectors:

  ```
  h1, h2, h3 {font-family: Verdana; color: red}
  h1, h3 {font-weight: bold; color: pink}
  ```

  - The last one overrides any earlier ones

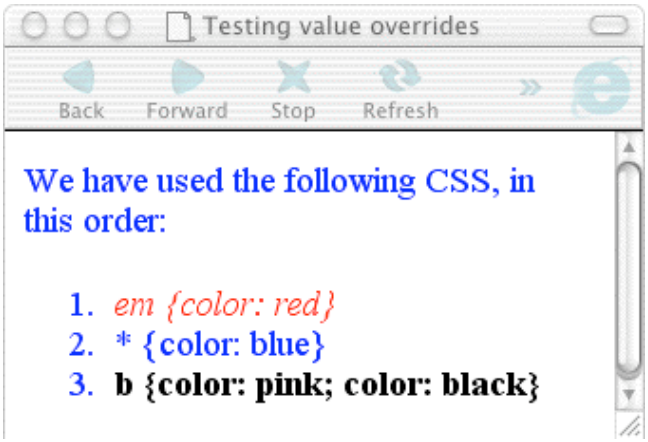
- The universal selector * applies to any and all elements:

  ```
  * {color: blue}
  ```

  - More specific selectors override general ones (so em elements would still be red)

45

Example of overriding



46

More about selectors (2)

A descendent selector chooses a tag with a specific ancestor:

- p code { color: brown }
- selects a code element if it is somewhere inside a paragraph

- A child selector > chooses a tag with a specific parent:

  h3 > em { font-weight: bold }

  selects an em only if its immediate parent is h3

- An adjacent selector chooses an element that immediately follows another:

  b + i { font-size: 8pt }

  Example: I'm bold and I'm italic

  Result will look something like: **I'm bold and** *I'm italic*

47

More about selectors (3)

- A simple attribute selector chooses elements that have a given attribute:
  - Syntax: ***element***[***attribute***] { ***...*** }
  - Example: table[border] { ***...*** }
- An attribute value selector chooses elements that have a given attribute with a given value:
  - Syntax: ***element***[***attribute***="***value***"] { ***...*** }
  - Example: table[border="0"] { ***...*** }

48

More about values

- The syntax for a CSS rule is:
  selector, …, ***selector*** { ***property***: ***value***; . . . ***property***: ***value*** }
- The value is whatever occurs between the colon and the semicolon (or closing brace)
- Example: * {font-family: Trebuchet, Verdana, sans-serif;}
  - Use Trebuchet, else use Verdana, else use default browser sans serif font
- section {border: thin solid blue;}
  - Borders are to be thin *and* solid *and* blue

49

The class attribute

- Allows you to have *different* styles for (different occurences of) the *same* element
  - In the style sheet:
    p.important {font-size: 24pt; color: red}
    p.fineprint {font-size: 8pt}
  - In the HTML:
    The end is nigh!
    Offer ends 1/1/97.
- To define a selector that applies to any element with that class, just omit the tag name (but keep the dot):
  .fineprint {font-size: 8pt}

50

The id attribute

- The id attribute is defined like the class attribute, but uses # instead of .
  - In the style sheet:
    p#important {font-style: italic} or
    # important {font-style: italic}
  - In the HTML:
    <p id="important">
- class and id can both be used, and do not need to have different names:
  <p class="navigation_bar" id="navigation_bar">
- **Important difference:** id is used to specify a **unique** identifier, so it should only be used **once** in any given document

51

div and span

- div and span are HTML elements whose only purpose is to hold CSS information
- div ensures there is a line break before and after (so it's like a paragraph); span does not
- Example:
  - CSS: div {background-color: #66FFFF}
    span.color {color: red}
  - HTML: This div is treated like a paragraph, but this span is not.

52

Using style sheets

- External style sheet
  - This is the most powerful
  - Applies to both XHTML/HTML and XML
  - All of CSS can be used
- Inline styles
  - Applies to XHTML/HTML, not to XML
  - Limited form of CSS syntax

53

External style sheets

- In XHTML/HTML, within the <head> element:
  <link rel="stylesheet" type="text/css"
  href="***Style Sheet URL***">
- As a PI in the prologue of an XML document:
  <?xml-stylesheet href="***Style Sheet URL***"
  type="text/css"?>
- Note: "text/css" is the MIME type

54

Embedded style sheets

- In XHTML/HTML, within the <head> element:

```
<style type="text/css">
    <!--
    CSS Style Sheet
    -->
</style>
```

55

Inline style sheets

- The style attribute can be added to any HTML element:

```
<html-tag style="property: value">        or
<html-tag style="property: value;
    property: value; ...; property: value">
```

- Advantage:
  - Useful if you only want a small amount of markup
- Disadvantages:
  - Mixes display information into HTML
  - Clutters up HTML code
  - Can't use full range of CSS features

56

Cascading order

- Styles will be applied to HTML in the following order:
  1. Browser default
  2. External style sheet
  3. Internal style sheet (inside the <head> tag)
  4. Inline style (inside other elements, outermost first)
- When styles conflict, the "nearest" (most recently applied) style wins

57

Example of cascading order

- External style sheet:

```
h3 {  color: red;
      text-align: left;
      font-size: 8pt
   }
```

- Internal style sheet:

```
h3 {  text-align: right;
      font-size: 20pt
   }
```

- Resultant attributes:

```
color: red;
text-align: right;
font-size: 20pt
```

58

A novel example: XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The Great American Novel</title>
  <link rel="stylesheet" href="novel.css" type="text/css" />
</head>

<body>
  <p class="foreword">This is the great American novel.</p>

  <div class="chapter">
    <p>It was a dark and stormy night.</p>

    <p>Suddenly, a shot rang out!</p>
  </div>
</body>
</html>
```

59

A novel example: CSS

```
.chapter {
    font-family: Papyrus,
        "Comic Sans MS",
fantasy;
}
.chapter {
    display: block
}
.chapter:before {
    content: "New chapter: "
}

.chapter:first-letter {
    font-size: 200%;
    float: left
}
```

```
p {
    display: block
}
p.foreword {
    border: solid red;
    padding: 10px;
    font-family: Impact;
    color: blue;
}
```

60

A novel example: Results

Opera (Macintosh)

This is the great American novel.

New chapter:

It was a dark and stormy night.

Suddenly, a shot rang out!

IE 6 (Windows)

This is the great American novel.

It was a dark and stormy night.

Suddenly, a shot rang out!

Firefox (Macintosh)

This is the great American novel.

New chapter:

It was a dark and stormy night.

Suddenly, a shot rang out!

IE 8 (Windows)

This is the great American novel.

New chapter:

It was a dark and stormy night.

Suddenly, a shot rang out!

61

Some font properties and values

- font-family:
  - inherit (same as parent)
  - Verdana, "Courier New", ... (if the font is on the client computer)
  - serif | sans-serif | cursive | fantasy | monospace (Generic: your browser decides which font to use)
- font-size:
  - inherit | smaller | larger | xx-small | x-small | small | medium | large | x-large | xx-large | ***12***pt
- font-weight:
  - normal | bold |bolder | lighter | 100 | 200 | ... | 700
- font-style:
  - normal | italic | oblique

62

Shorthand properties

- Often, many properties can be combined:

  h2 { font-weight: bold; font-variant: small-caps; font-size: 12pt; line-height: 14pt; font-family: sans-serif }

  can be written as:

  h2 { font: bold small-caps 12pt/14pt sans-serif }

63

Colors and lengths

- color: and background-color:
  - aqua | black | blue | fuchsia | gray | green | lime | maroon | navy | olive | purple | red | silver | teal | white | #FF0000 | #F00 | rgb(255, 0, 0) | ***Additional browser-specific names (not recommended)***
- These are used in measurements:
  - em, ex, px, %
    - font size, x-height, pixels, percent of inherited size
  - in, cm, mm, pt, pc
    - inches, centimeters, millimeters, points (1/72 of an inch), picas (1 pica = 12 points), relative to the inherited value

64

Some text properties and values

- text-align:
  - left | right | center | justify
- text-decoration:
  - none | underline | overline | line-through
- text-transform:
  - none | capitalize | uppercase | lowercase
- text-indent
  - ***length*** | ***10%*** (indents the first line of text)
- white-space:
  - normal | pre | nowrap

65

Pseudo-classes

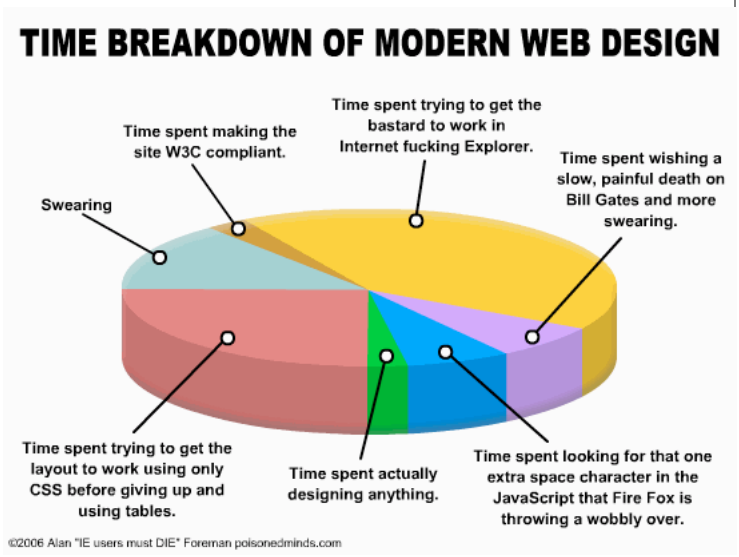
- Pseudo-classes are elements whose state (and appearance) may change over time
- Syntax: ***element*:*pseudo-class*** {...}
  - :link
    - a link which has not been visited
  - :visited
    - a link which has been visited
  - :active
    - a link which is currently being clicked
  - :hover
    - a link which the mouse is over (but not clicked)
- Pseudo-classes are allowed anywhere in CSS selectors

66

Choosing good names

- CSS is designed to *separate content from style*
  - Therefore, names that will be used in HTML or (especially) in XML should describe *content, not style*
- Example:
  - Suppose you define `span.huge {font-size: 36pt}` and you use `<span class="huge">` throughout a large number of documents
  - Now you discover your users hate this, so you change the CSS to be `span.huge {font-color: red}`
  - Your name is inappropriate; do you change all your documents?
  - If you had started with `span.important {font-size: 36pt}`, your documents wouldn't look so dumb

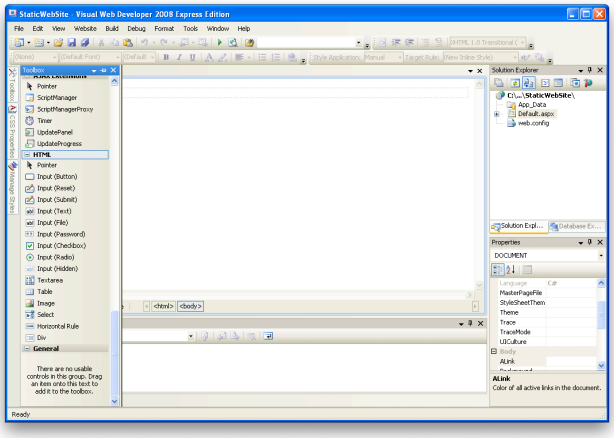
67



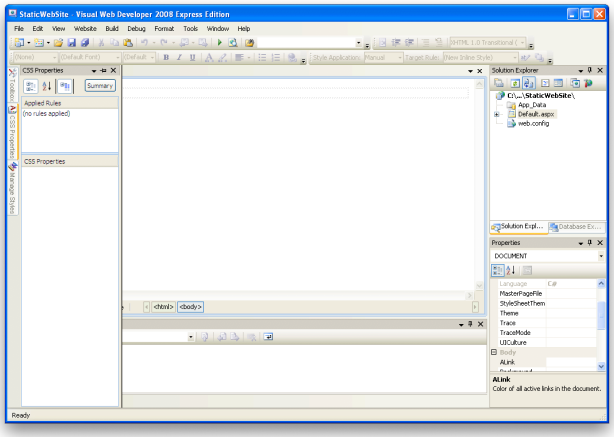
68

Creating Web Pages in Visual Web Developer

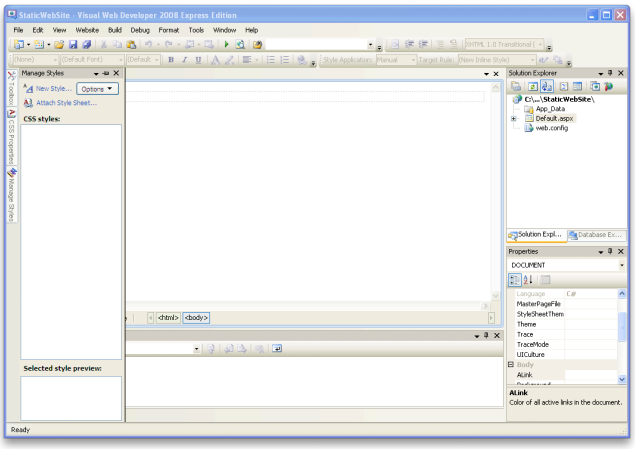
Toolbox for adding controls and HTML elements



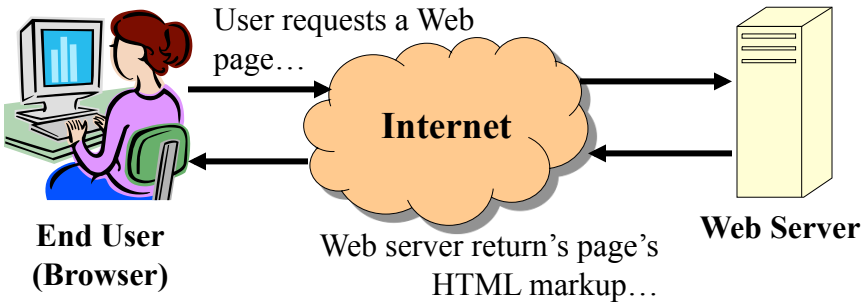
CSS Rules and Properties



Style Sheets



The Client-Server Model



CLIENT SERVER

73

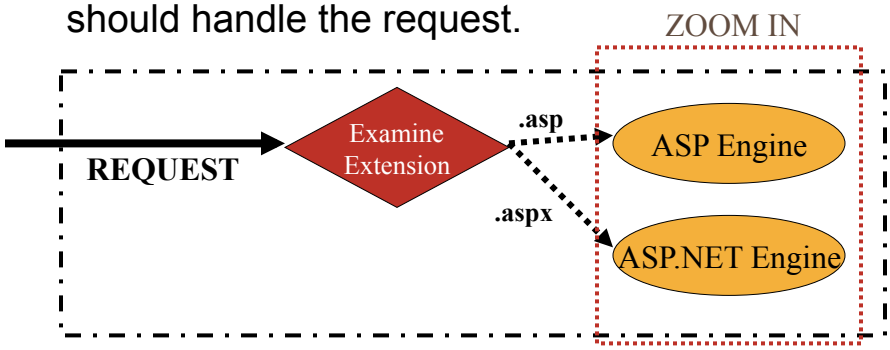
Server-Side Processing

- For static content – HTML pages, images, CSS files, etc. – return the requested content as-is.
- For dynamic Web technologies, like ASP.NET, delegate the request to the appropriate *engine*.
  - Must generate valid markup

74

Server-Side Processing

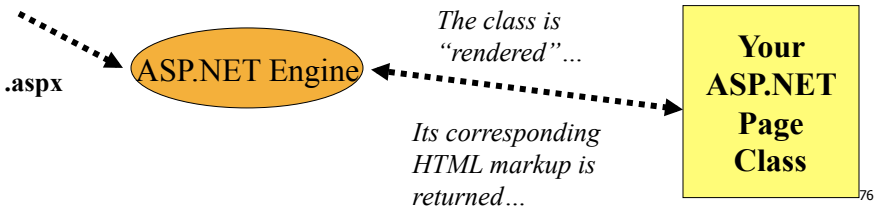
- IIS receives the request, examines the extension, and decides who should handle the request.



75

Server-Side Processing

The ASP.NET engine loads the requested page's corresponding class and *renders* the page, generating the page's HTML output. This output is then returned to the Web server, which is returned to the requesting browser.



76

Server-Side Processing

- To execute server-side code, the client must either:
  1. Click a "submit" button to submit a form
  2. Execute client-side Javascript

77

Server-Side Processing

- A *form* is an HTML element that allows for a user to send data to a specified URL.

```
<form method="post|get" action="URL">
   …
</form>
```

- When the form is submitted, the browser directs the user to the specified *URL* and sends along the values of all of the <input> HTML elements from within the form.

78

Server-Side Processing

- Button Web controls are rendered as *submit buttons*, which are buttons that, when clicked, cause the form to postback.
  - <input type="submit" … />
- ASP.NET also can provide the plumbing for submitting based on some user action, such as choosing an item from a DropDownList, or changing the value of a TextBox.

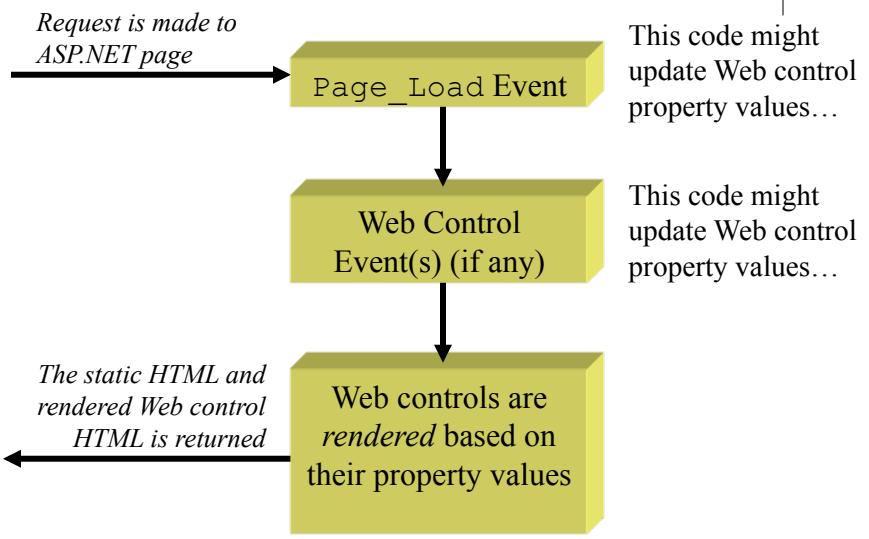
79

Web Controls in ASP.NET

- Recall that ASP.NET pages are composed of two regions:
  1. **A markup region** – contains a mix of static HTML and *Web controls*
  2. **A code region** – contains server-side code and server-side event handlers
- When an ASP.NET page is requested, the ASP.NET engine processes the page, resulting in HTML output.

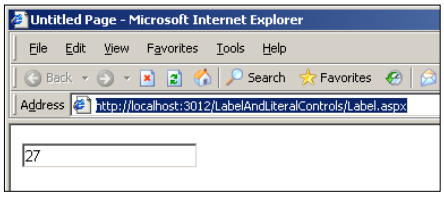
80

From Web Control to HTML (1)



From Web Control to HTML (2)

- Each Web control is rendered into an appropriate HTML element
  - Elements and attributes based on properties of Web control
- Example: A TextBox Web control renders as an <input> element with its type attribute set to "text".
  - With an ID property value of Age and a Text property value of 27, it will render as:




```
<input type="text" id="Age" name="Age" value="27" />
```

Master Pages

Master Pages

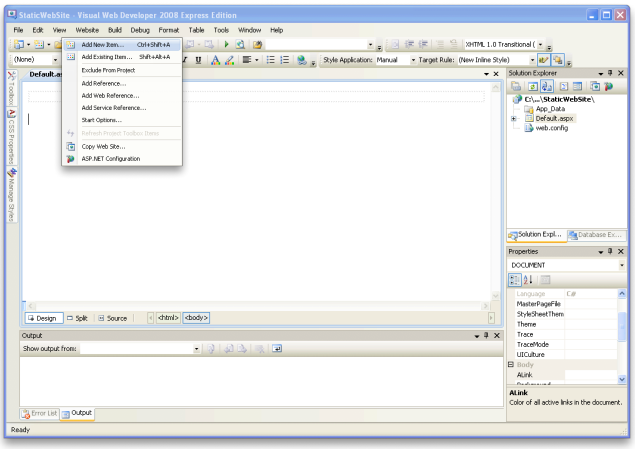
- A master page is a single, reusable template for all pages
  - Design the page layout
  - Allocate region(s) where content unique to each page will go
- New pages are linked to master template
  - Changes to master reflected to all pages
  - No more page layout copy and paste!

Tips for Creating a Professional Looking Website

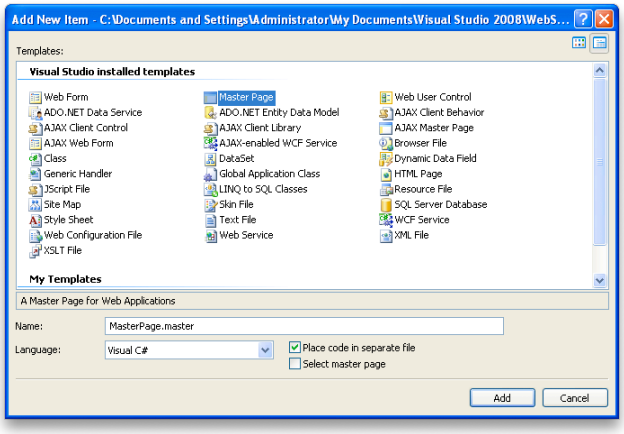
- Most of us are not artistically inclined.
  - Ask a heart surgeon to design a tattoo?
- Use existing designs created by artistically talented individuals:
  - www.OpenDesign.org – more than 1,000 free, open-source website designs.
  - www.FreeCssTemplates.org – more than 350 free, open-source website designs.

85

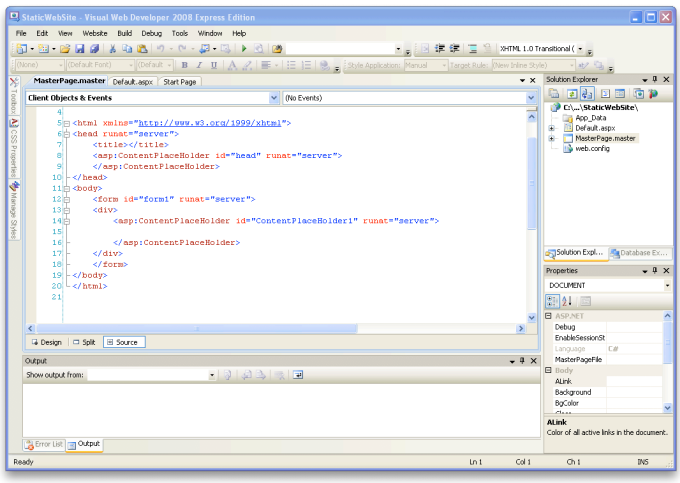
Creating ASP.NET Pages from Master Pages



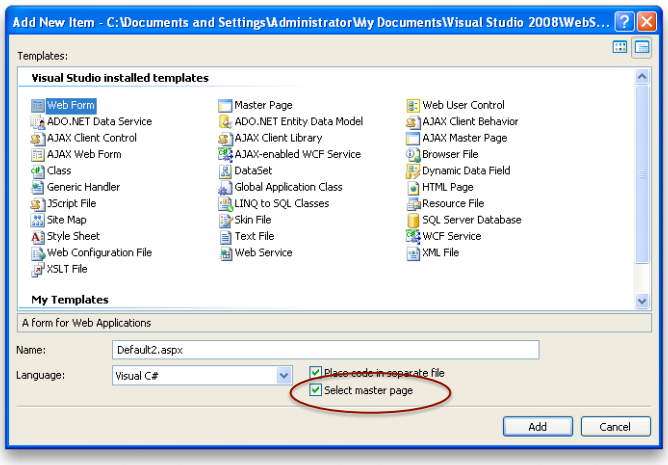
Creating ASP.NET Pages from Master Pages



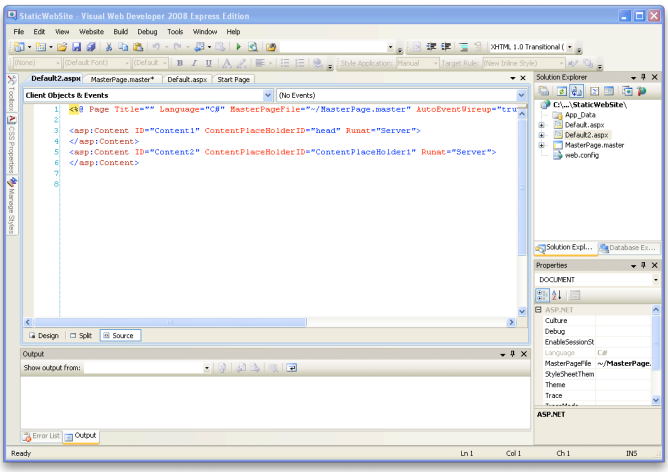
Creating ASP.NET Pages from Master Pages



Creating ASP.NET Pages from Master Pages



Creating ASP.NET Pages from Master Pages

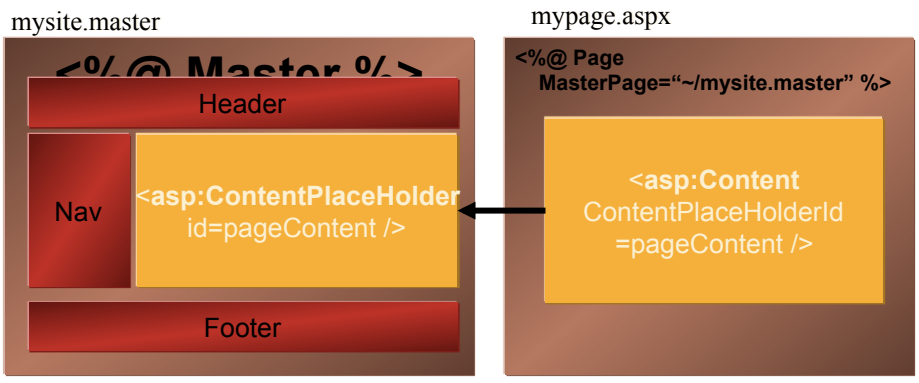


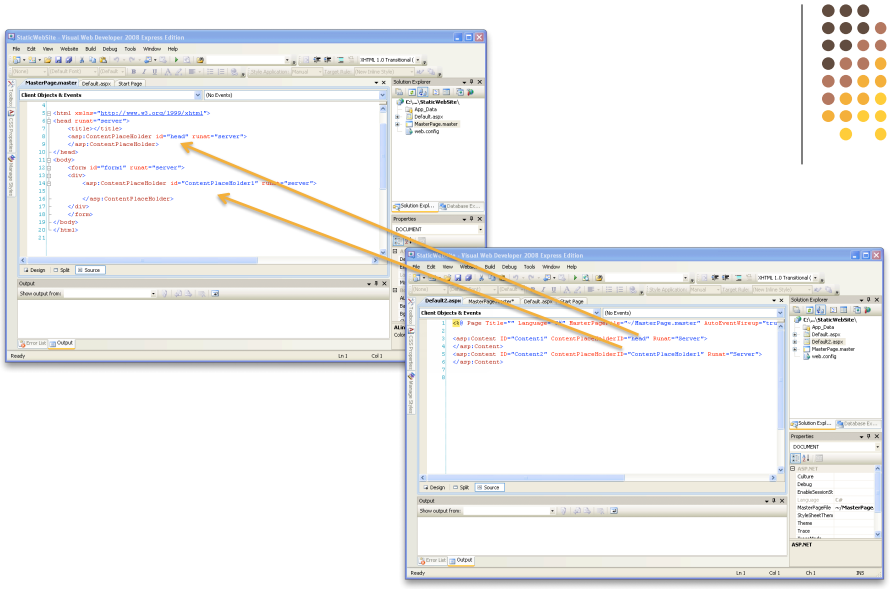
What's Happening Back There?

- When an ASP.NET page that uses a master page is requested, the ASP.NET engine grabs the Master Page being used
- It then fuses the ASP.NET page's Content regions into the Master Pages corresponding ContentPlaceHolder regions.

91

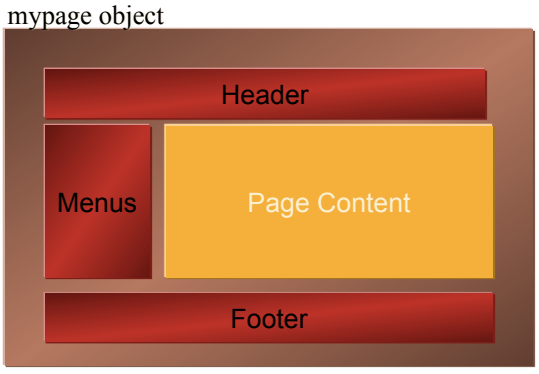
- The Page's Content element defines the markup that is "infused" into the Master Page's corresponding ContentPlaceHolder at runtime.





Master Pages

- A single `Page` object is compiled from the resulting infusion of page and master page content.



94

What's Happening Back There?

- This "fusion" happens
  - the first time a page is requested and
  - subsequent requests after the master page or ASP.NET page has been modified.
- So pages referencing master pages are automatically refreshed after changes, the next time the page is requested.

95

More Advanced Master Page Trickery

- Master pages can contain server-side source code.
  - Goes through the same lifecycle as the ASP.NET page
    - Has a `Page_Load` event handler
    - Etc.

96

More Advanced Master Page Trickery

- In the master page, you can specify the default markup for a given region.
  - Simply add the markup within the ContentPlaceHolder Web control.
- If the ASP.NET Web page does not contain a Content Web control that references a ContentPlaceHolder, the default ContentPlaceHolder content is used.

97