HTML5, CSS3, and JavaScript

6th Edition

Tutorial 6

Working with Tables and Columns

Carey

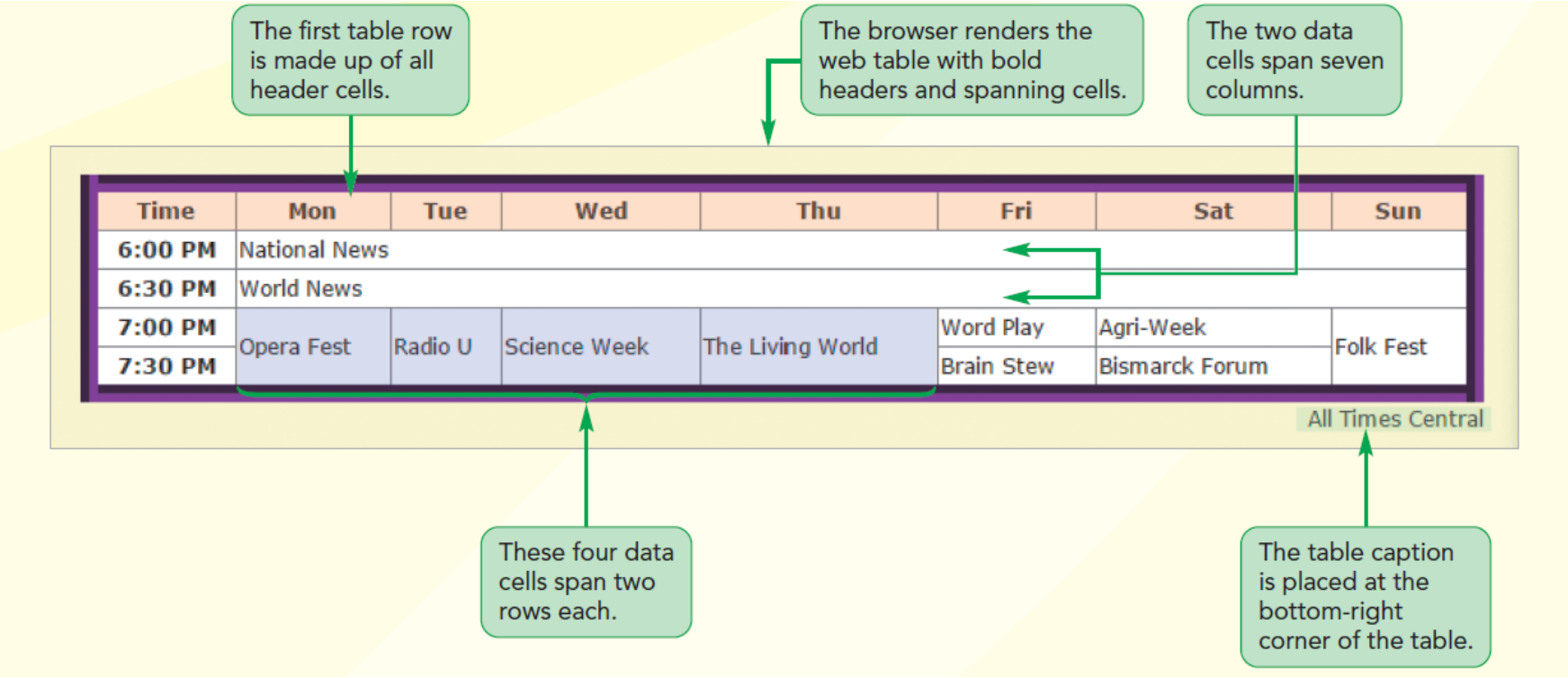
Objectives

- Explore the structure of a web table
- Create table heading and data cells
- Apply CSS styles to a table
- Create cells that span multiple rows and columns
- Add a caption to a table

Objectives (continued)

- Create row and column groups
- Apply styles to row and column groups
- Display page elements in table form
- Create a multi-column layout

Structure of a Web Table



Introducing Web Tables

- **Web table**
  - HTML structure that consists of multiple table rows
  - Each table row contains one or more table cells
  - Effective tool for organizing and classifying web page content
  - Consists of a `table` element

Marking Tables and Table Rows

- A `table` element contains a collection of table rows marked using the `tr` (table row) element
- A table contains cells within each row
- Size of a table is defined by
  - number of table rows
  - number of cells within rows

Marking Tables and Table Rows (continued 1)

- General structure of a web table:

```
<table>
    <tr>
        table cells
    </tr>
    <tr>
        table cells
    </tr>
    …
</table>
```

Marking Tables and Table Rows (continued 2)

Figure 6-2 Marking a table and table rows

```
<section id="main">
   <img src="dlr_logo.png" alt="DLR" />
   <h2>DLR Nightly Schedule</h2>
   <p>DLR airs listener-supported public radio in Bismarck,
      North Dakota from 5:00 a.m. to 10:30 p.m.
      You can <a href="#">Listen Live</a> to streaming audio
      of our broadcast. Please refer below for our current
      nightly schedule.</p>

   <table class="schedule">
      <tr>
      </tr>
      <tr>
      </tr>
      <tr>
      </tr>
   </table>

</section>
```

table element marks the web table

tr element marks each table row

class attribute with a value of "schedule" sets this table in its own class

Marking Table Headings and Table Data

- Web tables support two types of table cells
  - Header cells
    - Contains content placed at the top of a column or beginning of a row
    - By default, displays text in bold and centers text horizontally
    - Marked using the `th` element

Marking Table Headings and Table Data (continued 1)

- Data cells
  - Contains content within columns or rows
  - By default, displays text as unformatted text and is aligned to the left within the cell
  - Marked using the `td` element

Marking Table Headings and Table Data (continued 2)

Figure 6-3 Marking table header cells

the th element marks header cells placed in the first row to identify the content of each column

header cells at the start of each row identifies the row content

```
<table class="schedule">
   <tr>
      <th>Time</th>
      <th>Mon</th>
      <th>Tue</th>
      <th>Wed</th>
      <th>Thu</th>
      <th>Fri</th>
      <th>Sat</th>
      <th>Sun</th>
   </tr>
   <tr>
      <th>6:00 PM</th>
   </tr>
   <tr>
      <th>6:30 PM</th>
   </tr>
</table>
```

Marking Table Headings and Table Data (continued 3)

Figure 6-4 Marking table data cells

```
    <tr>
        <th>6:00 PM</th>
        <td>National News</td>
        <td>National News</td>
        <td>National News</td>
        <td>National News</td>
        <td>National News</td>
        <td>National News</td>
        <td>National News</td>
    </tr>
    <tr>
        <th>6:30 PM</th>
        <td>World News</td>
        <td>World News</td>
        <td>World News</td>
        <td>World News</td>
        <td>World News</td>
        <td>World News</td>
        <td>World News</td>
    </tr>
</table>
```

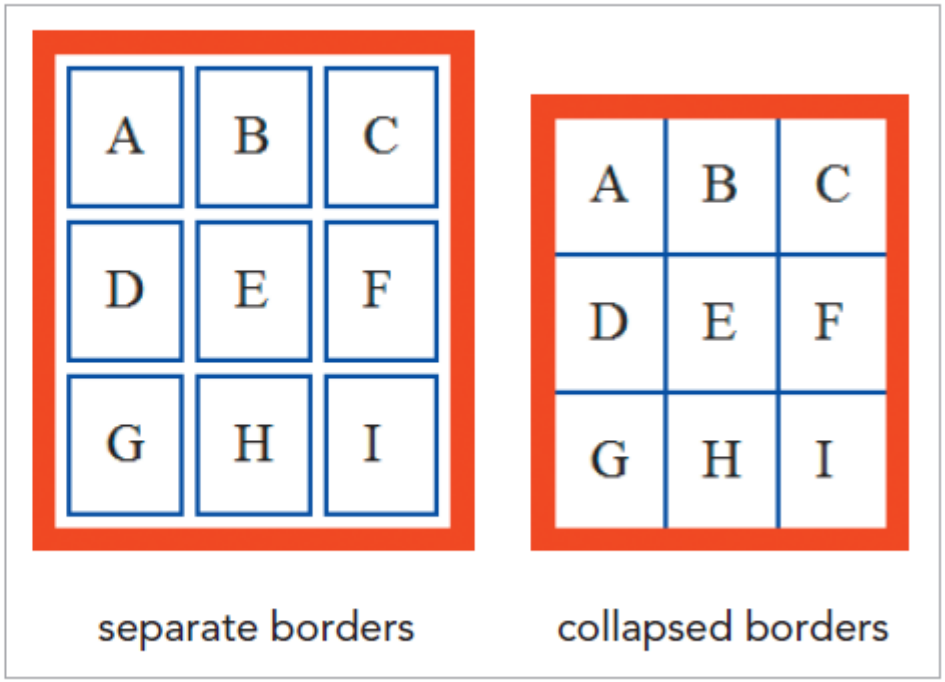
the td element marks table cell content that is not considered the head of a row or column

Adding Table Borders with CSS

- The CSS `border` property is used to add borders to any part of a web table
- Borders need not be of the same style
- Two style choices for borders
  - Separate borders
  - Collapsed borders

Adding Table Borders with CSS (continued 1)

Figure 6-8 Separate and collapsed borders



© 2016 Cengage Learning

Adding Table Borders with CSS (continued 2)

- To choose between separate or collapsed borders model apply the following property to the `table` element:

  `border-collapse: type;`

  where *`type`* is either `separate` or `collapse`

- The separate borders model sets the spacing between borders using

  `border-spacing: value`

  where *`value`* is in CSS units of measure

Adding Table Borders with CSS (continued 3)

- The collapsed borders model
  - Borders from adjacent elements are merged together to form a single border
  - Borders are joined to combine their features
  - Combining adjacent borders with different widths, styles, or colors is complicated

Adding Table Borders with CSS (continued 4)

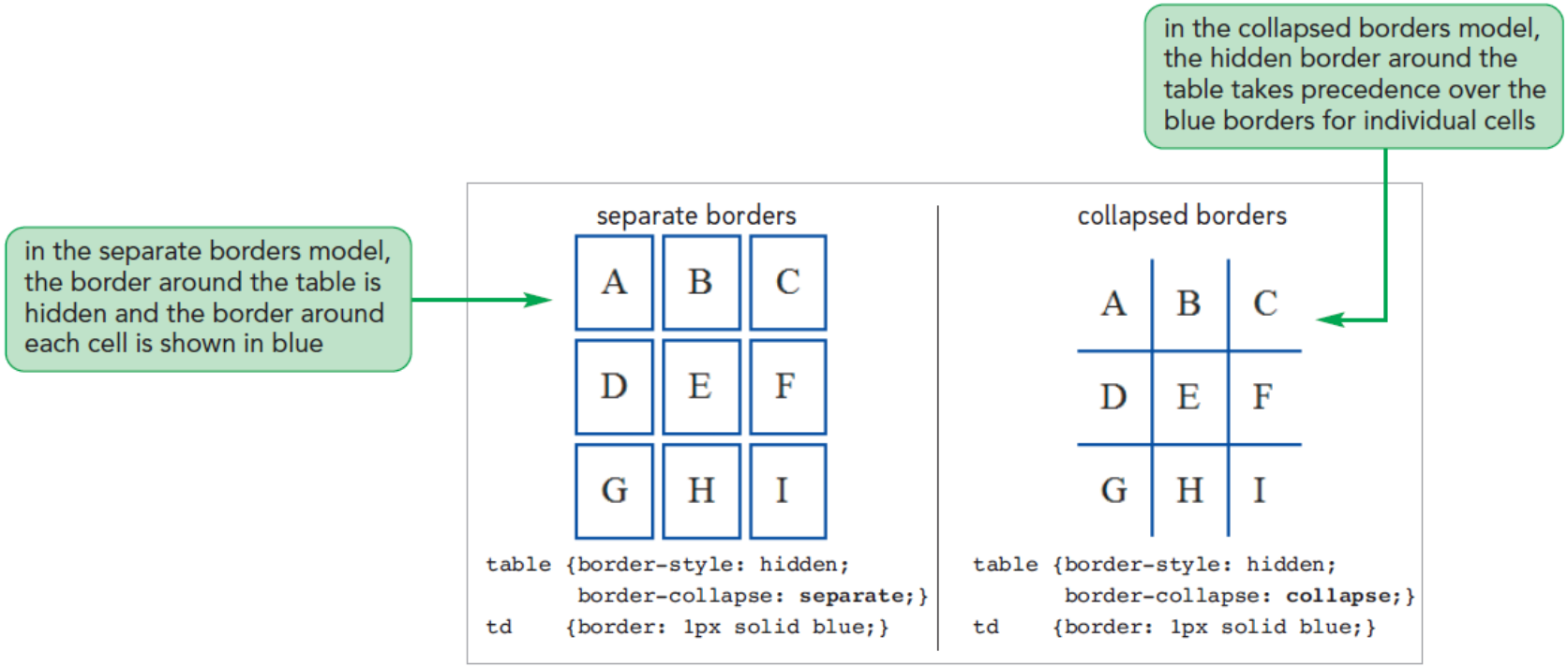
- Five rules to reconcile the differences between adjacent borders
  - If either border has a style of `hidden`, the collapsed border is hidden
  - Border style of `none` is overridden by another border style
  - The style of wider border takes priority over the narrower border if neither border is hidden

Adding Table Borders with CSS (continued 5)

- Double borders have higher precedence followed by solid, dashed, ridge, outset, groove and inset
- If borders differ only in color, precedence is given to borders
- Precedence to borders in decreasing order
  - Borders around individual table cells
  - Borders for table rows
  - Borders for row groups
  - Borders for columns
  - Borders for column groups
  - Borders around the entire table

Adding Table Borders with CSS (continued 6)

Figure 6-9 Reconciling hidden borders



© 2016 Cengage Learning

Spanning Rows and Columns

- Spanning cells
  - Single cell that occupies more than one cell row and/or column
  - Created by adding `rowspan` and/or `colspan` attributes to `td` or `th` elements
  - Spanning starts in the cell with `rowspan` or `colspan` attributes and covers the cells in the right and below the initial cell

Spanning Rows and Columns (continued 1)

- `rowspan="rows"`

  where *`rows`* is the number of rows that the cell occupies

- `colspan="cols"`

  where *`cols`* is the number of columns that the cell occupies

Spanning Rows and Columns (continued 2)

Figure 6-13 Spanning several columns with a single cell

```
<table class="schedule">
   <tr>
      <th>Time</th>
      <th>Mon</th>
      <th>Tue</th>
      <th>Wed</th>
      <th>Thu</th>
      <th>Fri</th>
      <th>Sat</th>
      <th>Sun</th>
   </tr>
   <tr>
      <th>6:00 PM</th>
      <td colspan="7">National News</td>
   </tr>
   <tr>
      <th>6:30 PM</th>
      <td colspan="7">World News</td>
   </tr>
</table>
```

remaining six td elements removed from the second and third rows to keep the size at 8 total columns

sets each cell to span 7 columns within its row

Spanning Rows and Columns (continued 3)

Figure 6-15 Row-spanning cells

four table cells in the first row

only three table cells are required for the second and third rows

```
<table>
   <tr>
      <td rowspan="3">1: This cell spans three rows</td>
      <td>2</td>
      <td>3</td>
      <td>4</td>
   </tr>
   <tr>
      <td>5</td>
      <td>6</td>
      <td>7</td>
   </tr>
   <tr>
      <td>8</td>
      <td>9</td>
      <td>10</td>
   </tr>
</table>
```

HTML code

1: This cell spans three rows	2	3	4
	5	6	7
	8	9	10

resulting table

© 2016 Cengage Learning

Creating a Table Caption

- Marked using the `caption` element

  `<caption>`*`content`*`</caption>`

  where *`content`* is the content contained within the caption
- Listed immediately after the `<table>` tag
- Only one caption is allowed per web table
- Inherits the text styles associated with the table

Creating a Table Caption (continued 1)

- By default, captions are placed above the tables
- To specify the location, use the `caption-side` property

  `caption-side:` *`position`*;

where *`position`* is either `top` or `bottom`

Creating a Table Caption (continued 2)

Figure 6-20 Adding a caption to a web table

```
<table class="schedule">
   <caption>All Times Central</caption>
   <tr>
      <th>Time</th>
      <th>Mon</th>
      <th>Tue</th>
      <th>Wed</th>
      <th>Thu</th>
      <th>Fri</th>
      <th>Sat</th>
      <th>Sun</th>
   </tr>
```

Figure 6-21 Adding a caption to a web table

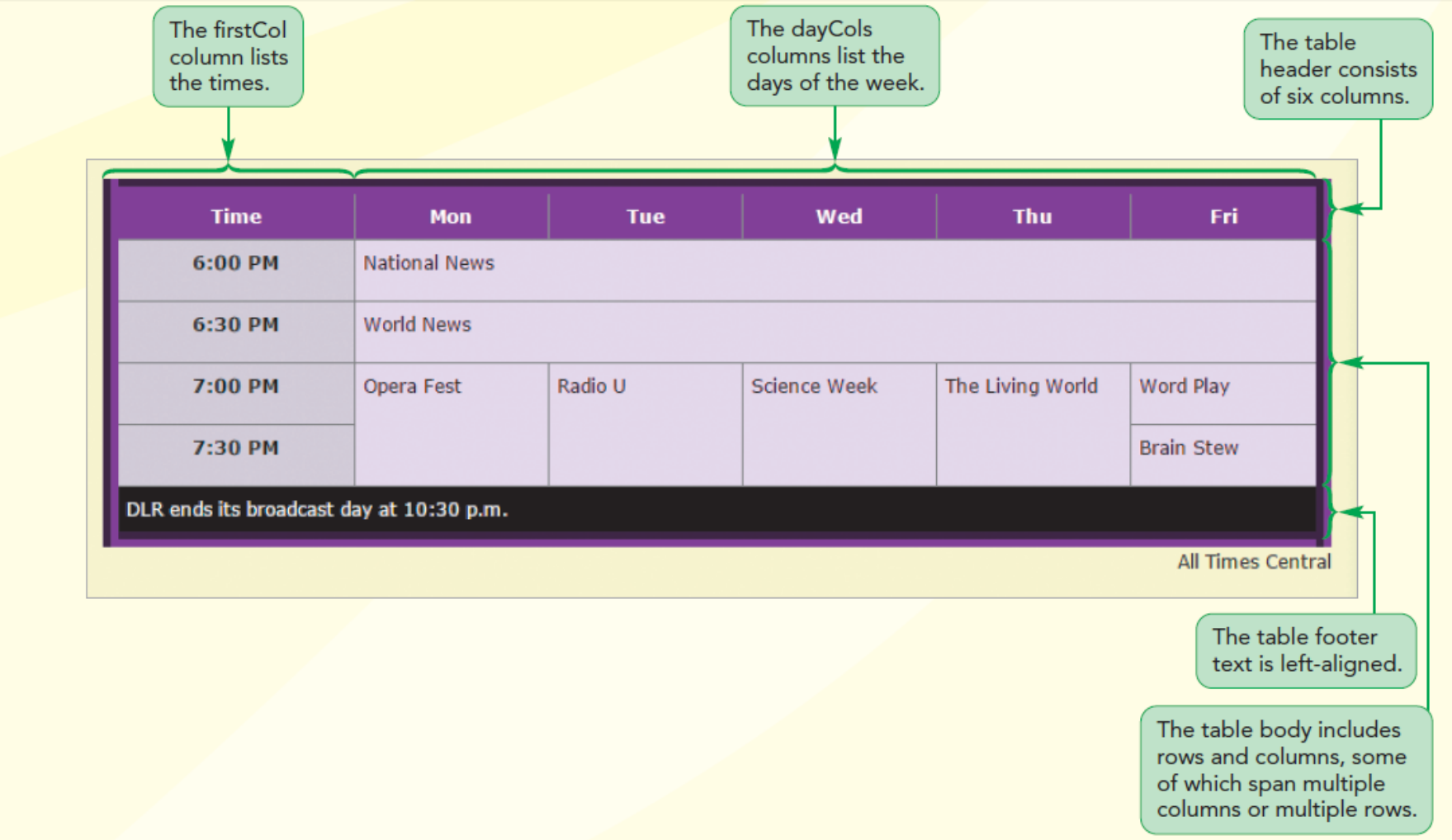
```
/* Table Caption Styles */

table.schedule caption {
   caption-side: bottom;
   text-align: right;
}
```

places the caption at the bottom of the table

right-aligns the caption text

Rows and Column Groups



Creating Row Groups

- Row groups contain specific table information
- Allows to create different styles for groups of rows
- HTML supports three row groups
  - Rows that belong to the table head
    - Marked using `thead` element
  - Rows that belong to the table footer
    - Marked using `tfoot` element
  - Rows that belong to the table body
    - Marked using `tbody` element

Creating Row Groups (continued 1)

- The general structure of a web table divided into row groups

```
<table>
   <thead>table rows</thead>
   <tfoot>table rows</tfoot>
   <tbody>table rows</tbody>
</table>
```

- Only one `thead` and `tfoot` element is allowed per table

Creating Row Groups (continued 2)

Figure 6-23 Marking row groups

The thead element marks the row group for the table header

The tfoot element marks the row group for the table footer

The tbody element marks the row group for the table body

```
<table class="schedule">
   <caption>All Times Central</caption>
   <thead>
      <tr>
         <th>Time</th>
         <th>Mon</th>
         <th>Tue</th>
         <th>Wed</th>
         <th>Thu</th>
         <th>Fri</th>
         <th>Sat</th>
         <th>Sun</th>
      </tr>
   </thead>
   <tfoot>
      <tr>
         <td colspan="8">DLR ends its broadcast day at 10:30 p.m.</td>
      </tr>
   </tfoot>
   <tbody>
   <tr>
      <th>6:00 PM</th>
      <td colspan="7">National News</td>
   </tr>
```

```
      <tr>
         <th>10:00 PM</th>
         <td colspan="7">World News Feed</td>
      </tr>
   </tbody>
</table>
```

Creating Column Groups

- Columns are determined implicitly based on the number of cells within the table rows
- Columns are identified by the `col` element
- To identify individual columns, use the `id` and/or `class` attributes

Creating Column Groups (continued 1)

- Columns can be referred using the following `colgroup` element:

```
<table>
   <colgroup>
        columns
   </colgroup>
   table rows
</table>
```

where *`columns`* are the individual columns defined within the group

Creating Column Groups (continued 2)

- Columns and column groups accept only CSS style properties to modify
  - column borders
  - background
  - width
  - visibility

Creating Column Groups (continued 3)

Figure 6-26 Defining a column group

col element references the first column

col element references the next seven columns

```
<table class="schedule">
   <caption>All Times Central</caption>
   <colgroup>
      <col id="firstCol" />
      <col class="dayCols" span="7" />
   </colgroup>
   <thead>
```

Figure 6-27 Formatting the table columns

displays the first column with a gray background

displays the day columns with a pink background

```
/* Column Group Styles */

col#firstCol {
   background: rgb(218, 210, 218);
}

col.dayCols {
   background: rgb(255, 220, 255);
}
```

Exploring CSS Styles and Web Tables

- Levels of precedence in the table styles in decreasing order
  - Table cells
  - Rows
  - Row groups
  - Columns
  - Column groups
  - Table

Working with Width and Height

- By default, browsers attempt to fit more content in each column before wrapping the cell text
- Extra space is divided equally among columns if the width of a table is larger than its individual columns
- Column widths are set using the `width` property

Working with Width and Height (continued 1)

- The height of each row is based on the height of the tallest cell
- A uniform row height is defined by applying the height style to table rows within each row group
- The `vertical-align` property is used to move the cell text

Working with Width and Height (continued 2)

Figure 6-30 Setting the column width

sets the width of the first column to 16% of the width of the table

sets the width of the day columns to 12%

```
col#firstCol {
   background: rgb(218, 210, 218);
   width: 16%;
}

col.dayCols {
   background: rgb(255, 220, 255);
   width: 12%;
}
```

Figure 6-31 Setting the row height

sets the height of the row in the table header to 30 pixels

sets the height of the rows in the table body to 40 pixels

```
table.schedule tfoot {
   background: black;
   color: white;
}

table.schedule thead tr {
   height: 30px;
}

table.schedule tbody tr {
   height: 40px;
}
```

Applying Table Styles to Other Page Elements

- Apply a table layout to other HTML elements using the CSS `display` property

Figure 6-35 Table display styles

Display Style	Equivalent HTML Element
`display: table;`	table (treated as a block-level element)
`display: table-inline;`	table (treated as an inline element)
`display: table-row;`	tr
`display: table-row-group;`	tbody
`display: table-header-group;`	thead
`display: table-footer-group;`	tfoot
`display: table-column;`	col
`display: table-column-group;`	colgroup
`display: table-cell;`	td or th
`display: table-caption;`	caption

© 2016 Cengage Learning

Applying Table Styles to Other Page Elements (continued)

Figure 6-36 Applying table styles to a definition list

```
<dl>
   <div>
      <dt>bandwidth</dt>
      <dd>A measure of data transfer speed over
          a network</dd>
   </div>
   <div>
      <dt>HTTP</dt>
      <dd>The protocol used to communicate with
          Web servers</dd>
   </div>
</dl>
```

definition list code

bandwidth	A measure of data transfer speed over a network
HTTP	The protocol used to communicate with Web servers

definition list displayed as table

```
dl {
 display: table;
 border-collapse: collapse;
 width: 300px;
}
dl div {
 display: table-row;
}
dt, dd {
 display: table-cell;
 border: 1px solid black;
 vertical-align: top;
 padding: 5px;
}
```

table styles

© 2016 Cengage Learning

Tables and Responsive Design

- Tables do not scale well to mobile devices
- Problems faced by users to view a table in a mobile device
  - Table is too small to read
  - Table does not fit the visual viewport
  - Table columns are too narrow to read the cell content

Tables and Responsive Design (continued 1)

Figure 6-37 Web tables on mobile devices

complete table is too small to read

table is easier to read but does not fit within the viewport

table fits within the viewport but columns are too narrow to read comfortably

BenBois/openclipart

Tables and Responsive Design (continued 2)

- A new layout of table data for mobile screens is required
- Several table columns are reduced to two:
  - One column containing all data labels
  - Second column containing data associated with each label

Tables and Responsive Design (continued 3)

- To create a responsive web table, add the text of data labels as attributes of all `td` elements in the table body
- Store data labels using a **data attribute**
- General format of a data attribute is

  `data-`*`text`*`="`*`value`*`"`

  where *`text`* is the `name` of the data attribute and *`value`* is its value

Tables and Responsive Design (continued 4)

- Data attributes use names specific to the function it is used for
- For example, the following code uses a data attribute named `data-label` to store the text of the labels associated with the data cell:

```
<td data-label="Date">April 2, 2017</td>
```

Tables and Responsive Design (continued 5)

- The result is a list of data cells that are aligned as block elements
- Within each block element, the data label is followed by the data cell content
- The goal is to transform table with multiple columns into two-column layout

Tables and Responsive Design (continued 6)

Figure 6-38 Two-column layout for a mobile device

Designing a Column Layout

- Column layout enables display of content side-by-side in a page
- Layouts that use float elements or flexboxes differ from column layout
  - Single element can flow from one column to the next
  - Flow of content adjusts to match the page width

Setting the Number of Columns

- Size of a column is set using the `column-count` property

  `column-count: value;`

  where *`value`* is the number of columns in the layout

- Browser extensions are included to ensure cross-browser compatibility

Setting the Number of Columns (continued)

Figure 6-40 Applying a 2-column layout to the article element

applies the column style only to screen devices wider than 640 pixels

uses browser extensions to ensure compatibility across browsers

```
/* Column Styles */

@media only screen and (min-width: 641px) {
   article {
      -moz-column-count: 2;
      -webkit-column-count: 2;
      column-count: 2;
   }
}
```

displays the article content across 2 columns

Defining Column Widths and Gaps

- Columns are laid out evenly across the width of the parent element by default
- To set the column width, use the `column-width` property

  `column-width: size;`

  where *`size`* is the minimum width of the column
- Column width acts like the basis value for items in a flexbox

Defining Column Widths and Gaps (continued 1)

- The `column-width` and `column-count` properties are combined to form shorthand `columns` property

  `columns: width count;`

- The default gap between columns is 1em
- To set a different gap size, use the `column-gap` property

  `column-gap: size;`

where *`size`* is the width of the gap

Defining Column Widths and Gaps (continued 2)

- Another way to separate columns is with a graphic dividing line created using the `column-rule` property

  `column-rule: border;`

  where *`border`* defines the style of dividing line

- The `column-rule` property can be broken into individual properties like `column-rule-width`, `column-rule-style`, and `column-rule-color`

Defining Column Widths and Gaps (continued 3)

Figure 6-42 Setting the size of the column gap

```
@media only screen and (min-width: 641px) {
   article {
      -moz-column-count: 2;
      -webkit-column-count: 2;
      column-count: 2;

      -moz-column-gap: 30px;
      -webkit-column-gap: 30px;
      column-gap: 30px;
   }
}
```

sizes the size of the gap between the columns to 30 pixels

Managing Column Breaks

- The size of column orphans is controlled using the `orphans` property

  `orphans: value;`

  where *`value`* is the minimum number of lines stranded before a column break

- The size of column widows is controlled using the `widows` property

  `widows: value;`

  where *`value`* is the minimum number of lines placed after a column break

Managing Column Breaks (continued 1)

- Other properties to define column breaks

  `break-before: type;`

  `break-after: type;`

  where *`type`* is one of the following:

  - `auto` (browser automatically sets column break)
  - `always` (to always place a column break)
  - `avoid` (to avoid placing a column break)

Managing Column Breaks (continued 2)

- To control placement of column breaks within an element, use the property

  `break-inside: type;`

  where *type* is either `auto` or `avoid`

Managing Column Breaks (continued 3)

Figure 6-45 Defining widows and orphans around column breaks

keeps at least three lines together after the column break

keeps at least three lines together before the column break

```
    -moz-column-rule: 2px solid gray;
    -webkit-column-rule: 2px solid gray;
    column-rule: 2px solid gray;

    widows: 3;
    orphans: 3;
}
```

Spanning Cell Columns

- To span cell columns, use the `column-span` property

  `column-span: span;`

  where *`span`* is either `none` to prevent spanning or `all` to enable the content to span across all the columns

Spanning Cell Columns (continued)

Figure 6-46 Creating a column-spanning heading

```
        widows: 3;
        orphans: 3;
    }

    article h1 {
        -moz-column-span: all;
        -webkit-column-span: all;
        column-span: all;
    }

}
```

sets the heading so that it extends across all columns