

# Building the User Interface by Using HTML5: Organization, Input, and Validation

# Agenda

- Semantic markup
- `div` element
- header and footer elements
- section element
- nav element
- article element
- aside element
- HTML tables
- Ordered and unordered lists
- Forms and input
- Validation

# Semantic Markup

- Tag names that are intuitive
- Makes it easier to build and modify HTML documents
- Makes it easier for Web browsers and other programs to interpret
- Developers can still use `<div>` in HTML5 documents; should use new structure elements whenever appropriate

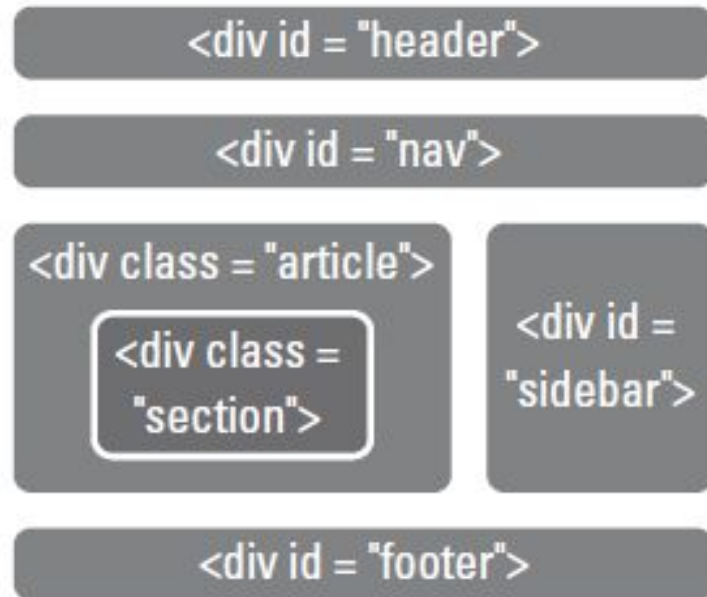
# div Element

- Used for years to create structure of an HTML document
- Often includes a class or ID attribute
- May include CSS styles such as background-color, height, and width
- Example:
  - `<div id="header" > This is a header </div>`

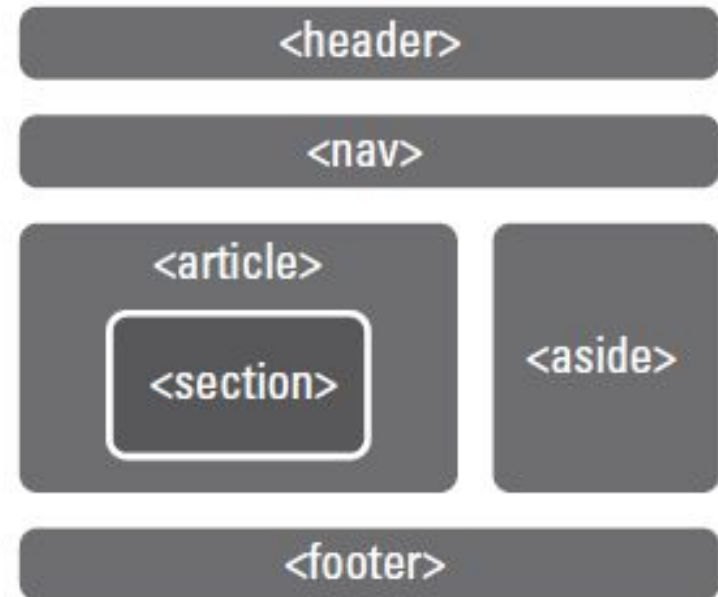
# New HTML5 Elements for Structuring and Organizing Content

- header, footer, section, nav, article, and aside

## HTML 4.01



## HTML 5

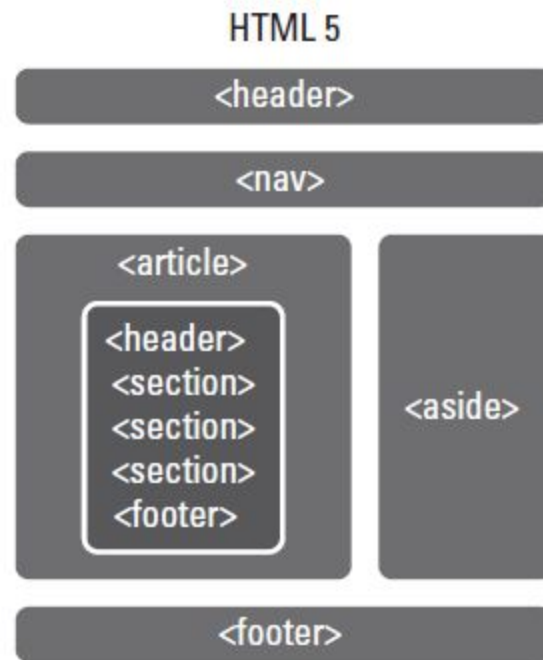


# header and footer Elements

- The `header` element defines a header for a document, section, or article. HTML 4.01 uses the header div (`<div id="header">`).
- The `footer` element defines a footer for a document or section, and typically contains information about the document or section, such as the author name, copyright data, links to related documents, and so on.

# header and footer Elements (Continued)

- Can include multiple headers or footers in an HTML5 document



# header and footer Markup Example

```
<article>
  <header>
    <h1>Learning HTML5</h1>
  </header>
  <p>Here comes my text</p>
  <footer>
    <p>Published <time datetime="2016-03-19">March
      19, 2016</time></p>
  </footer>
</article>
```

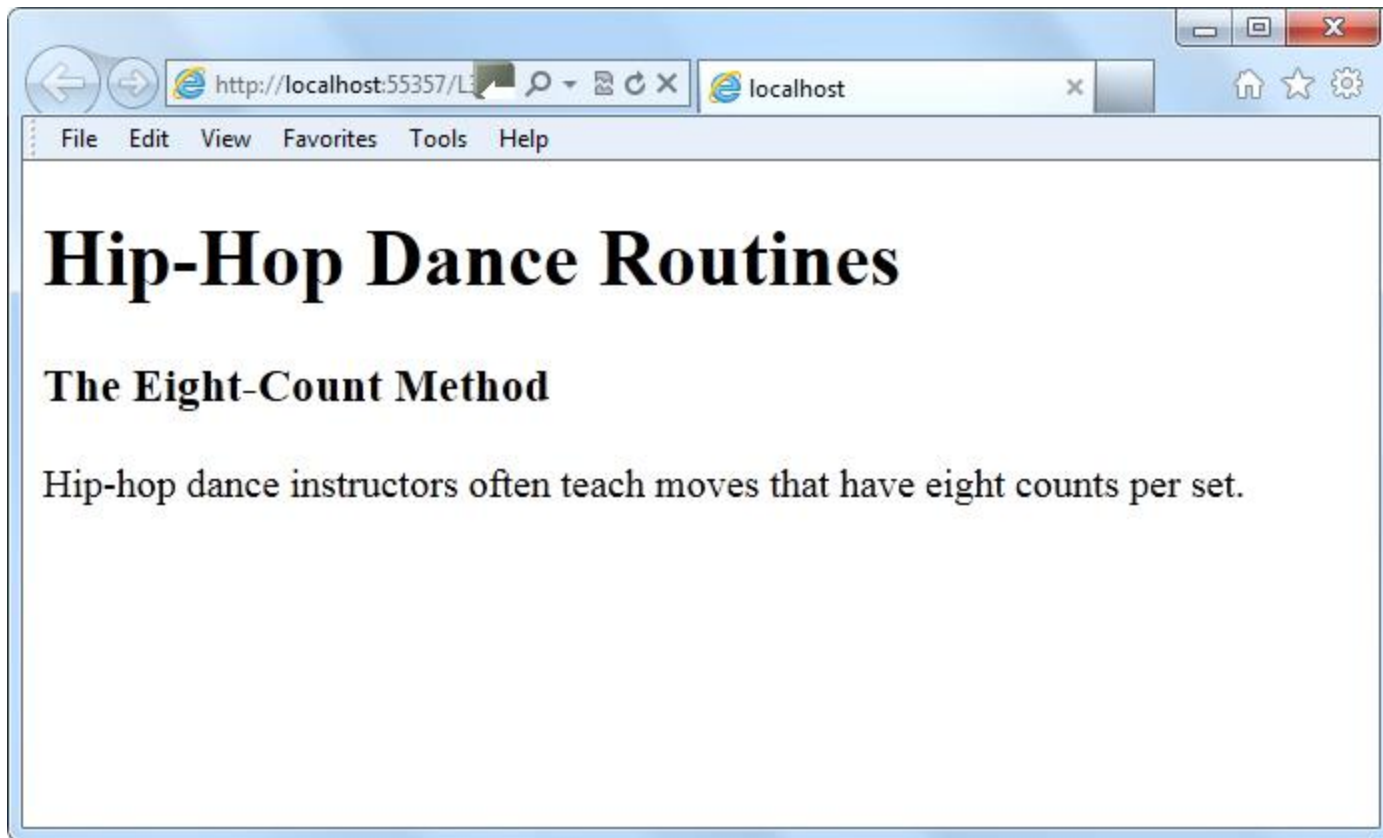
<https://jsfiddle.net/koldovsky/qb664wjy/>



# section Element

- Defines a section in a document, such as a chapter, parts of a thesis, or parts of a Web page whose content is distinct from each other
- According to the W3C, must contain at least one heading and define something that would ordinarily appear in the document's outline

# section Example



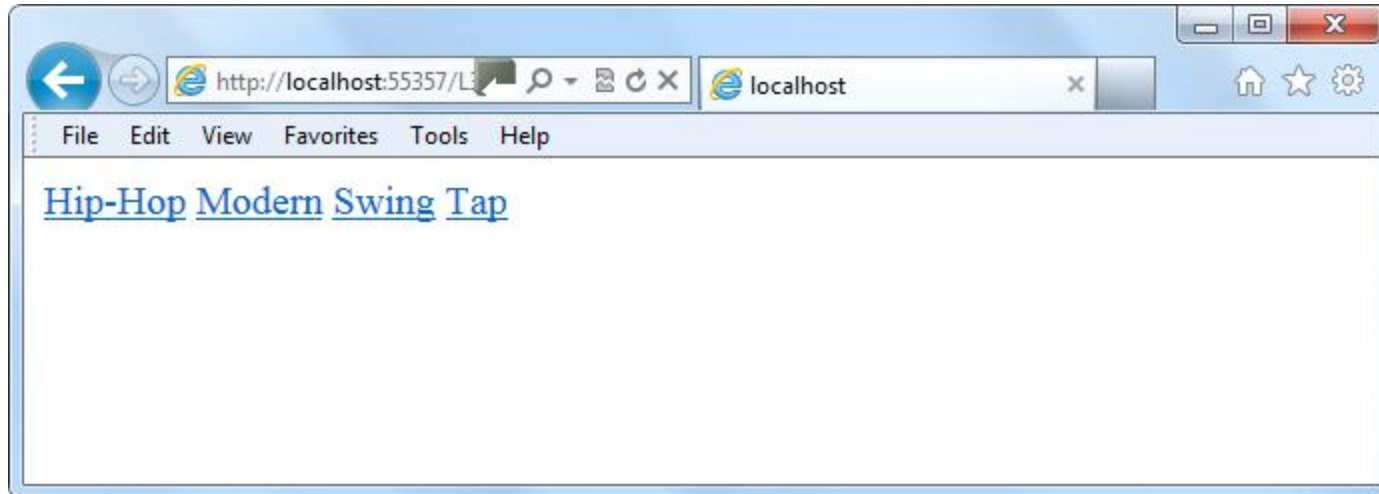
# nav Element

- Defines a block of navigation links and is useful for creating
  - A set of navigation links as a document's primary navigation
  - A table of contents
  - Breadcrumbs in a footer
  - Previous-Home-Next links

# nav Example

```
<nav>  
  <a href="/hiphop/">Hip-Hop</a>  
  <a href="/modern/">Modern</a>  
  <a href="/swing/">Swing</a>  
  <a href="/tap/">Tap</a>  
</nav>
```

# nav Example



# article Element

- Defines a part of an HTML document that consists of a “self-contained composition” independent from the rest of the content in the document
- Content set off by `<article>` tags can be distributed in syndication
  - Think of it as content that makes sense on its own

# aside Element

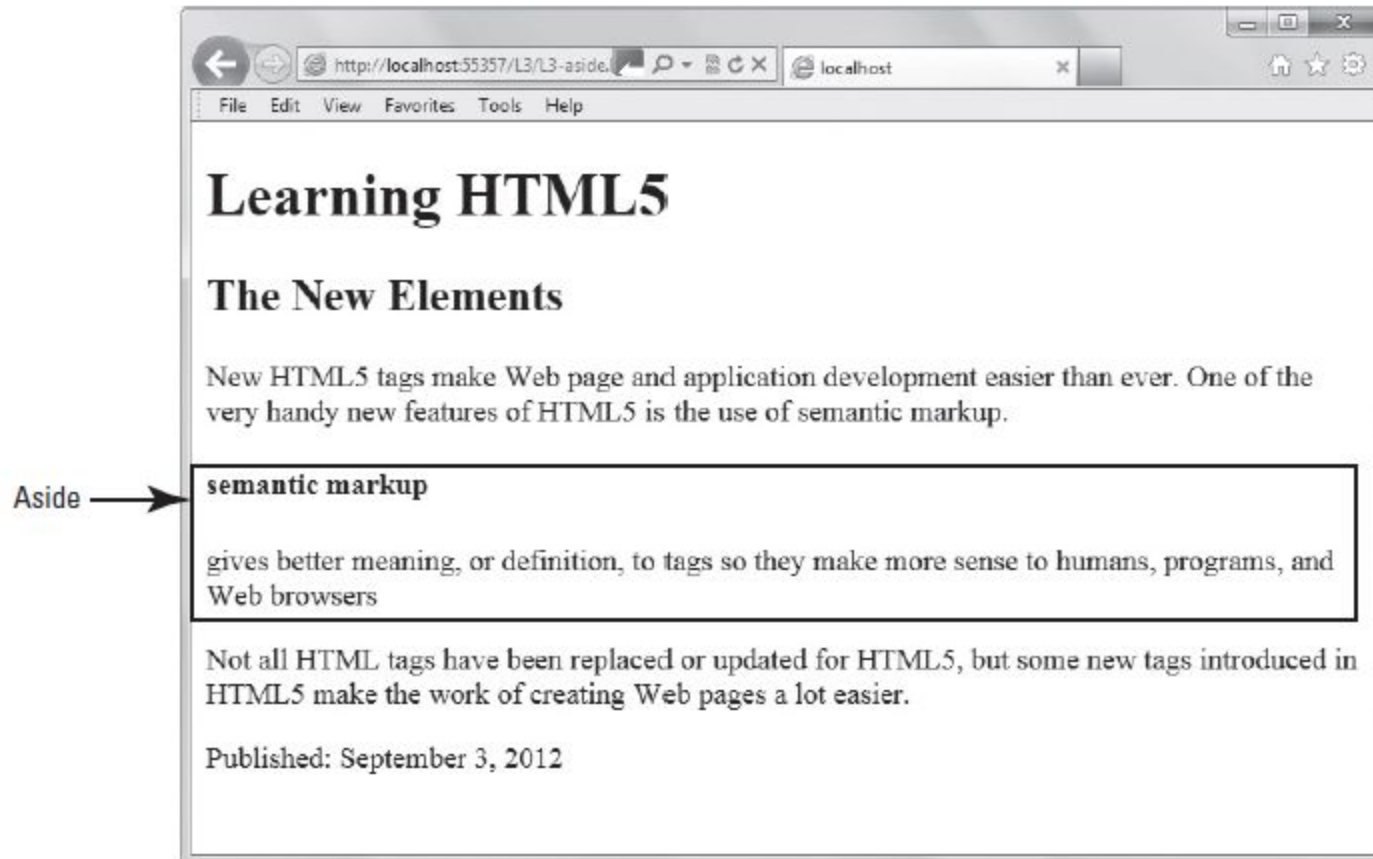
- Used for sidebars and notes—content that's related to the current topic but would interrupt the flow of the document if left inline

# aside Example

```
<article>
  <header>
    <h1>Learning HTML5</h1>
    <h2>The New Elements</h2>
  </header>
  <p>New HTML5 tags make Web page and application
  development easier than ever. One of the very
  handy new features of HTML5 is the use of
  semantic markup.</p>
  <aside>
    <h4><b>semantic markup</b></h4>
    <p> gives better meaning, or definition,
    to tags so they make more sense to humans,
    programs, and Web browsers</p>
  </aside>
  <p>Not all HTML tags have been replaced or updated
  for HTML5, but some new tags introduced in HTML5
  make the work of creating Web pages a lot
  easier.</p>
</footer>
  <p>Published: <time datetime="2012-09-
  03">September 3, 2012</time></p>
</footer>
</article>
```



# aside Example



# section, article, aside



# HTML Tables

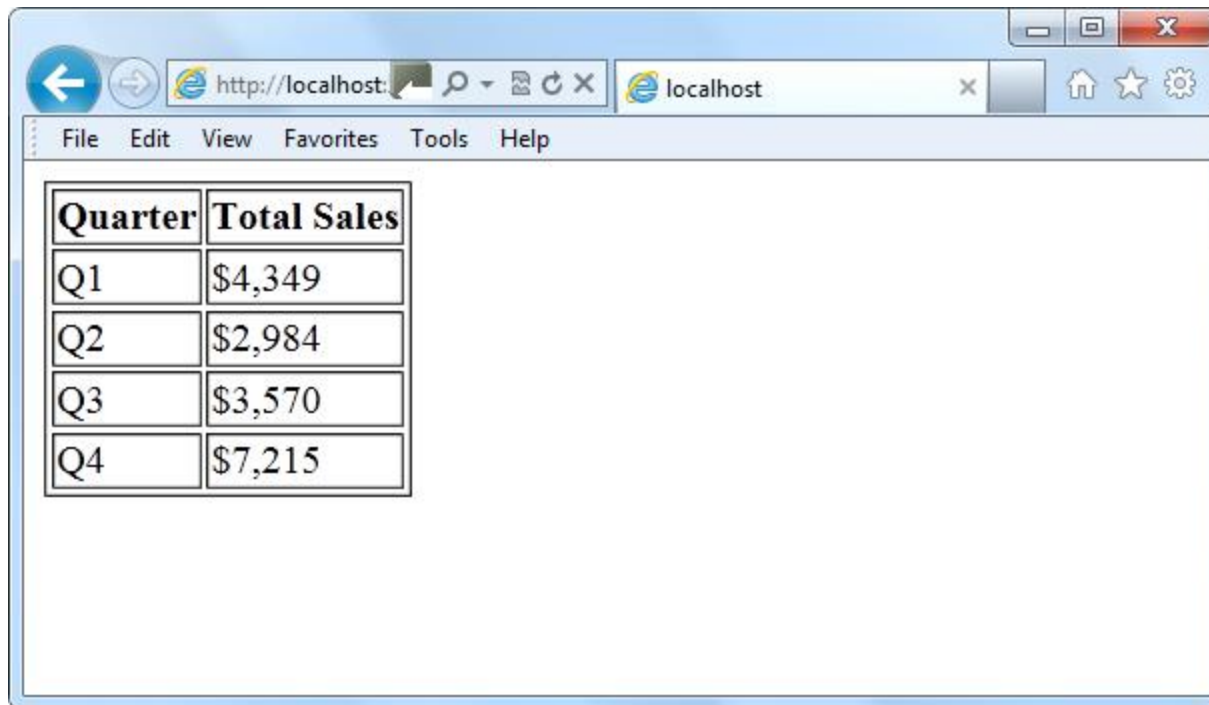
- `<table>` defines overall table
- `<tr>` defines rows
- `<th>` defines column headers
- `<td>` defines cells
- `<caption>` adds a caption above or below table
- `<col>` applies inline CSS styles
- Long, scrolling tables use `<thead>`, `<tfoot>`, and `<tbody>` tags

# HTML Table Example

```
<table border="1">
  <tr> <!--first row-->
    <th>Quarter</th> <!--first column in first row-->
    <th>Total Sales</th> <!--first row, second column-->
  </tr>
  <tr> <!--second row-->
    <td>Q1</td>
    <td>$4,349</td>
  </tr>

  <tr> <!--third row-->
    <td>Q2</td>
    <td>$2,984</td>
  </tr>
  <tr> <!--fourth row-->
    <td>Q3</td>
    <td>$3,570</td>
  </tr>
  <tr> <!--fifth row-->
    <td>Q4</td>
    <td>$7,215</td>
  </tr>
</table>
```

# HTML Table Example



A screenshot of a web browser window showing an HTML table. The browser's address bar displays 'http://localhost:...' and the page title is 'localhost'. The menu bar includes 'File', 'Edit', 'View', 'Favorites', 'Tools', and 'Help'. The table has two columns: 'Quarter' and 'Total Sales'. The data rows are: Q1 (\$4,349), Q2 (\$2,984), Q3 (\$3,570), and Q4 (\$7,215).

Quarter	Total Sales
Q1	\$4,349
Q2	\$2,984
Q3	\$3,570
Q4	\$7,215

# HTML colspan, rowspan

	<b>I</b>	<b>II</b>	<b>III</b>	<b>IV</b>
	1	2	3	4

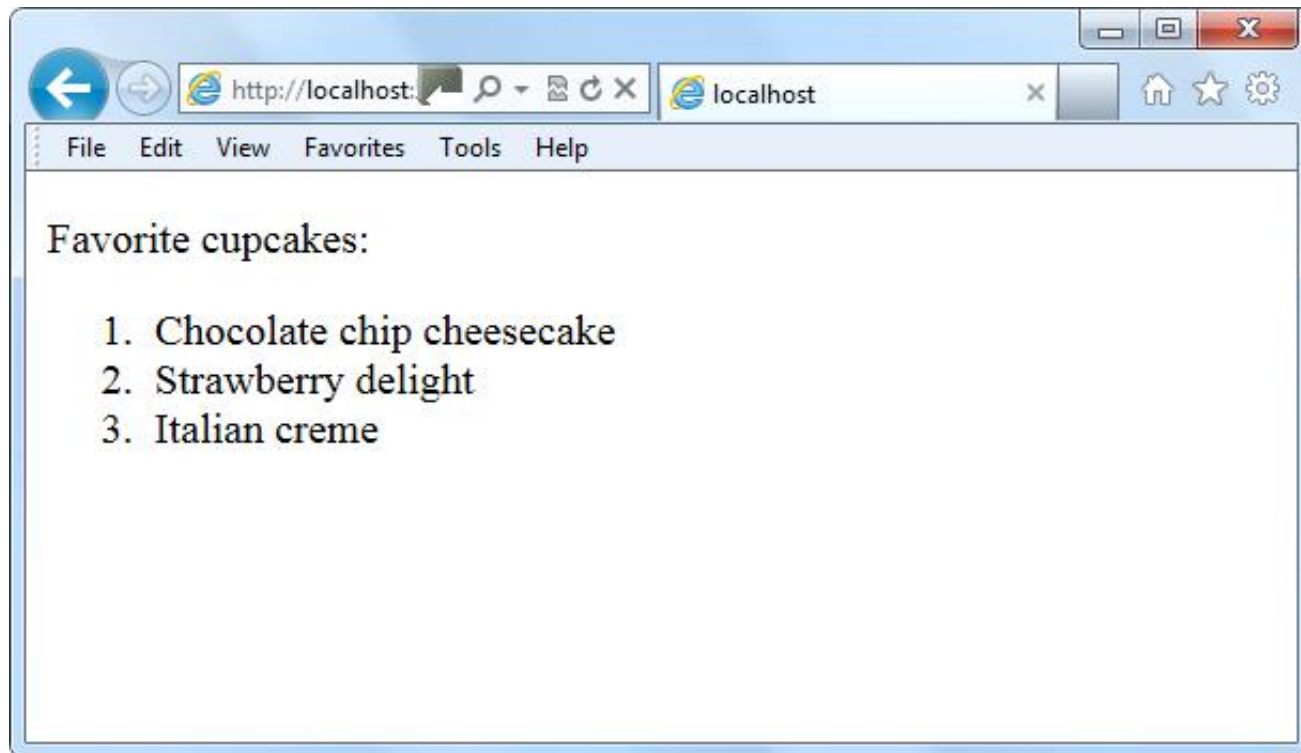
```
<table>
  <thead>
    <tr>
      <th rowspan="2"></th>
      <th colspan="4">&nbsp;</th>
    </tr>
    <tr>
      <th>I</th>
      <th>II</th>
      <th>III</th>
      <th>IIII</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td></td>
      <td>1</td>
      <td>2</td>
      <td>3</td>
      <td>4</td>
    </tr>
  </tbody>
</table>
```

# Ordered List

- Orders list entries using numbers, by default
- Uses the `<ol>` tag with attributes:
  - `reversed`: Reverses the order of the list
  - `start number`: Specifies the start value of the ordered list
  - `type`: Specifies list item marker, such as "1" for displaying decimal numbers

# Ordered List Example

```
<p>Favorite cupcakes:</p>
<ol>
  <li>Chocolate chip cheesecake</li>
  <li>Strawberry delight</li>
  <li>Italian creme</li>
</ol>
```



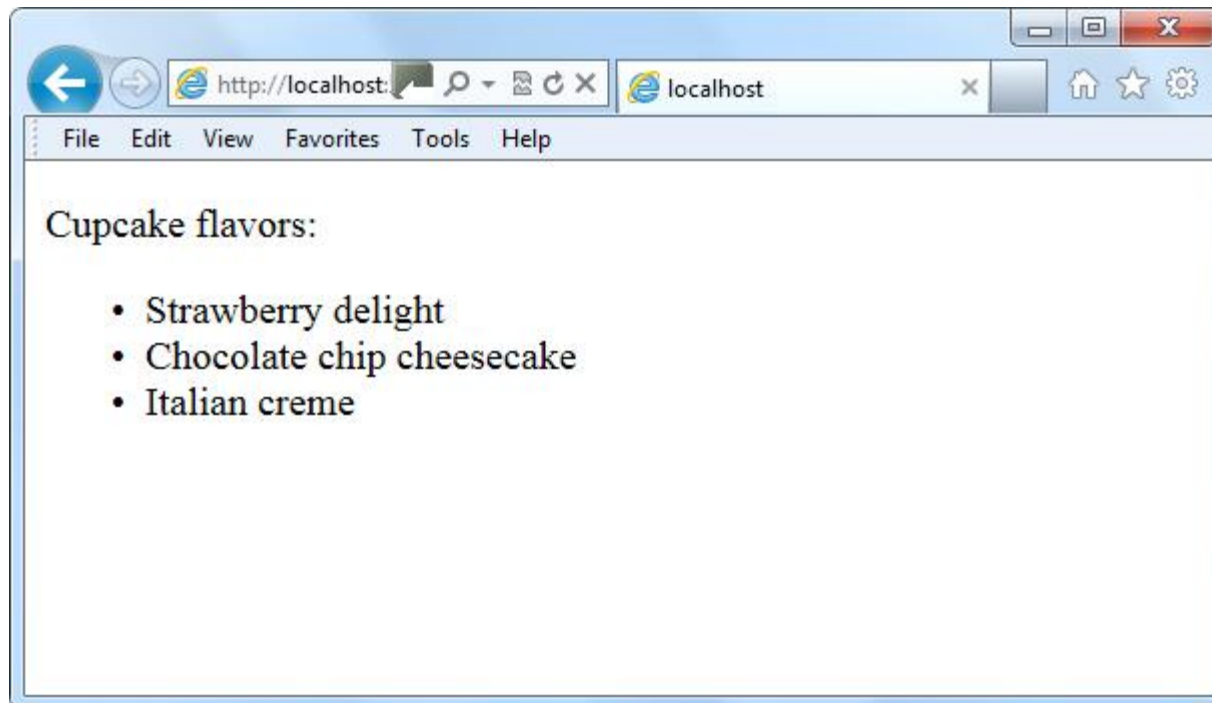


# Unordered List

- Displays list entries in a bulleted list
- Uses a `<ul>` tag
- Round bullet symbol is the default marker for list items
- Can change bullet symbols
  - For squares, add `type="square"` to the `<ul>` tag
  - For empty circles, add `type="circle"`

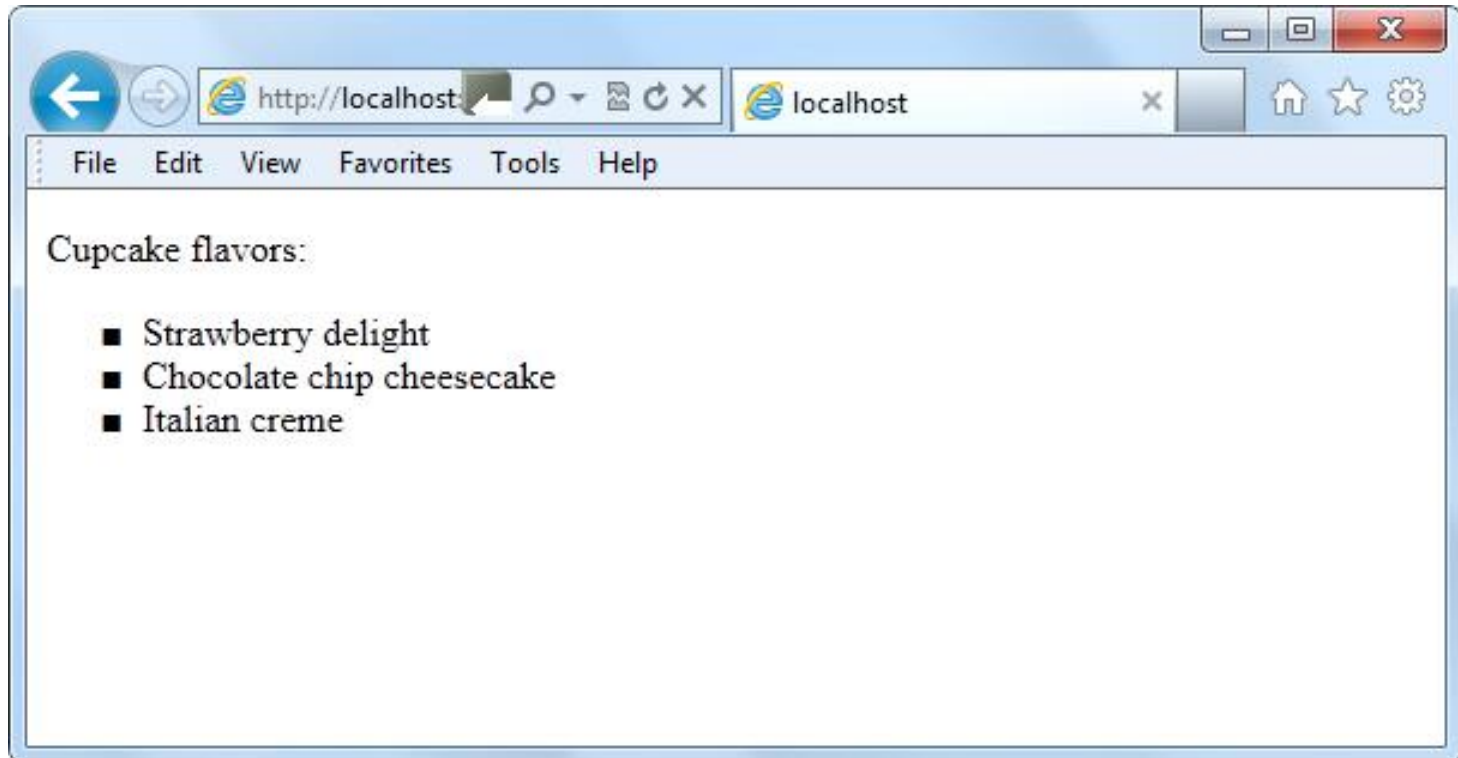
# Unordered List Example

```
<p>Cupcake flavors:</p>
<ul>
  <li>Strawberry delight</li>
  <li>Chocolate chip cheesecake</li>
  <li>Italian creme</li>
</ul>
```



# Unordered List Example

```
<p>Cupcake flavors:</p>
<ul type="square">
  <li>Strawberry delight</li>
  <li>Chocolate chip cheesecake</li>
  <li>Italian creme</li>
</ul>
```



# Forms and Input

- Form input is the information a user enters into fields in a Web or client application form.
- HTML5 introduces several new form and input element attributes; some are:
  - `url` for entering a single Web address
  - `email` for a single email address or a list of email addresses
  - `search` to prompt users to enter text they want to search for

# Creating a Form

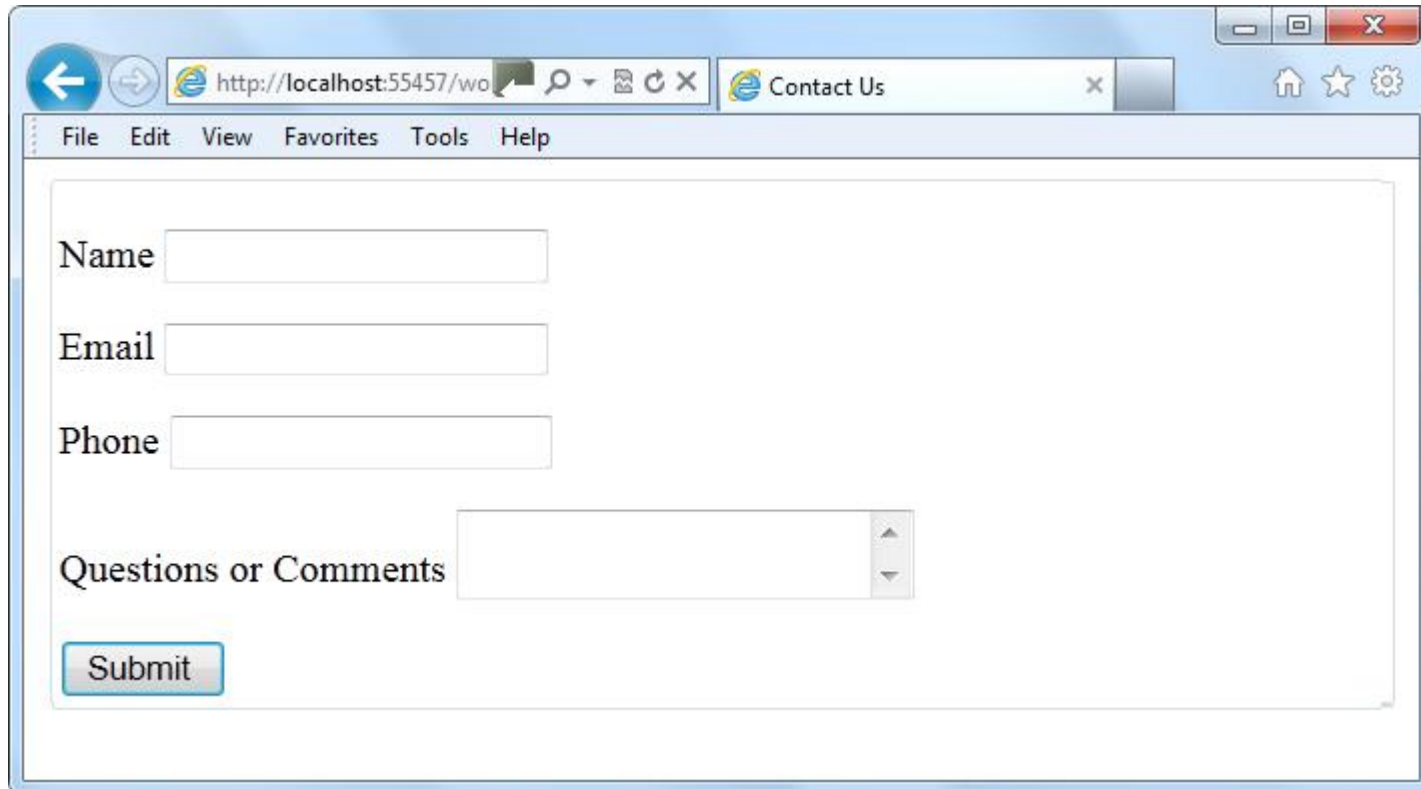
- Use the `<form>` start and end tags
- All form content and fields are between `<form>` tags
- Common syntax:

```
<form id="keyword">  
<content and fields>  
</form>
```

# Creating a Form (Continued)

- The `fieldset` element is used with many forms to group related elements.
- The `<fieldset>` tag draws a box around individual elements and/or around the entire form.

# Form with Fieldset Example



The image shows a screenshot of a web browser window. The address bar displays the URL `http://localhost:55457/wo` and the page title is "Contact Us". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The main content area contains a form enclosed in a fieldset. The form has four input fields: "Name", "Email", "Phone", and "Questions or Comments". The "Questions or Comments" field is a text area with a vertical scrollbar. A "Submit" button is located at the bottom left of the form.

Browser address bar: `http://localhost:55457/wo` Contact Us

Menu bar: File Edit View Favorites Tools Help

Form fields:

- Name
- Email
- Phone
- Questions or Comments

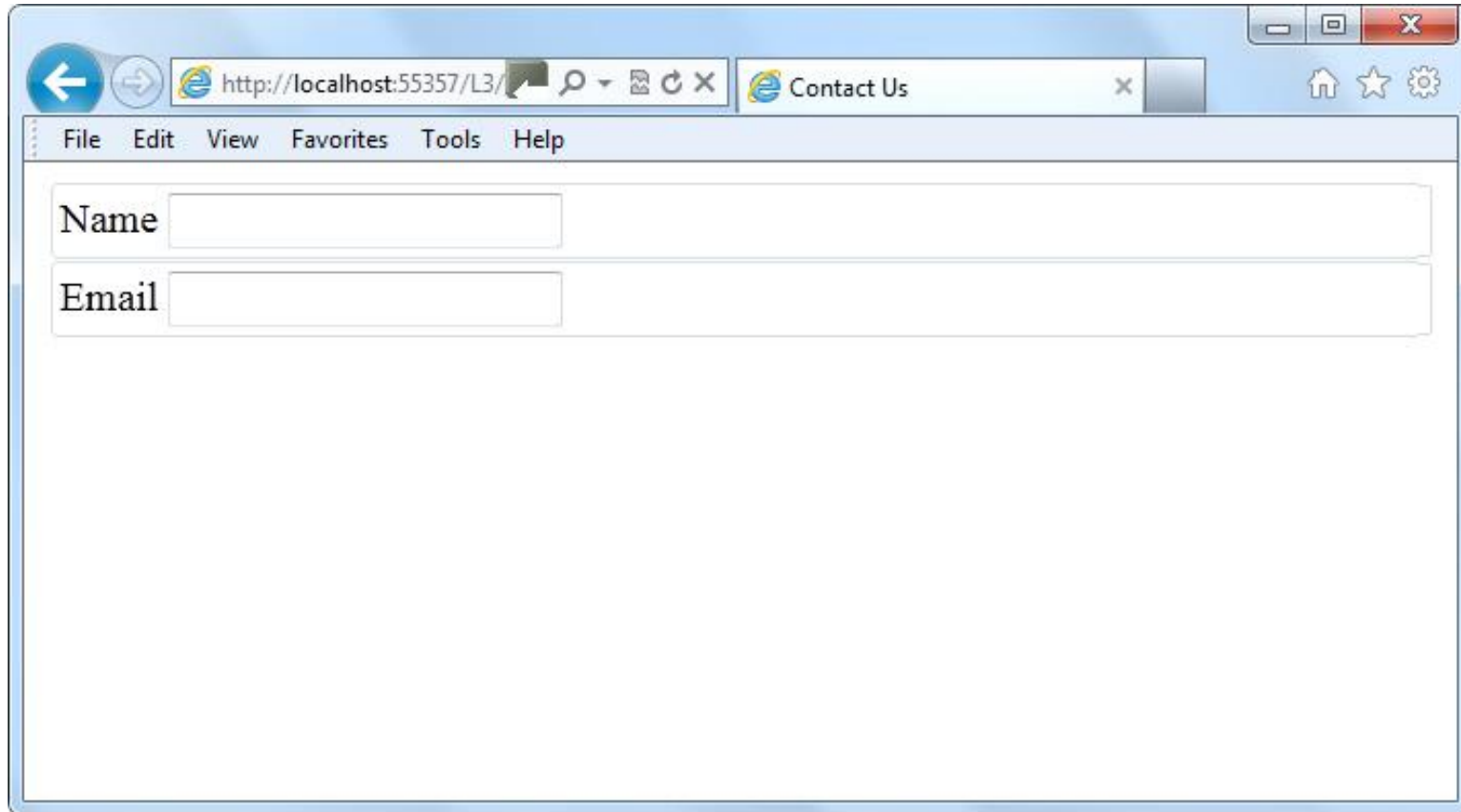
Submit

# Simple Form Example

```
<div id="contact-form"
  style="font-family:'Arial Narrow','Nimbus Sans
  L',sans-serif;">
  <form id="contact" method="post" action="">
    <fieldset>
      <label for="name">Name</label>
      <input type="text" name="name" />
    </fieldset>
    <fieldset>
      <label for="email">Email</label>
      <input type="email" name="email" />
    </fieldset>
  </form>
</div> <!-- end of contact-form -->
```



# Simple Form Example



A screenshot of a web browser window displaying a simple contact form. The browser's address bar shows the URL `http://localhost:55357/L3/` and the page title is "Contact Us". The menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The form consists of two input fields: "Name" and "Email", each with a corresponding text box. The "Name" field is positioned above the "Email" field. The browser window also shows standard navigation icons (back, forward, search, refresh, stop) and window control buttons (minimize, maximize, close).

# Form `required` and `email` Attributes

- The ***required attribute*** requires information in a field when the form is submitted.
- The ***email attribute*** requires the user to enter an email address.
- Markup example:

```
<input type="email" required />
```

# required Example

Name

E-mail

Phone

Web site

Please fill out this field.

# Form placeholder Attribute

- **Placeholder text** is text displayed inside an input field when the field is empty. It helps users understand the type of information they should enter or select. When you click on or tab to the input field and start typing, the newly entered text replaces the placeholder text.

- Markup example:

```
<input name="fName" placeholder="First Name" />
```

# Form pattern Attribute

- The *pattern attribute* provides a format (a regular expression) for an input field, which is used to validate whatever is entered into the field.
- Markup example:

```
<input type="text" id="empID"  
name="EmployeeID" required  
pattern="[A-Z]{2}[0-9]{4}"  
title="Employee ID is two capital  
letters followed by four digits">
```

# Form `pattern` Attribute (Continued)

- You can use the `pattern` attribute with these `<input>` types:
  - `text`
  - `search`
  - `url`
  - `telephone`
  - `email`
  - `password`

# Form autofocus Attribute

- The ***autofocus attribute*** moves the focus to a particular input field when a Web page loads.
- Markup example:

```
<input type="text" name="fname"  
autofocus="autofocus" />
```

# Validation

- The process of verifying that information entered or captured in a form is in the correct format and usable before sending the data to the server
- Some things verified during validation:
  - Required fields are empty
  - Email addresses are valid
  - Dates are valid
  - Text does not appear in a numeric field or vice versa



# Validation (Continued)

- Automatic validation of input means the browser checks the data the user inputs.
  - Also referred to as **client-side validation**
- **Server-side validation** occurs when server validates data received from an input form

# Validation Example

```
<input type= "text" name= "zipcode"  
  pattern= "[0-9] {5}"  
  title= "Five-digit zip code" />
```

Zip Code

Please match the requested format: Five-digit.

# Practice Task

# Contacts

## Europe Headquarters

52 V. Velykoho Str.  
Lviv 79053, Ukraine

Tel: +380-32-240-9090

Fax: +380-32-240-9080

E-mail: [info@softserveinc.com](mailto:info@softserveinc.com)

Website: [www.softserveinc.com](http://www.softserveinc.com)

## US Headquarters

12800 University Drive, Suite 250  
Fort Myers, FL 33907, USA

Tel: 239-690-3111

Fax: 239-690-3116

# Thank You!