

HTML Basics: Part 1

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.



For customized training related to JavaScript or Java, email hall@coreservlets.com
Marty is also available for consulting and development support

Taught by lead author of *Core Servlets & JSP*,
co-author of *Core JSF* (4th Ed), and this tutorial.

Available at public venues, or
custom versions can be held on-site at your organization.

- **Courses developed and taught by Marty Hall**
 - JavaScript, jQuery, Ext JS, JSF 2.3, PrimeFaces, Java 8 programming, Spring Framework, Spring MVC, Android, GWT, custom mix of topics
 - Courses available in any state or country.
 - Maryland/DC companies can also choose afternoon/evening courses.
- **Courses developed and taught by coreservlets.com experts (edited by Marty)**
 - Hadoop, Hibernate/JPA, HTML5, RESTful Web Services

Contact hall@coreservlets.com for details



Topics in This Section

- Differences among HTML versions
- Basic structure of an HTML 5 document
- Hypertext links and URLs
- Block-level elements: overview
- Headings and basic paragraphs
- The div and span elements
- Lists
- Inline elements

4

coreservlets.com – custom onsite training



HTML Versions

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Official Rules

- **XHTML**

- Tags must be in lowercase
 - `<p>...</p>`, not `<P>...</P>`
- All tags require end tags
 - `...` or `
`
- Attribute values always need quotes
 - `<table border="1">...</table>`

- **HTML 4 and HTML 5**

- Tags can be in any case
 - `<p>` or `<P>`, `<table>` or `<tAbLe>`
- Some end tags are optional
 - `p`, `li`, `tr`, `th`, `td`, `hr`, `br`, `img`, etc.
- Attribute values do not need quotes unless they have special characters
 - `<table border=1>...</table>`

6

Approach You Should Follow

- **Use HTML 5 but follow general rules of XHTML**

- Use lowercase tags
- Include end tags
- Put quotes around attribute values
- Don't feel guilty about using a few non-CSS formatting capabilities
 - CSS usually better, but not always: `<table border="1">` hard to do with CSS.

- **Motivation**

- HTML 5 has simpler DOCTYPE, `<html>`, `<script>`, `<link>`
- HTML 5 has a few extra capabilities
 - `<input type="date"/>` and a few more
- By following rules of XML, the code you write corresponds directly to what the browser stores internally. So, it is easier to visualize how to use JavaScript to modify the HTML.

7

HTML 5 Document Format

```
<!DOCTYPE html>
<html>
<head>
<title>...</title>
<link href="css/some-stylesheet.css"
      rel="stylesheet"/>
<script src="scripts/some-script.js"></script>
</head>
<body>
...
</body>
</html>
```

8

coreservlets.com – custom onsite training



Hypertext Links and URLs

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Hypertext Links

- **Usage**

- Links can contain images and other inline elements (i.e., `<a href...> ... `)

- **Link to absolute URL**

- Use a complete URL beginning with `http://`
JavaScript is discussed in
`Chapter 2.`

- **Link to relative URL**

- Use a filename or relative path to filename
 - If no slash, interpreted with respect to location of current fileJavaScript is discussed in
`Chapter 2.`

10

Hypertext Links (Continued)

- **Link to section**

- Use a section name (see below) preceded by `#`
Images are discussed in
`Section 2.`

- **Link to section in URL**

- Use absolute or relative URL, then `#`, then section name
Images are discussed in
`Section 2 of Chapter 1.`

- **Naming a Section**

- Use the `id` attribute and do not include the pound sign
`<h2 id="section2">Images</h2>`

11

Interpreting Relative URLs

- **URLs starting with `http://` (absolute URLs)**
 - Interpreted independently of the URL of current page
- **URLs starting with `.` (or anything other than the other options)**
 - Interpreted with respect to the “directory” of current file
 - Really with respect to the rightmost `/` in the URL, since URLs don’t necessarily refer to real folders or files
- **URLs starting with `..`**
 - Interpreted with respect to the parent directory of current file
 - Again, based on the URL browser sees, not necessarily real location of resource
- **URLs starting with `/`**
 - Interpreted with respect to the hostname of the URL

12

Interpreting Relative URLs (Examples)

- **Assume URL of current page is `http://somehost.com/a/b/c/d/e/f.html`**

Link	Result
<code>...</code>	<code>http://otherhost.com/a/b.html</code>
<code>...</code> <u>or</u> <code>...</code>	<code>http://somehost.com/a/b/c/d/e/g.html</code>
<code>...</code>	<code>http://somehost.com/a/b/c/d/h.html</code>
<code>...</code>	<code>http://somehost.com/i.html</code>

Note: In addition to `<a href...>`, these URLs can apply to `img`, `form`, `script`, `style`, and a few other elements.

Overview of Block-Level Elements (Paragraph-Like Elements)

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Block-Level Elements

- **Headings**
 - h1 ... h6
 - align attribute
- **Basic Text Sections**
 - p
 - pre
- **Tables**
 - Covered in next lecture
- **Forms**
 - Covered in next lecture
- **Lists**
 - ol
 - li
 - ul
 - li
 - dl
 - dt
 - dd
- **Misc.**
 - hr
 - div
 - center

Shared Attributes

- **id**

- A *unique* identifier, usually used so that JavaScript (usually via jQuery) can reference the tag later, but also used (as seen earlier) for internal hypertext links.

```
<div id="placeholder-for-ajax"></div>
<input id="firstNameField" name="..." />
```

- **class**

- A stylesheet class name
- This will have no effect unless the “warning” name is defined in your CSS file.
 - See later lecture on CSS.

16

Shared Attributes (Continued)

- **style**

- An inline style

```
<p style="color: red; font-size: 18px;">Warning!</p>
```

- **JavaScript event handlers**

- onclick, onchange, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onkeypress, onkeydown, onkeyup
- These are used so that JavaScript can be invoked when the user takes certain actions

```
<input type="button" value="Click Me"
      onclick="doSomethingCool()" />
```

- Lots more detail in the jQuery sections

17

Headings and Basic Paragraphs

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Headings

- **Heading Types**

- `<h1 ...> ... </h1>`
- `<h2 ...> ... </h2>`
- `<h3 ...> ... </h3>`
- `<h4 ...> ... </h4>`
- `<h5 ...> ... </h5>`
- `<h6 ...> ... </h6>`

- **Attribute: align**

- Values: left (default), right, center
 - But often better to do alignment with CSS, not explicitly with this attribute

- **Nesting tags**

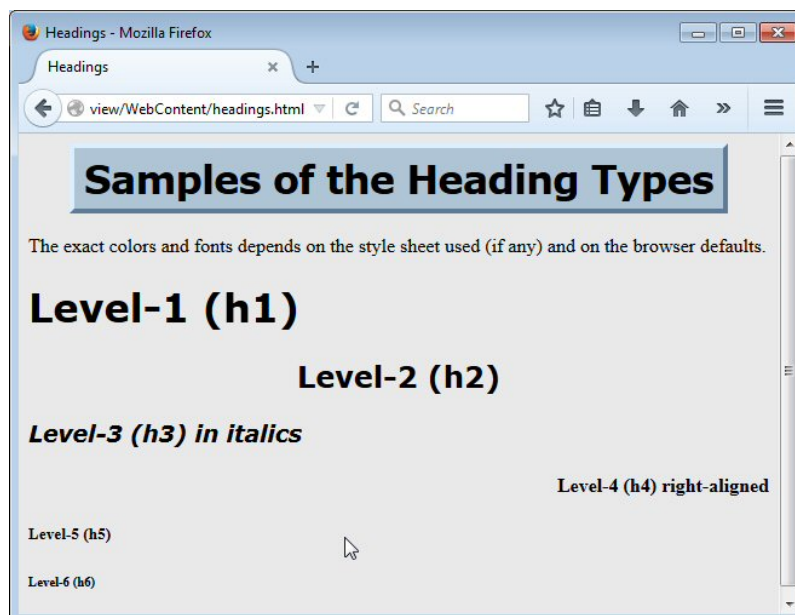
- Headings and other block-level elements can contain inline (text-level) elements, but *not* vice versa

Headings: Example

```
<!DOCTYPE html>
<html>
<head><title>Headings</title>
<link href="css/styles.css" rel="stylesheet"/>
</head>
<body>
...
<h1>Level-1 (h1)</h1>
<h2 align="center">Level-2 (h2) centered</h2>
<h3><i>Level-3 (h3) in italics</i></h3>
<h4 align="right">Level-4 (h4) right-aligned</h4>
<h5>Level-5 (h5)</h5>
<h6>Level-6 (h6)</h6>
</body></html>
```

20

Headings: Result



21

p – The Basic Paragraph

- **Attributes: align**

- left (default), right, center. Same as headings.
 - Again, often better to do alignment from CSS:
`<p class="some-css-name-that-gives-alignment">...</p>`

- **Behavior**

- Whitespace ignored (use `
` if you want line breaks)
- Consecutive `<p></p>`'s do not yield multiple blank lines
 - But multiple `
`'s do
- End tags not officially required in HTML 5, but you should use them so that what you see corresponds directly to how the browser represents the page internally
 - This makes it easier to visualize how to use JavaScript/jQuery to manipulate the page

22

Use of `<p>` Tag

Preferred Style	Technically Legal
<code><body></code>	<code><BODY></code>
<code><p></code>	Paragraph 1
Paragraph 1	<code><P></code>
<code></p></code>	Paragraph 2
<code><p></code>	<code><P></code>
Paragraph 2	Paragraph 3
<code></p></code>	<code></BODY></code>
<code><p></code>	
Paragraph 3	
<code></p></code>	
<code></body></code>	

23

pre: Preformatted Paragraphs

- **Characteristics**
 - Line wrapping disabled
 - Whitespace no longer ignored
 - Fixed-width font used
- **Problem: Special Characters**

```
<pre>
if (a<b) {
    doThis();
} else {
    doThat();
}
</pre>
```

Desired Character	Text Required
<	<
>	>
&	&
"	"
Non-breaking space	

24

coreservlets.com – custom onsite training



div and span

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

div

- **Purpose**
 - A container for enclosing other elements, including block-level elements
- **Attributes**
 - align, style, class, id
- **Usage**
 - Apply alignment or styles to multiple elements that are grouped inside the div
 - Make a placeholder where Ajax results will be placed
- **div vs. span**
 - div differs from span in that span can only enclose inline elements, whereas div can enclose both block and inline elements
 - Also, span flows into the existing paragraph whereas div by default has blank line above and below

26

span and center

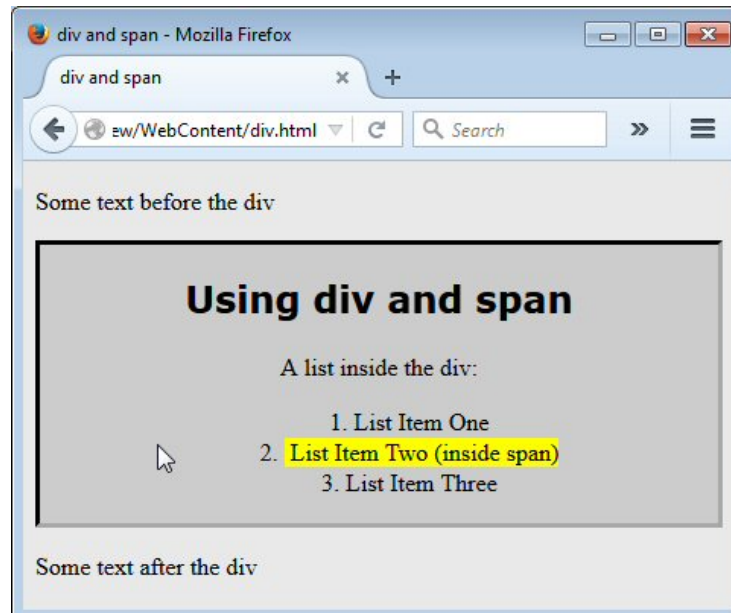
- **span**
 - Like div, used as container for other elements
 - But, span is an inline element, whereas div is block-level element. So,
 - Flows into existing paragraph
 - Cannot contain block-level elements
- **center (mostly obsolete)**
 - `<div align="center">...</div>`
 - is equivalent to
 - `<center>...</center>`
 - div is preferred in modern apps

27

div and span: Example (Code)

```
...<body>
<p>Some text before the div</p>
<div style="border: inset; background-color: #cccccc;"
  align="center">
<h2>Using div and span</h2>
<p>A list inside the div:</p>
<ol>
  <li>List Item One</li>
  <li><span style="background-color: yellow;">
    List Item Two (inside span)</span></li>
  <li>List Item Three</li>
</ol>
</div>
<p>Some text after the div</p>
</body>...
```

div and span: Example (Result)



Lists

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

ol: Ordered (Numbered) Lists

- **ol element**

```
<ol>
  <li>...</li>
  <li>...</li>
  ...
</ol>
```

- Attributes: type, start, compact

- **List items: li**

- Attributes: (When inside ol) value, type

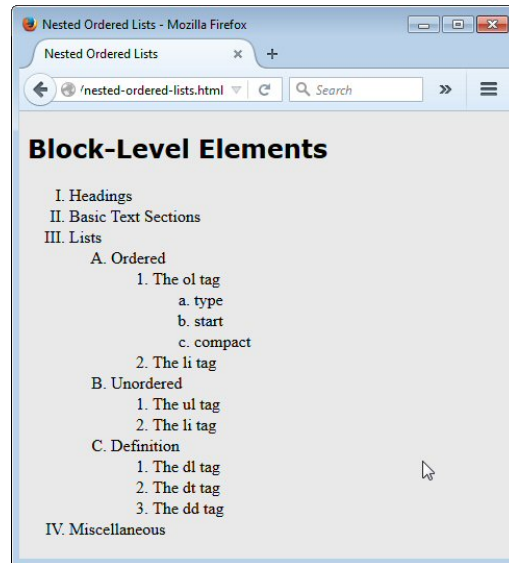
```
<p>A sample list:</p>
<ol>
  <li>List Item One</li>
  <li>List Item Two</li>
  <li>List Item Three</li>
</ol>
```

A sample list:

```
1. List Item One
2. List Item Two
3. List Item Three
```

Nested Ordered Lists

```
<ol type="I">
  <li>Headings</li>
  <li>Basic Text Sections</li>
  <li>Lists
    <ol type="A">
      <li>Ordered
        <ol type="1">
          <li>The ol tag
            <ol type="a">
              <li>type</li>
              <li>start</li>
              <li>compact</li>
            </ol></li>
          <li>The li tag</li>
        </ol></li>
      <li>Unordered ... </li>
      <li>Definition ... </li>
    </ol></li>
  <li>Miscellaneous</li>
</ol>
```



32

ul: Unordered (Bulleted) Lists

- **ul element**

```
<ul>
  <li>...</li>
  <li>...</li>
  ...
</ul>
```

- **Attributes: type, compact**

- type is disc, circle, or square

- **List items: li (type)**

- type is disc, circle, or square

```
<p>A sample list:</p>
<ul>
  <li>List Item One</li>
  <li>List Item Two</li>
  <li>List Item Three</li>
</ul>
```

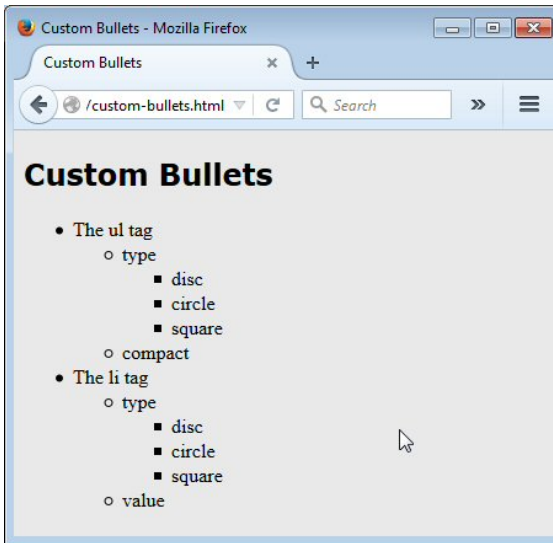
A sample list:

- List Item One
- List Item Two
- List Item Three

33

ul: Custom Bullets

```
<ul type="disc">
  <li>The ul tag
    <ul type="circle">
      <li>type
        <ul type="square">
          <li>disc</li>
          <li>circle</li>
          <li>square</li>
        </ul></li>
      <li>compact</li>
    </ul></li>
</ul></li>
<li>The li tag
  <ul type="circle">
    <li>type
      <ul type="square">
        <li>disc</li>
        <li>circle</li>
        <li>square</li>
      </ul></li>
    <li>value</li>
  </ul></li>
</ul>
```



coreservlets.com – custom onsite training



Inline (Text-Level) Elements (Font-Like Elements)

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Inline Elements

- **Physical character styles: idea**
 - Tag names describes how it should look
- **Physical character styles**
 - b, i, tt, u, sub, sup, small, big, strike, s, blink
 - font
 - size
 - color
 - face
 - basefont
- **Logical character styles: idea**
 - Tag name describes the type of content. Browser or CSS determines the look.
- **Logical character styles**
 - em, strong, code, samp, kbd, dfn, var, cite

36

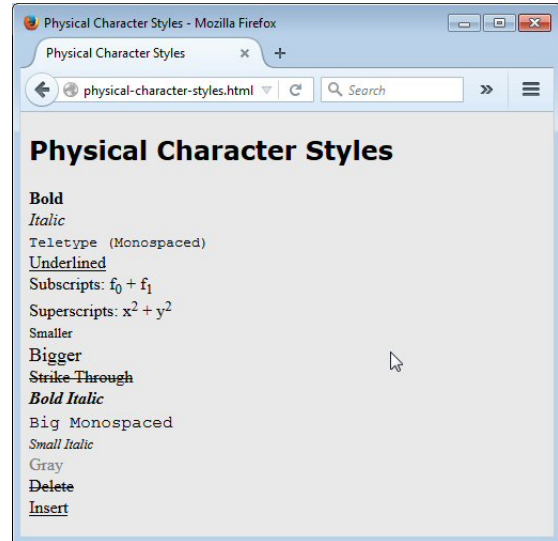
Inline Elements (Continued)

- **Hypertext Links**
 - a
 - href, name, target, ...
 - See earlier section on URLs and links
- **Images**
 - img
 - src (required), alt, align, width, height, hspace, vspace, border, usemap, ismap
- **Miscellaneous Inline Elements**
 - span (arbitrary style wrapped around text)
 - br (explicit line break)
 - area (client-side image map)
 - applet (Java)
 - ...

37

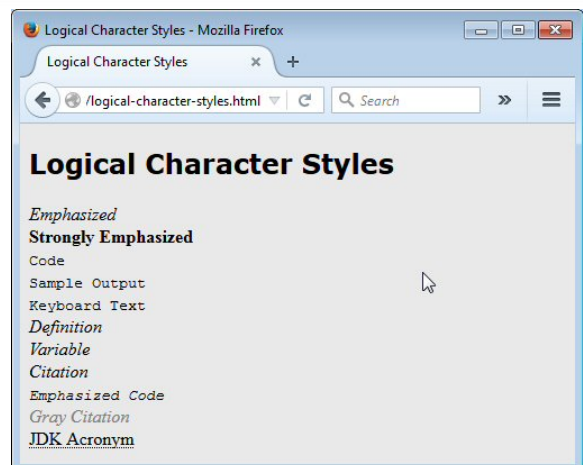
Physical Character Styles: Example

```
<body>
<h2>Physical Character Styles</h2>
<b>Bold</b><br/>
<i>Italic</i><br/>
<tt>Teletype (Monospaced)</tt><br/>
<u>Underlined</u><br/>
Subscripts: f<sub>0</sub> + f<sub>1</sub><br/>
Superscripts: x<sup>2</sup> + y<sup>2</sup><br/>
<small>Smaller</small><br/>
<big>Bigger</big><br/>
<del>Strike Through</del><br/>
<b><i>Bold Italic</i></b><br/>
<big><tt>Big Monospaced</tt></big><br/>
<small><i>Small Italic</i></small><br/>
<font color="gray">Gray</font><br/>
<del>Delete</del><br/>
<ins>Insert</ins><br/>
</body>
```



Logical Character Styles: Example

```
...
<h2>Logical Character Styles</h2>
<em>Emphasized</em><br/>
<strong>Strongly Emphasized</strong><br/>
<code>Code</code><br/>
<samp>Sample Output</samp><br/>
<kbd>Keyboard Text</kbd><br/>
<dfn>Definition</dfn><br/>
<var>Variable</var><br/>
<cite>Citation</cite><br/>
<em><code>Emphasized Code</code></em><br/>
<font color="gray"><cite>Gray Citation</cite></font><br/>
<acronym title="Java Development Kit">JDK Acronym</acronym>
...
```



img: Embedding Images

- **Example**

```

```

- **Attributes:**

- src (required)
- alt (technically required)
- align
 - See `<br clear="all"/>` in next lecture to make sure left or right aligned images do not hang down into the next paragraph
- width, height
- hspace, vspace
- border
- usemap, ismap

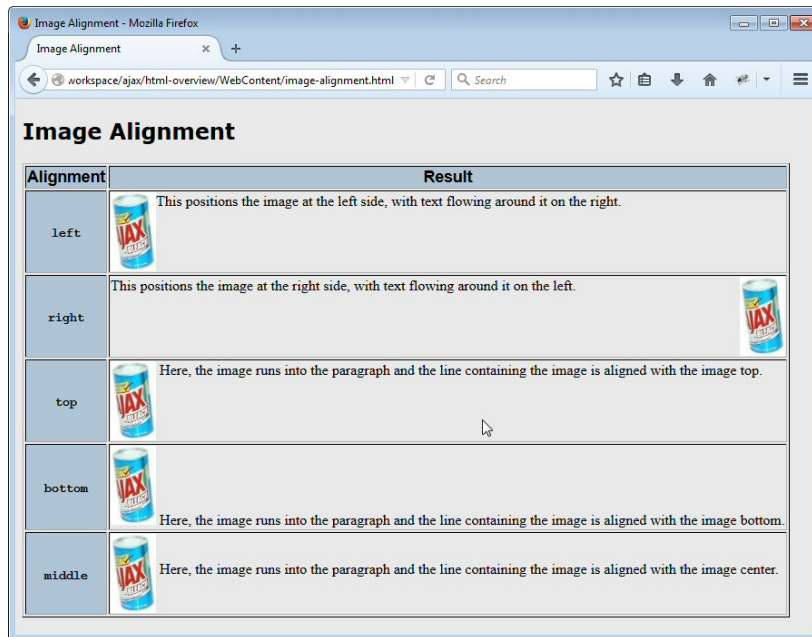
40

Image Alignment: Example (Code)

```
<table border="1">
  <tr><th>Alignment</th>
    <th>Result</th></tr>
  <tr><th><code>left</code></th>
    <td> ... </td></tr>
  <tr><th><code>right</code></th>
    <td> ... </td></tr>
  <tr><th><code>top</code></th>
    <td> ... </td></tr>
  <tr><th><code>bottom</code></th>
    <td>
      Here, the image runs into the paragraph ... </td></tr>
  <tr><th><code>middle</code></th>
    <td> ... </td></tr>
</table>
```

41

Image Alignment: Example (Result)



42

coreservlets.com – custom onsite training



Wrapup

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Summary

- **Basic structure**

```
<!DOCTYPE html>
<html>
<head>
<title>Some Title</title>
<link href="css/some-stylesheet.css" rel="stylesheet"/>
<script src="scripts/some-script.js"></script>
</head>
<body>
...
</body></html>
```

- **Approach**

- Use lowercase for tag and attribute names (<h1 align="center">)
- Include end tags (<p>...</p> or
)
- Use quotes around attribute values (<table border="1">)

44

coreservlets.com – custom onsite training



Questions?

More info:

<http://www.coreservlets.com/javascript-jquery-tutorial/> – Tutorial on JavaScript, jQuery, and jQuery UI

<http://courses.coreservlets.com/Course-Materials/java.html> – General Java programming tutorial

<http://www.coreservlets.com/java-8-tutorial/> – Java 8 tutorial

<http://courses.coreservlets.com/java-training.html> – Customized Java training courses, at public venues or onsite at your organization

<http://coreservlets.com/> – JSF 2, PrimeFaces, Java 8, JavaScript, jQuery, Ext JS, Hadoop, RESTful Web Services, Android, HTML5, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training

Many additional free tutorials at coreservlets.com (JSF, Android, Ajax, Hadoop, and lots more)

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.