Building the User Interface by Using HTML5: Text, Graphics, and Media
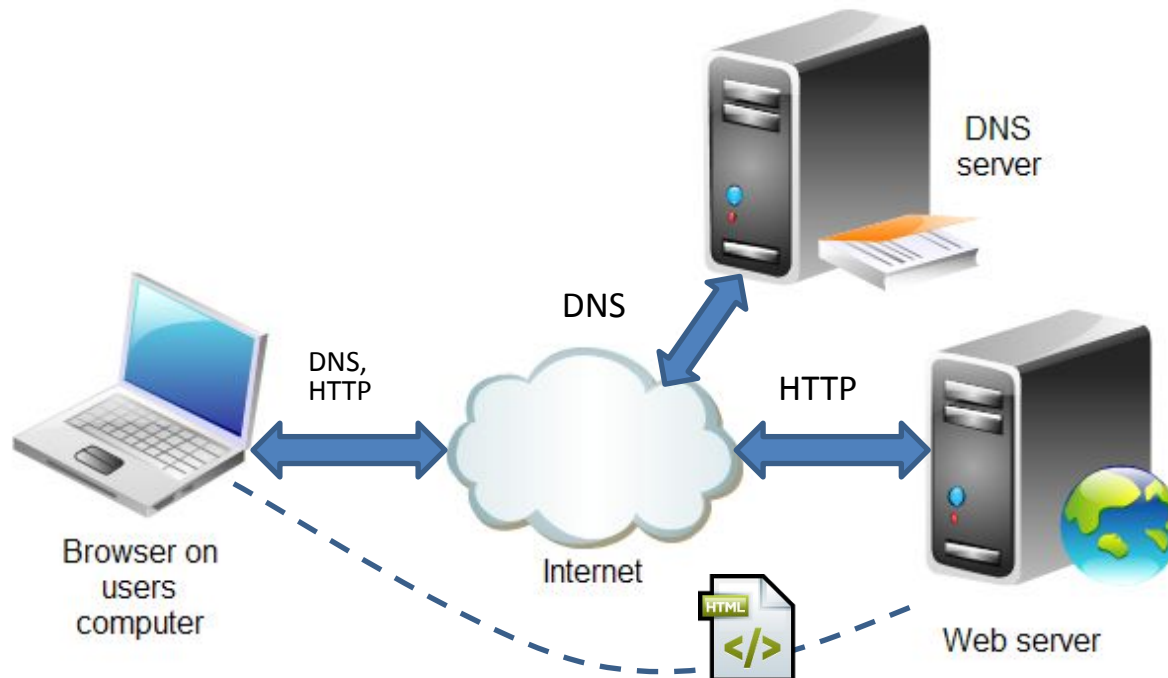
Vyacheslav Koldovskyy
Last update: 25/06/2015

Agenta

- Tags
- Elements
- Attributes
- Nesting
- Entities
- Doctype
- Modified HTML 4.01 elements and new HTML5 elements
- Deprecation
- Figures and figure captions
- Canvas and SVG
- Audio and video

HTML Documents and HTTP Protocol

- Hypertext Markup Language (HTML) is a markup language for Web pages
- Hypertext Transport Protocol (HTTP) is a protocol that ensures transport of pages to clients

HTML Document


```
<!doctype html>
<html>
  <head>
    <title>Sample</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```
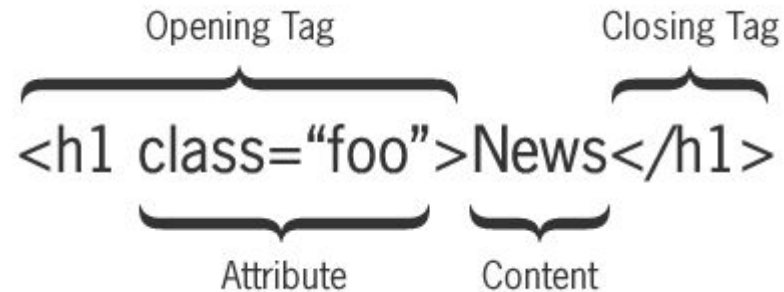
- HTML document is a text document created to represent formatted information including text, vide, images, sounds
- HTML document is created by *tags* and consists of *sections*

Tag Pairs

- Tags are keywords that determine structure of an HTML page
- Keyword is surrounded by angled brackets
- Most tags come in pairs
  - Opening or start tag
  - Closing or end tag

```
<h1>Pet Care 101</h1>
```

- Closing tag must have same case as opening tag

Empty Tags

- Empty tags don't require an end tag
- Examples:
  - `<br />` for a line break
  - `<hr />` for a horizontal line

Required HTML Tags

```
<!doctype html>
<html>
  <head>
    <title>Sample</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

- Tags required on every Web page:
  - `<!doctype>` directive
  - `<html>`
  - `<head>`
  - `<title>`
  - `<body>`

Doctype

- A declaration found at the very top of almost every HTML document
- Specifies the language or rules the page uses
- In HTML, the DOCTYPE is case insensitive. The following DOCTYPEs are all valid:

```
<!doctype html>
<!DOCTYPE html>
<!DOCTYPE HTML>
<!DoCtYpE hTmL>
```

Doctype

- HTML 4.01 doctype example:

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

- HTML5 doctype:

```
<!doctype html>
```

Common HTML Tags

- `<html>`: Identifies the page as an HTML document
- `<head>`: Contains markup and code used by the browser, such as scripts that add interactivity, and keywords to help search engines find the page
- `<title>`: Displays the title of the Web page, which appears at the top of the Web browser, usually on the page's tab in a tabbed browser
- `<body>`: Surrounds content that's visible on the Web page when viewed in a Web browser

Common HTML Tags (Continued)

- `<a href=URL>`: Generally used to anchor a URL to text or an image; can create a named anchor within a document to allow for linking to sections of the document

- `<b>`: Applies boldface to text

- `<hx>`: Creates a heading, which can be first level (h1) through sixth level (h6)

- `<img>`: Inserts an image from a file or another Web site

- `<p>`: Defines text as a paragraph

Elements

- A tag pair or an empty tag is also called an **element.**
- An element can describe content, insert graphics, and create hyperlinks.

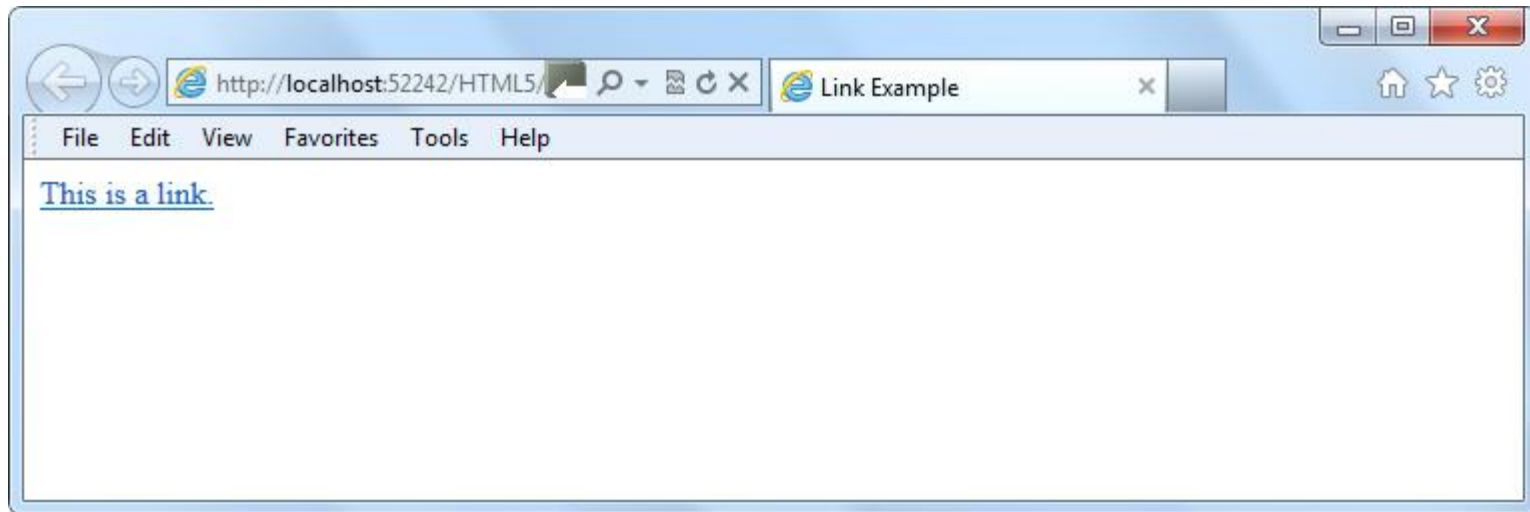# Attributes

- Modifiers of HTML elements that provide additional information
- Are extensions of elements
- Syntax: `<tag attribute="value">`

Creating a Link
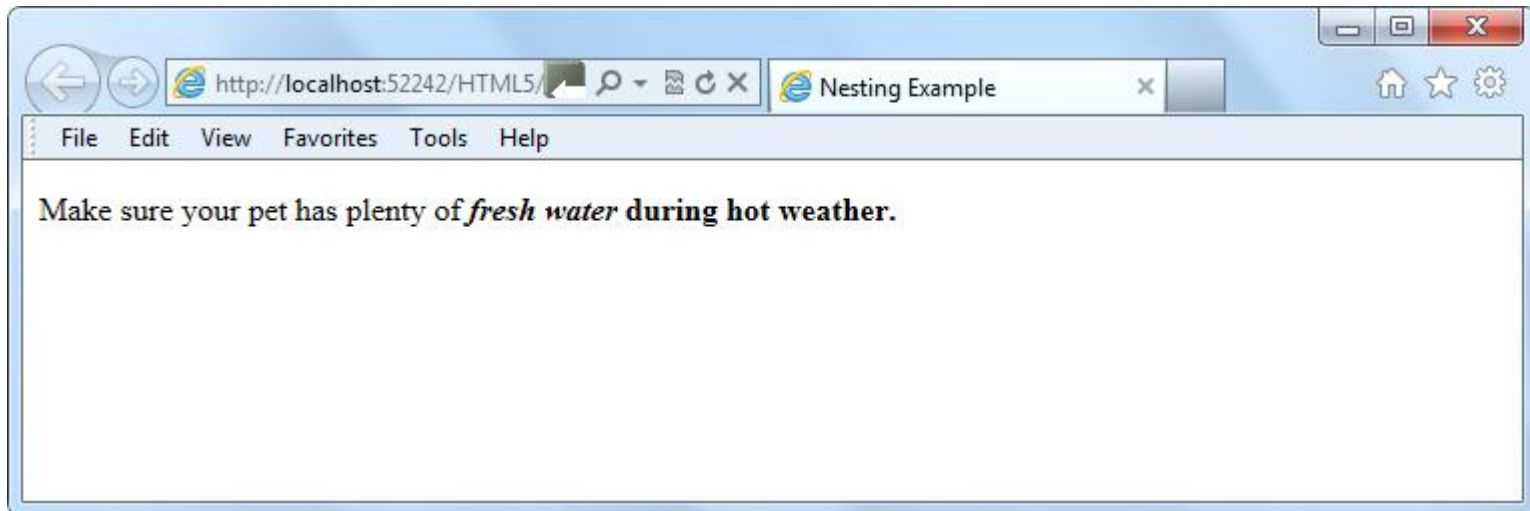
- Example:
  - `<a href="http://www.example.com">This is a link.</a>Global attribute`

Nesting

- To place one element inside another
- Example:
    - ```
      <p>Make sure your pet has plenty of
      <i><b>fresh water</i> during hot
      weather.</p></b>
      ```

Entities

- A special character, such as the dollar symbol, the registered trademark (a capital R within a circle), and accented letters
- Begins with an ampersand (&) and ends with a semicolon (;)
- Examples:
  - entity ® represents the registered trademark symbol
  - Its numerical code is ®

Entities (Continued)

- Called character encoding
- Use UTF-8 encoding whenever possible
- Add the following declaration to the head element:

```
<meta charset="UTF-8">
```

HTML 4 Text-related Elements with New Meaning or Functionality in HTML5

- The `<b>` element should now be used to offset text without conveying importance, such as for keywords or product names.
- The `<i>` element now indicates content in an alternate voice or mood, like spoken text.
- The `<strong>` element indicates strong importance.
- The `<em>` element indicates emphatic stress.
- The `<small>` element should be used for small print, like a copyright line.

Some New HTML5 Elements

- The `<mark>` element highlights text on a page, similar to the highlighting feature in Microsoft Word.
- The `<time>` element displays a machine-readable time and date, such as 10:10 A.M., CST, July 19, 2012, which is handy for blogs and calendars, and potentially helps search engines provide better results when time and date are part of the search criteria.
- The `<wbr>` defines possible line-break.

Deprecation

- While new elements become available, the W3C earmarks other elements for eventual removal because their functionality is no longer useful.

- Removing elements from the list of available HTML elements is referred to as **deprecation.**

- **Note:** The same thing applies to attributes and CSS properties.

Examples of Deprecated HTML Elements

- `<big>`: Makes text bigger relative to the current font size

- `<center>`: Center-aligns text and content

- The fix: Use CSS instead

Image Basics

- A **raster image** is made up of pixels.
    - Example: A photograph
    - Formats: JPG, PNG, GIF, BMP
- A **vector image** is made up of lines and curves based on mathematical expressions.
    - Example: Adobe Illustrator AI file
    - Formats: PNG or GIF for Web display

The `img` element

- Use `img` to add images to an HTML document
- Example: `<img src="images/redball.jpg" alt="Red ball graphic" />`
- The `src` attribute and the `alt` attribute are required to be fully valid.
- The W3C requires the alt attribute for accessibility by people with disabilities.

`figure` and `figcaption` Elements

- The `figure` element specifies the type of figure you're adding
- The `figcaption` element adds a caption to an image on a Web page
  - Can display the caption before or after the image

figure and figcaption Example

```
<figure>

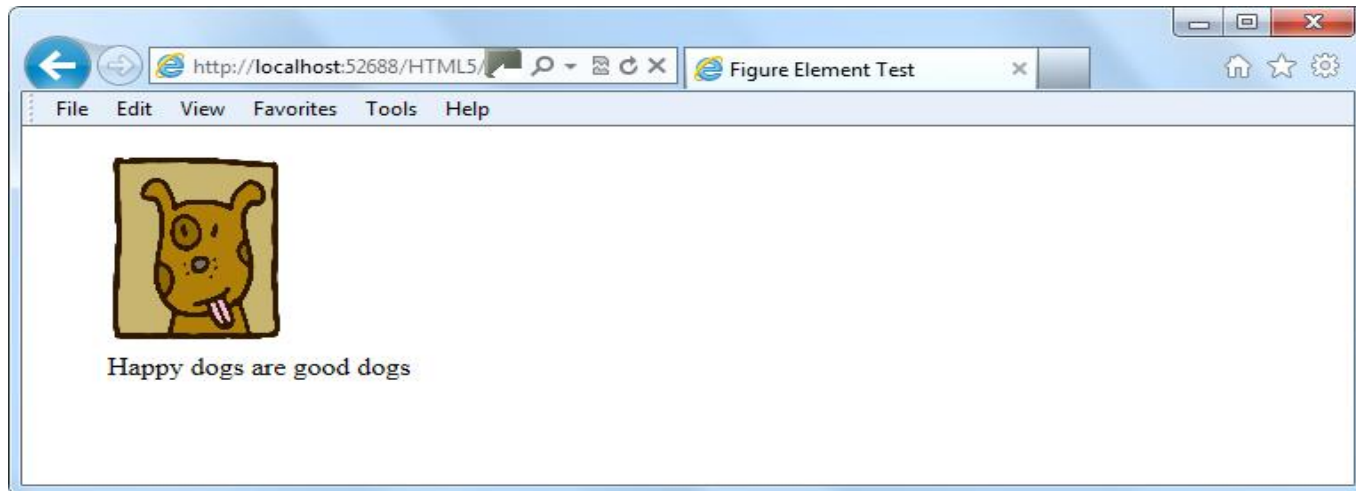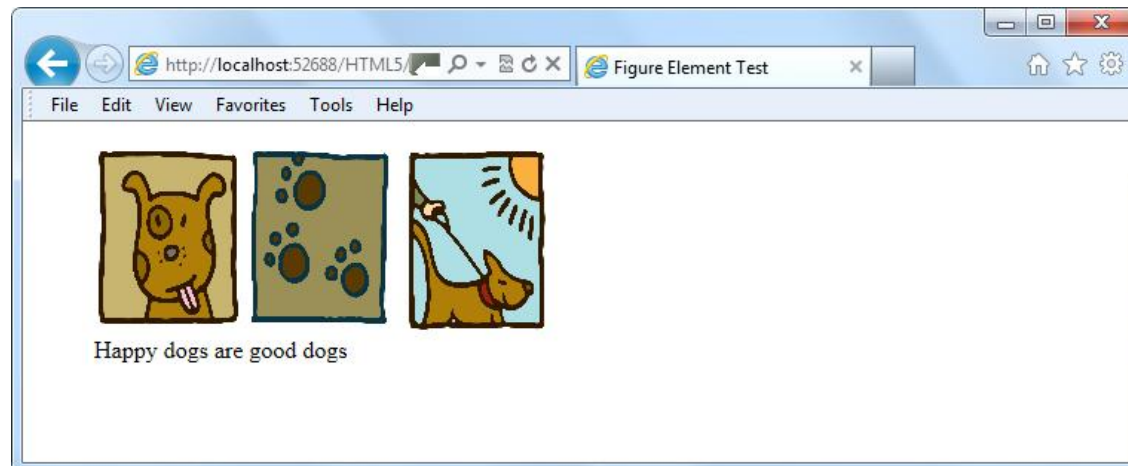   <img src="doghappy.jpg" alt="Happy dog"
   width="100" height="125" />

   <figcaption>Happy dogs are good dogs</figcaption>

</figure>
```


Illustration: © MightyIsland/iStockphoto

Side by Side Example

```
<figure>

    <img src="doghappy.jpg" alt="Happy dog"
    width="100" height="125" />

    <img src="dogpaws.jpg" alt="Dog paws"
    width="100" height="125" />

    <img src="dogwalk.jpg" alt="Walking a dog"
    width="100" height="125" />

    <figcaption>Happy dogs are good dogs</figcaption>

</figure>
```


Illustrations: © MightyIsland/iStockphoto

Canvas

- Use JavaScript to draw pixel-based shapes on a canvas
- Include color, gradients, and pattern fills
- Render text with various embellishments
- Animate objects by making them move, change scale, and so on
- Basic syntax for the canvas element:

```
<canvas id="smlRectangle" height="100" width="200"></canvas>
```

Canvas Example

```html
<!doctype html>

<html>
  <head>
    <meta charset="UTF-8">
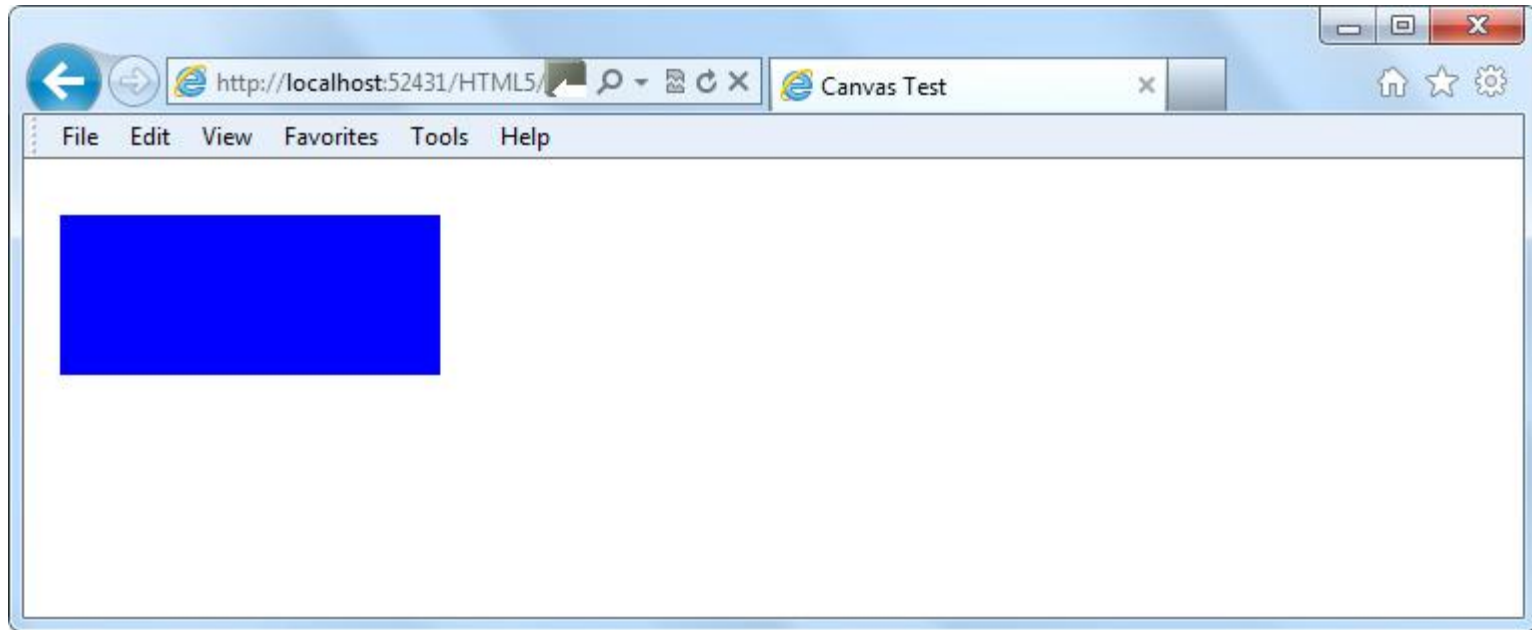    <title>Canvas Test</title>

<script>
    function f1() {
        var canvas =
        document.getElementById("smlRectangle");
        context = canvas.getContext("2d");
        context.fillStyle = "rgb(0,0,255)";
        context.fillRect(10, 20, 200, 100);
    }
  </script>
  </head>

<body onload = "f1();">

<canvas id="smlRectangle" height="100" width="200 ">
</canvas>

</body>

</html>
```

Canvas Example

Fallback

- "Backup" content that displays if primary content cannot
- Can be a problem with some older browsers
  - Cannot render canvas drawings or animation, for example
- Fallback adds an image, text, or some other HTML content within the canvas element that displays if the drawing cannot

Scalable Vector Graphics (SVG)

- A language for describing 2D vector graphics in Extensible Markup Language (XML)
- SVG graphics referred to as objects
- SVG loads into the DOM
- Vector graphic changes size to fit screen, whether 32–inch PC monitor or smartphone
- SVG is not new, but HTML5 can embed SVG objects in Web pages without using `<object>` or `<embed>` tags

SVG Example

```html
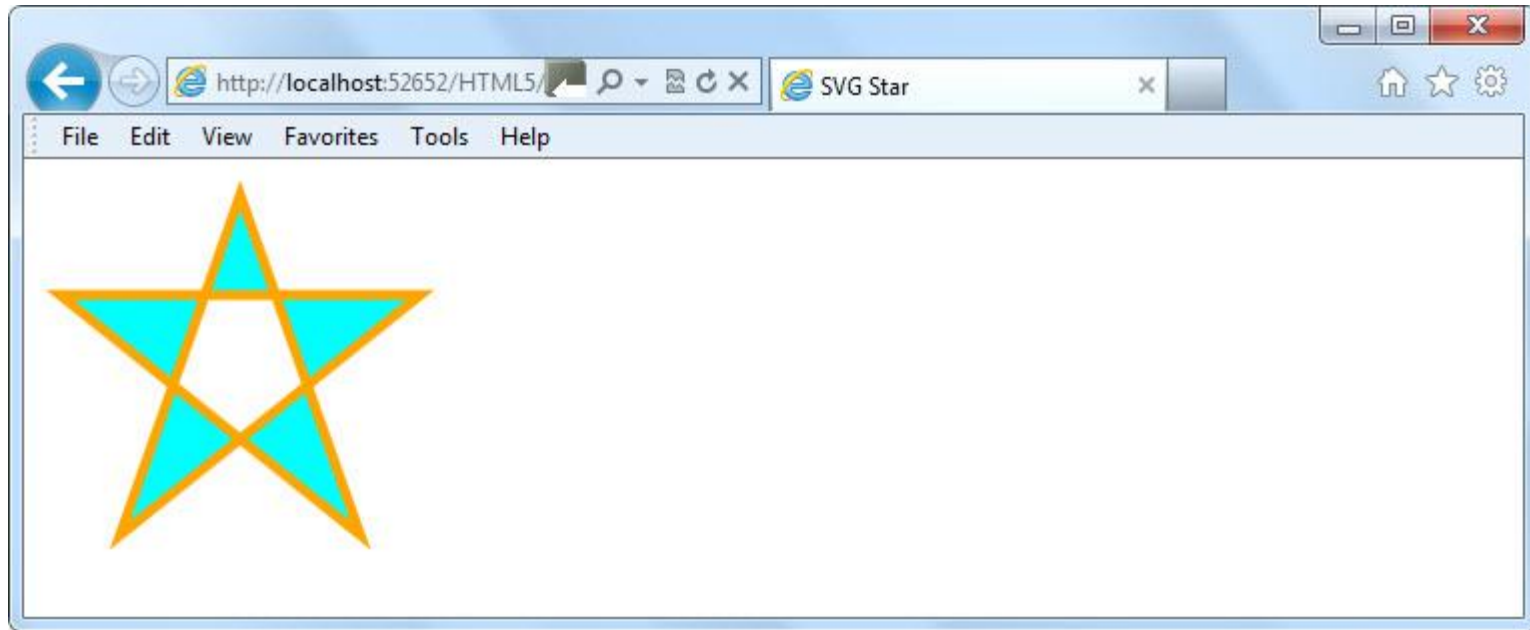<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>SVG Star</title>
  </head>
<body>
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <polygon points="100,10 40,180 190,60 10,60 160,180"
style="fill:aqua;stroke:orange;stroke-width:5;
fill-rule:evenodd;"/>
</svg>
</body>
</html>
```

SVG Example

When to Use Canvas Instead of SVG

- If the drawing is relatively small, use canvas.
- If the drawing requires a large number of objects, use canvas; SVG degrades as it continually adds objects to the DOM
- Generally, use canvas for small screens and SVG for larger screens.
- If you must create highly detailed vector documents that must scale well, go with SVG.
- If you are displaying real-time data output, such as maps, map overlays, weather data, and so on, use canvas.

`video` Element

- Enables you to incorporate videos in HTML documents using minimal code

- Markup example:

```
<video src="intro.mp4" width="400"
height="300">
</video>
```

`video` Attributes

- `poster`: Displays a static image file before the video loads

- `autoplay`: Start playing the video automatically upon page load

- `controls`: Displays a set of controls for playing, pausing, and stopping the video, and controlling the volume

- `loop`: Repeats the video

Video Markup Example

```html
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

`audio` Element

- Enables you to incorporate audio (music, other sounds) in HTML documents using minimal code

- Markup example:

```
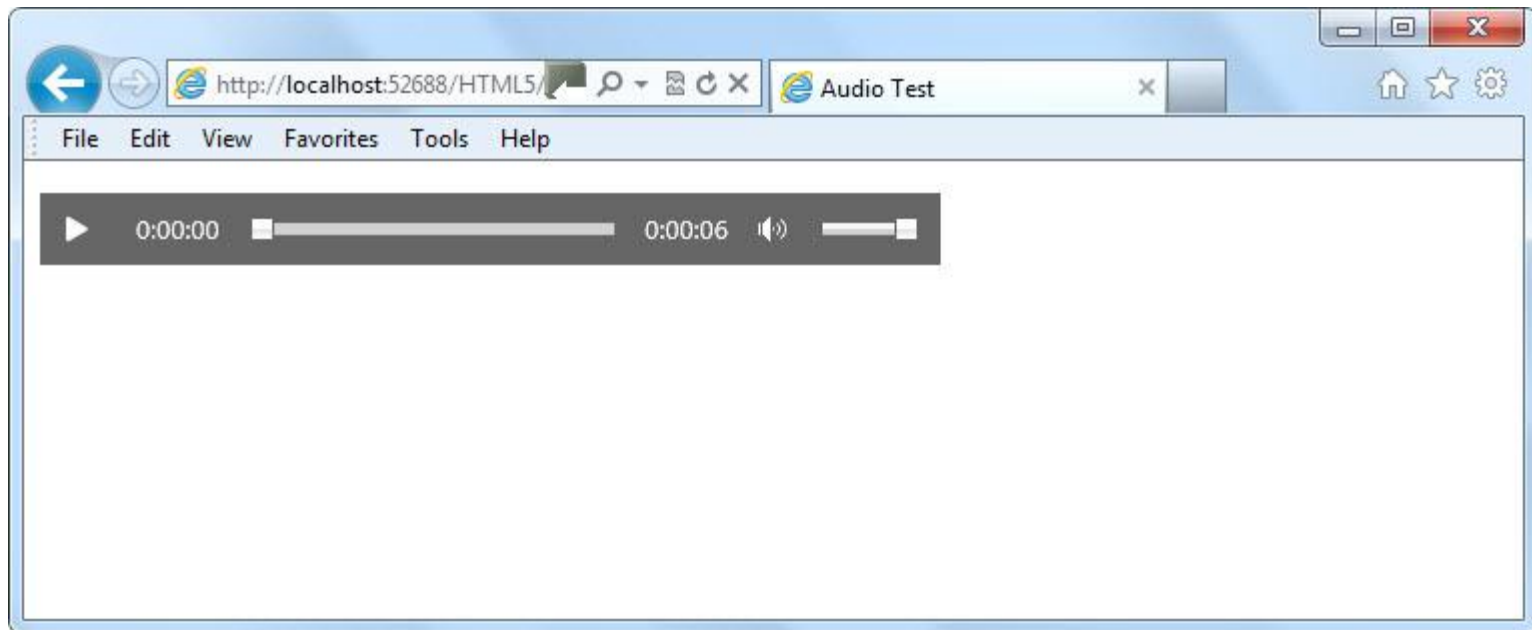<audio src="sample.mp3"
controls="controls">
</audio>
```

audio Example

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Audio Test</title>
  </head>
<body>
  <audio src="sample.mp3" controls="controls">
  </audio>
</body>
</html>
```

`audio` Example

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
Website: **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!