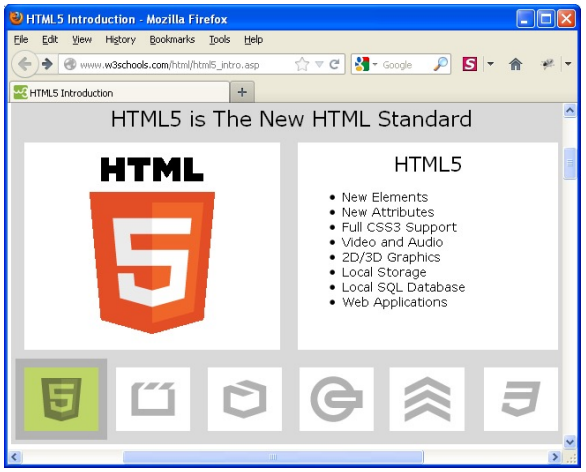
HTML5

Background

- HTML5 is the next major revision of HTML
- The Web Hypertext Application Technology Working Group (WHATWG) started work on the specification in June 2004
- HTML5 supersedes HTML 4.01, XHTML 1.0 and XHTML 1.1
- HTML5 vocabulary/APIs and canvas "Recommendations" available at
  http://www.w3.org/TR/html5/
  http://www.w3.org/TR/2dcontext/
- The latest versions of browsers have support for HTML5:
  - http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML5)
- HTML 5.2, 2nd Edition "Recommendation" (Dec. 2017) available at
  https://www.w3.org/TR/html52/
- HTML Living Standard (April 2021) available at:
  https://html.spec.whatwg.org/multipage/
- W3C and WHATWG Agreement to collaborate on single version of HTML:
  https://www.w3.org/blog/news/archives/7753

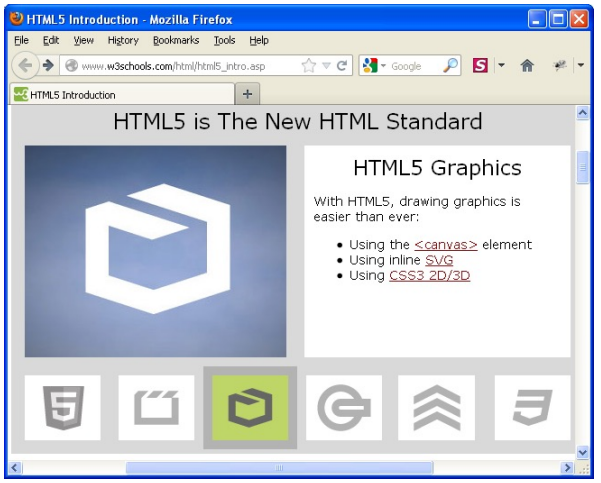
Major New Elements in HTML5

- New semantic elements like <header>, <footer>, <article> and <section>
- <video> and <audio> that you can embed on your web pages without resorting to third-party plug-ins
- Canvas, a two-dimensional drawing surface that you can program with JavaScript
- Scalable vector graphics (SVG)
- Geolocation, whereby visitors can choose to share their physical location with your web application
- Persistent local storage without resorting to third-party plug-ins
- Offline web applications that work even after network access is interrupted
- Improvements to HTML web forms
- Microdata, which lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics
- New APIs for complex web applications including support for mobile devices
- See: http://www.w3schools.com/html/html5_intro.asp

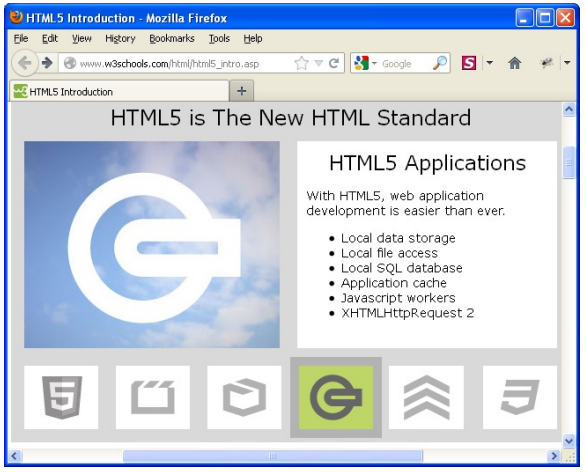


New Elements and attributes

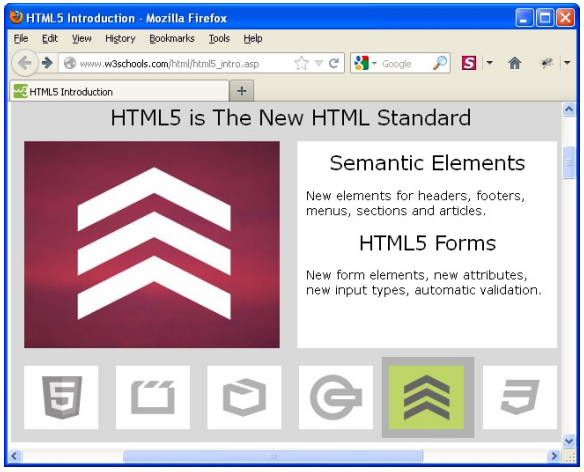
New Video/Audio support



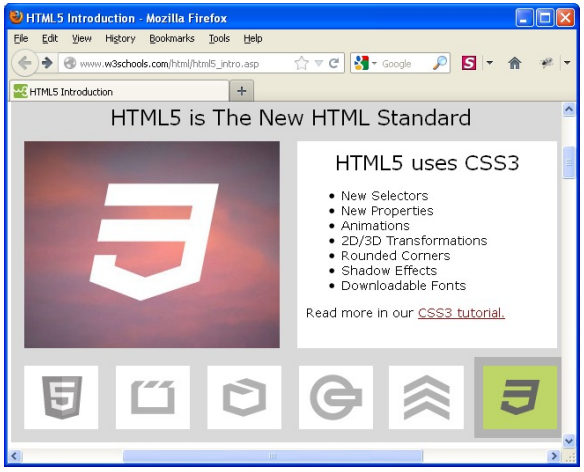
New Graphics support



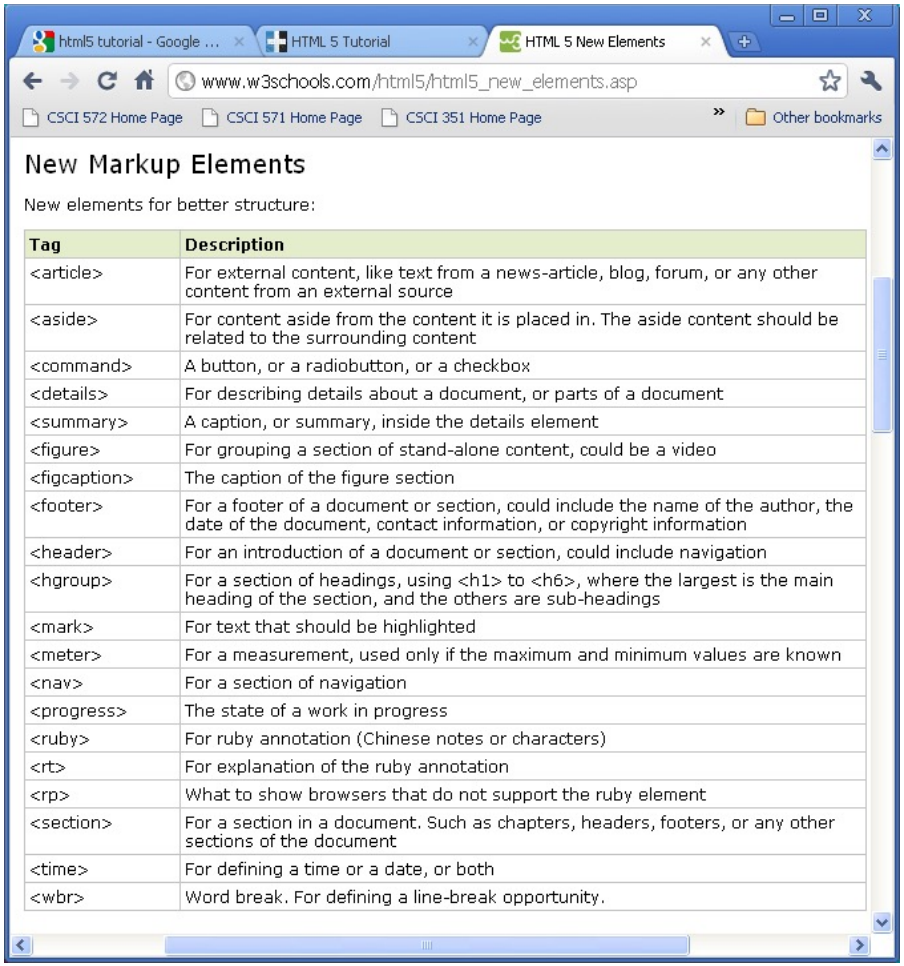
Local Storage API support



New Semantic Elements/Forms

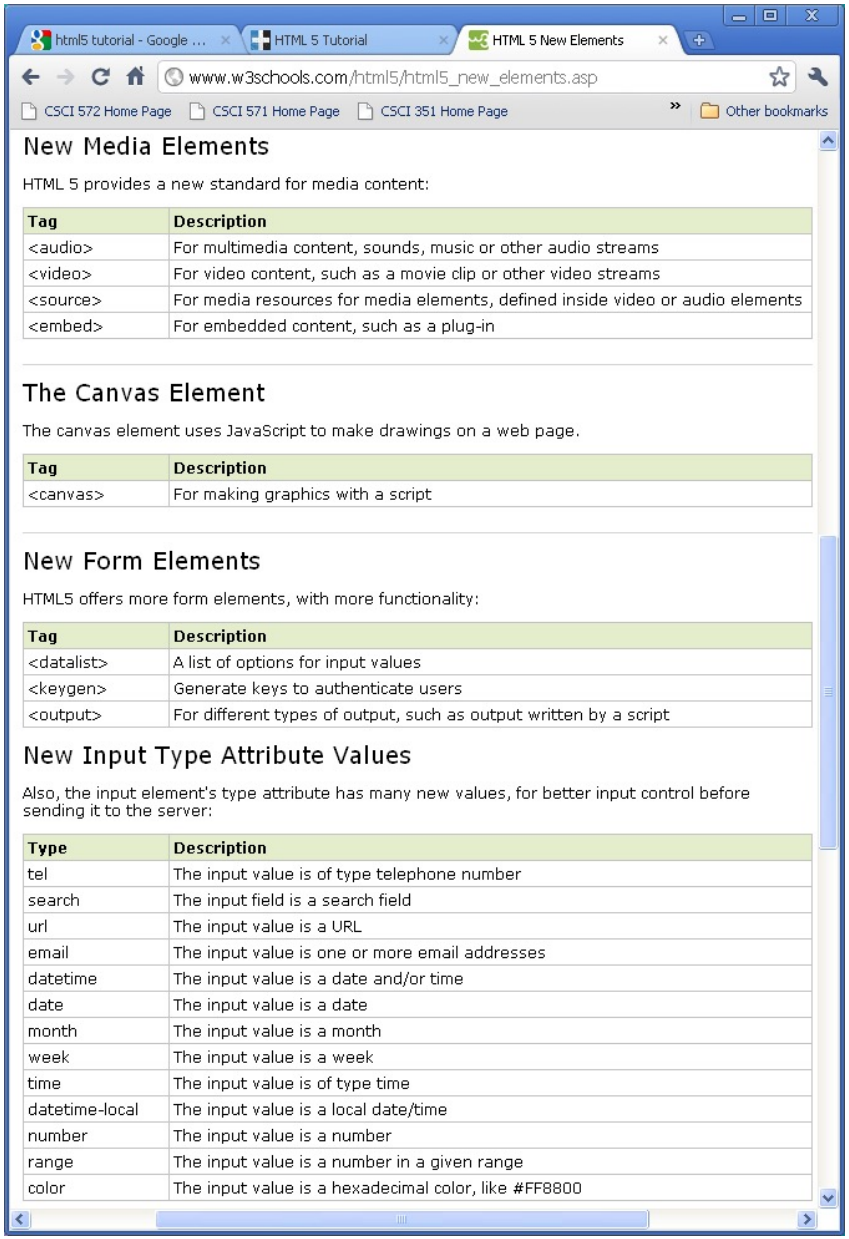


Support for CSS3



New Markup Elements

https://www.w3schools.com/html/default.asp



New Media, Canvas, Form Elements

Summary of new HTML5 Elements

HTML5 - Removed Elements

- The following elements are not in HTML 5 because their effect is purely presentational and therefore better handled by CSS:
  - *basefont, big, center, font, s, strike, tt, u*
- The following elements are not in HTML 5 because their usage affected usability and accessibility for the end user in a negative way:
  - *frame , frameset, noframes*
- *The following elements are not included because they have not been used often, created confusion or can be handled by other elements*
  - *acronym* is not included because it has created lots of confusion. Authors are to use *abbr* for abbreviations.
  - *applet* has been obsoleted in favor of *object*.
  - *isindex* usage can be replaced by usage of form controls.
  - *dir* has been obsoleted in favor of *ul*.

HTML5 - Removed Attributes - Handled by CSS

- align attribute on caption, iframe, img, input, object, legend, table, hr, div, h1, h2, h3, h4, h5, h6, p, col, colgroup, tbody, td, tfoot, th, thead and tr.
- alink, link, text and vlink attributes on body.
- background attribute on body.
- bgcolor attribute on table, tr, td, th and body.
- border attribute on table, img and object.
- cellpadding and cellspacing attributes on table.
- char and charoff attributes on col, colgroup, tbody, td, tfoot, th, thead and tr.
- clear attribute on br.
- compact attribute on dl, menu, ol and ul.
- frame attribute on table.
- frameborder attribute on iframe.
- height attribute on td and th.
- hspace and vspace attributes on img and object.
- marginheight and marginwidth attributes on iframe.
- noshade attribute on hr.
- nowrap attribute on td and th.
- Width attribute on hr, table, td, th, col, colgroup and pre
- scrolling attribute on iframe.
- size attribute on hr, input and select.
- type attribute on li, ol and ul.
- valign attribute on col, colgroup, tbody, td, tfoot, th, thead and tr.
- rules attribute on table.

To simplify HTML many attributes have been eliminated as their effect is more properly produced using CSS

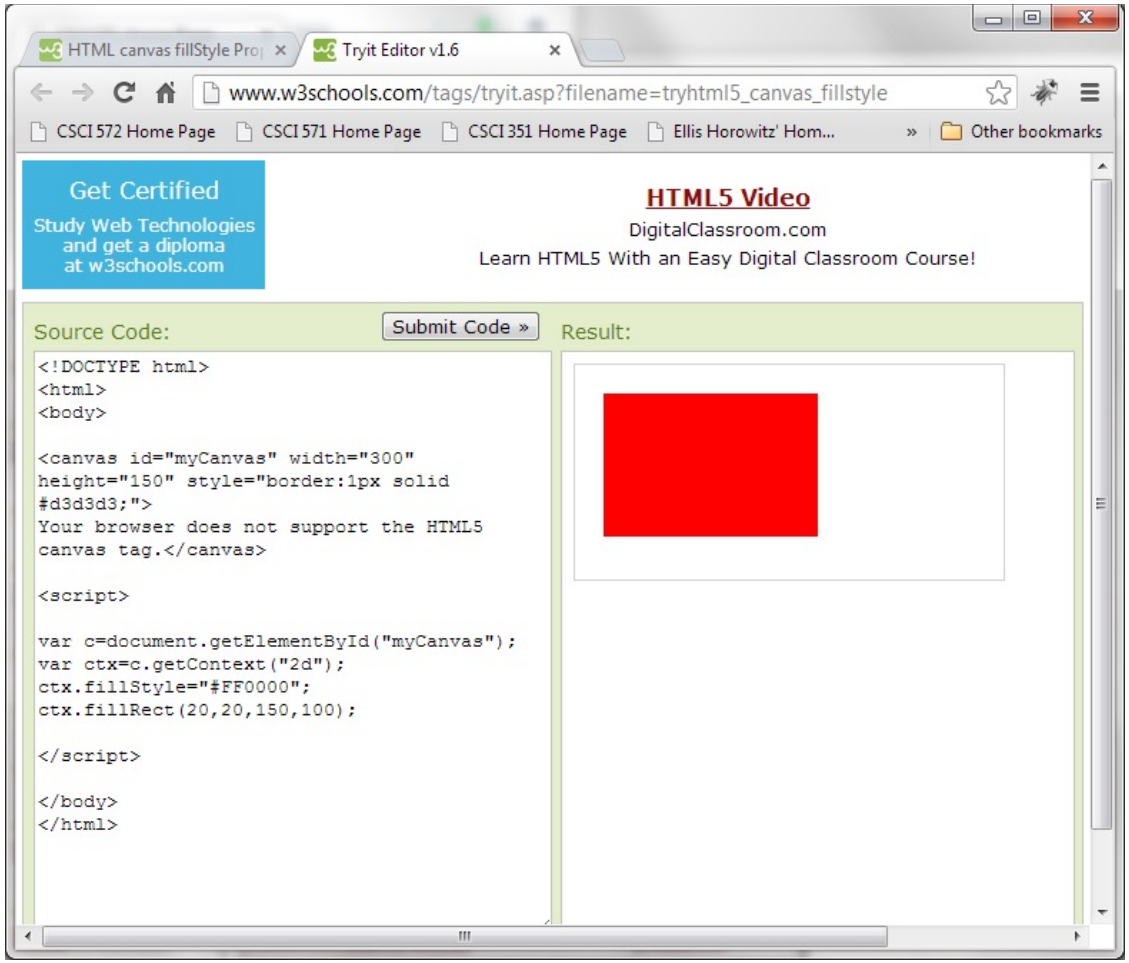
<canvas> Element - Drawing

- The <canvas> element is "a resolution-dependent bitmap canvas which can be used for rendering graphs, game graphics, or other visual images on the fly."
- A *canvas* is a rectangle in your page where you can use JavaScript to draw anything you want.
- A <canvas> element has **no content** and **no border** of its own.
- You can have more than one <canvas> element on the same page.
- Each canvas will show up in the DOM, and each canvas maintains its own state. If you give each canvas an **id attribute**, you can access them just like any other element.
- E.g. if one adds an id attribute, e.g.

  <canvas id="mycanvas" width="300" height="225"></canvas>
- Now you can easily find that <canvas> element in the DOM, e.g.

  var a_canvas = document.getElementById("mycanvas");

<canvas> element - Drawing a Red Rectangle



Every canvas has a drawing *context*.

Once you have found a <canvas> element in the DOM you call its `getContext()` method. You must pass the string "2d" to the `getContext()` method

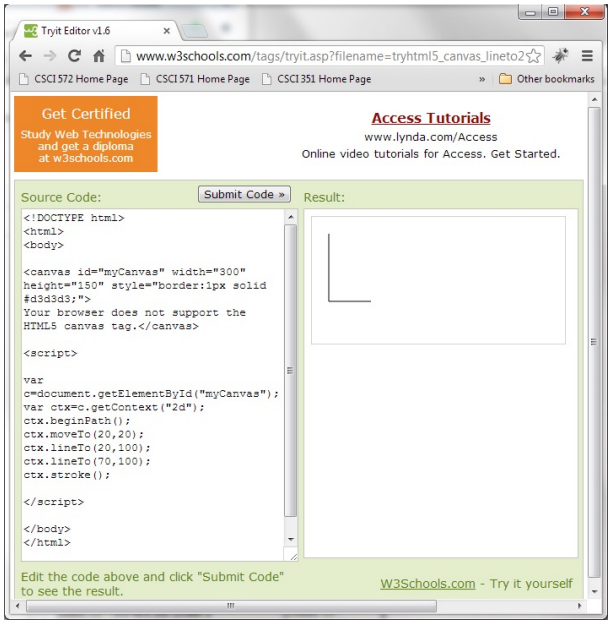
JavaScript and DOM are used to create a red rectangle

https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_canvas_fillrect

<canvas> element - Coordinates

- The canvas is a two-dimensional grid. The coordinate (0, 0) is at the upper-left corner of the canvas. Along the X-axis, values increase towards the right edge of the canvas. Along the Y-axis, values increase towards the bottom edge of the canvas.
- To draw straight lines in pencil, you use the following two methods:
- **moveTo(x, y)** moves the pencil to the specified starting point.
- **lineTo(x, y)** draws a line to the specified ending point.

Move to (20,20)
Draw a line to (20,100)
Draw a line to (70,100)
stroke() makes the lines Visible



https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_canvas_stroke

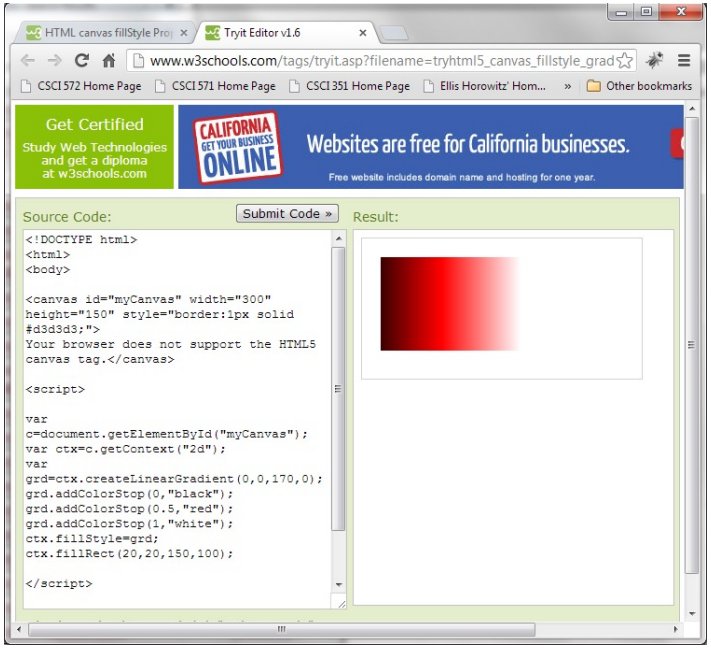
Drawing Context

- There's a whole group of properties and methods devoted to drawing rectangles:
- The fillStyle property can be a CSS color, a pattern, or a gradient. The default fillStyle is solid black, but you can set it to whatever you like.
- A gradient shows a difference in concentrations over an area, often using color intensity
- Createlineargradient(x0,y0,x1,y1) returns a canvas gradient object starting at (x0,y0) and ending at (x1,y1)
- addColorStop(offset, color) adds a color stop with the given color to the gradient at the offset; 0 offset is at one endpoint and 1 offset is at the other endpoint
- `fillRect(x, y, width, height)` draws a rectangle filled with the current fill style.
- `clearRect(x, y, width, height)` clears the pixels in the specified rectangle.

See http://www.colorzilla.com/gradient-editor/
for the Ultimate CSS Gradient Generator
or CSS Gradients in HTML5
http://www.fix-css.com/2011/07/css-gradients-in-html5/

Code to the right first defines a linear gradient region; AddColorStop is then used to set black at the left endpoint and white at the right endpoint; fillStyle is a context property that when set to grd causes the color to fill the region as shown

See
http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_canvas_tut_grad



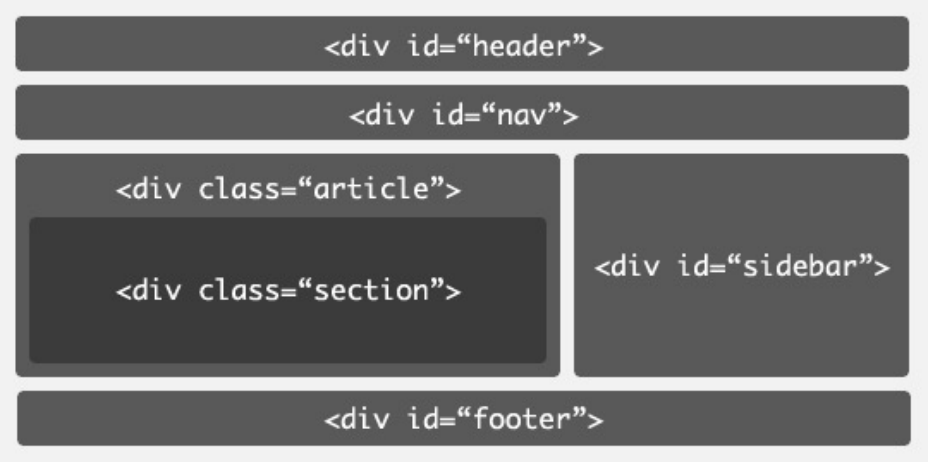
<svg> Element - Drawing

- The <svg> element is "a container for SVG graphics."
- SVG has several methods for drawing paths, boxes, circles, text and graphivs images.
- SVG graphics is supported by all major browsers.
- SVG is a language for describing 2D graphics in XML.
- Since SVG is XML based, every element is available in the SVG DOM.
- If attributes of an SVG object are changed, the browser can automatically re-render the shape.

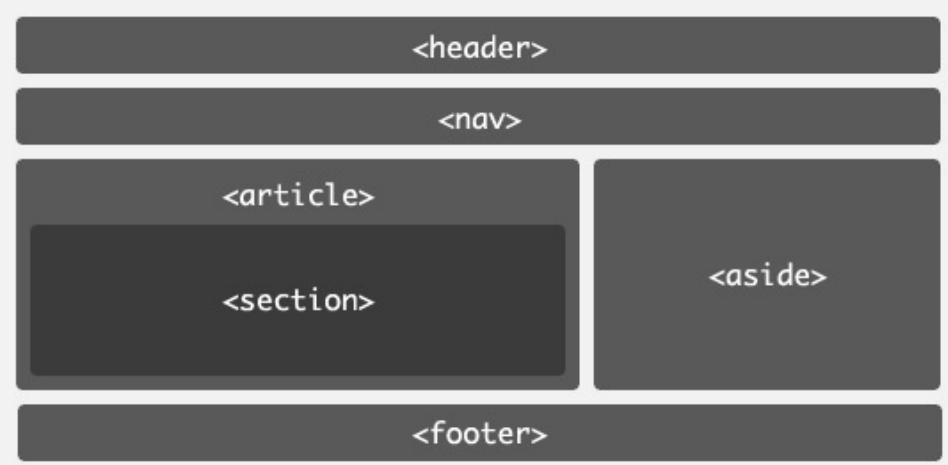
Canvas	SVG
• Resolution dependent • No support for event handlers • Poor text rendering capabilities • You can save the resulting image as .png or .jpg • Well suited for graphic-intensive games	• Resolution independent • Support for event handlers • Best suited for applications with large rendering areas (Google Maps) • Slow rendering if complex (anything that uses the DOM a lot will be slow) • Not suited for game applications

Comparison of Canvas and SVG

Document Structure



Old Way



New Way

- HTML5 introduces a whole set of new elements that make it much easier to structure pages.
- By identifying the purpose of sections in the page using specific sectioning elements, assistive technology (e.g. for the blind) can help the user to more easily navigate the page.
- For example, they can easily skip over the navigation section or quickly jump from one article to the next without the need for authors to provide skip links.
- Authors also benefit because replacing many of the divs in the document with one of several distinct elements can help make the source code clearer, easier to author, and easier to scrape

See https://www.w3schools.com/html/html5_semantic_elements.asp

Defining Sections of a Web Page with <section> Element

- <section> - a grouping of content, e.g. chapters or tabbed pages or a page is divided into Introduction, News Items and Contact Information
- Example of an article about apples with two sections

```
<article> <hgroup>
<h1>Apples</h1>
<h2>Tasty, delicious fruit!</h2> </hgroup>
<p>The apple is the pomaceous fruit of the apple tree.</p>
<section> <h1>Red Delicious</h1>
   <p>These bright red apples are the most common found in many
      supermarkets.</p> </section>
 <section> <h1>Granny Smith</h1>
   <p>These juicy, green apples make a great filling for apple pies.</p>
 </section>
</article>
```

Don't use <section> just as hook for styling or scripting;
For that you should use
Don't use <section> if <article>, <aside> or <nav> is more appropriate
Don't use <section> unless there is naturally a heading at the start of the section

Defining Sections of a Web Page with <nav> Element

- <nav> - a section of a page that links to other pages or to parts within the page; a section with navigation links
- In the following example, the page has several places where links are present, but only one of those places is considered a navigation section.

```
<body> <header>
<h1>Wake up sheeple!</h1>
<p><a href="news.html">News</a> -
   <a href="blog.html">Blog</a> -
   <a href="forums.html">Forums</a></p>
<p>Last Modified: <time>2009-04-01</time></p>
<nav> <h1>Navigation</h1>
   <ul>
     <li><a href="articles.html">Index of all articles</a></li>
     <li><a href="today.html">Things needed to wake up for today</a></li>
     <li><a href="successes.html">Sheep manage to wake</a></li>
    </ul>
 </nav>
</header> <div>
<article> <header> <h1>My Day at the Beach</h1>
```

Defining Sections of a Web Page with <article> Element

- <article> - a component of a page that consists of a self-contained portion intended to be independently distributable or reusable;
  - This could be a forum post, a magazine or newspaper article, a blog entry, a user-submitted comment, an interactive widget or gadget, or any other independent item of content; Here is an example of a blog post showing an article followed by two comments:

```
<article> <header>
  <h1>The Very First Rule of Life</h1>
  <p><time pubdate datetime="2009-10-09T14:28-08:00"></time></p>
          </header>
  <p>If there's a microphone anywhere near you, assume it's hot and sending
  whatever you're saying to the world. Seriously.</p> <p>...</p>
 <section> <h1>Comments</h1>
<article> <footer>
  <p>Posted by: George Washington</p>
  <p><time pubdate datetime="2009-10-10T19:10-08:00"></time></p>
  </footer>
  <p>Yeah! Especially when talking about your lobbyist friends!</p> </article>
<article> <footer>
<p>Posted by: George Hammond</p> <p><time pubdate datetime="2009-10-10T19:15-
08:00"></time></p> </footer>
<p>Hey, you have the same first name as me.</p>
</article> </section> </article>
```

Video on the Web

- HTML5 defines a standard way to embed video in a web page, using a <video> element.
- Video container files include video and audio files
- There are lots of competing container files, e.g.
  - MPEG4 compressed video (mp4 or m4v)
  - QuickTime (.mov)
  - Flash Video (.flv) from Adobe
  - Ogg (.ogv) open source
  - WebM (VP8/VP9 video + Vorbis audio), from Google
  - Audio Video Interleave (.avi), invented by Microsoft

Video Codecs

- There are *lossy* and *lossless* video codecs. Lossless video is much too big to be useful on the web. A *lossy video codec* means that information is being irretrievably lost during encoding.
- Popular video codecs are:
  - MPEG-4 (also known as MPEG4 Part 2)
  - H.264 (also known as MPEG4 Part 10), a.k.a. Advanced Video Coding (AVC)
  - H.265, successor to H.264 (doubles video compression and supports 8K UHD), a.k.a. High Efficiency Video Coding (HEVC) or MPEG-H part 2, used in FaceTime
  - Theora
  - VP8 open source codec, formerly from On2, now Google WebM
  - VP9, successor to VP8 (doubles video compression)
  - Sorenson Spark (H.263 variant) from Adobe
- The H.264 standard is split into "profiles," defining a set of optional features that trade complexity for file size. Higher profiles use more features, offer better visual quality at smaller file sizes, take longer to encode, and require more CPU power to decode in real-time.
- Apples iPhone 12 supports H.264 video format: playback in 1080p (30 & 60 fps), MPEG4 (30 fps), 4K (24, 30 and 60 fps), also supports H.265 (HEVC) for FaceTime; Apple TV 4K (5th gen.) set-top box supports H.264 video up to 2160p (60 fps) and MPEG-4 video (30 fps), Main 10 profile; and Adobe Flash on a desktop PC supports Baseline, Main, and High profiles.
- YouTube uses H.264 to encode HD videos (Google dropped Flash, replaced by HTML5 video).
  http://www.theverge.com/2015/1/27/7926001/youtube-drops-flash-for-html5-video-default

Video Codecs (recent development)

- The H.264 vs. WebM video "war" has ended.
- On March 2012 Firefox CTO "capitulated" and decided to support H.264. See:

http://news.cnet.com/8301-30685_3-57397031-264/mozilla-execs-capitulate-in-h.264-web-video-war/

- HTML5 / H.264 now supported by YouTube. See:

http://www.youtube.com/html5

- On March 7, 2013, Google admitted its VP8/WebM codec infringes MPEG H.264 patents, and agreed to pay to license H.264 patents, see

http://www.businesswire.com/news/home/20130307006192/en/Google-MPEG-LA-Announce-Agreement-Covering-VP8

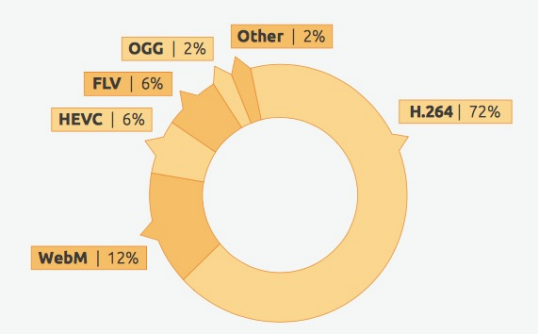
- As of January 2016, H.264 reigns supreme, Flash video on life support, see

https://www.encoding.com/blog/2016/01/27/h-264-and-hls-reign-in-online-video-finds-encoding-com-report/

- Adobe system "retired" Flash at the end of 2020. See:

http://www.scmp.com/tech/enterprises/article/2104092/adobe-systems-retiring-flash-end-2020

https://www.adobe.com/products/flashplayer/end-of-life.html



Audio Codecs

- And like lossless video, lossless audio is really too big to put on the web.
- The *audio codec* specifies how to decode the audio stream and turn it into digital waveforms that your speakers then turn into sound.
- On the web, there are really only three audio codecs you need to know about:
  - MP3
    - MP3s can contain **up to 2 channels** of sound. They can be encoded at different *bitrates*: 64 kbps, 128 kbps, 192 kbps, and a variety of others from 32 to 320. Higher bitrates mean larger file sizes and better-quality audio,
  - Advanced Audio Encoding, (AAC and AAC+)
    - It is the default format for Apple's iTunes
    - It supports up to **48 channels** of sound
  - Vorbis
    - Usually comes in a Ogg container
    - Android phones can play Vorbis audio

HTML MarkUp of Video

- To insert a video file in a web page, use the <video> element

```
<video src="pr6.webm" width="320" height="240"></video>
```

- The <video> element has methods like play() and pause() and a read/write property called currentTime. There are also read/write volume and muted properties.
- you can tell the browser to display a built-in set of controls. To do this, just include the controls attribute in your <video> tag.

```
<video src="pr6.webm" width="320" height="240" controls></video>
```

Using Attributes preload and autoplay for Video

- The `preload` attribute tells the browser that you would like it to start downloading the video file as soon as the page loads. This makes sense if the entire point of the page is to view the video. On the other hand, if it's just supplementary material that only a few visitors will watch, then you can set `preload` to none to tell the browser to minimize network traffic.
- Here's an example of a video that will start downloading (but not playing) as soon as the page loads:

```
<video src="pr6.webm" width="320" height="240" preload></video>
```

- And here's an example of a video that will *not* start downloading as soon as the page loads:

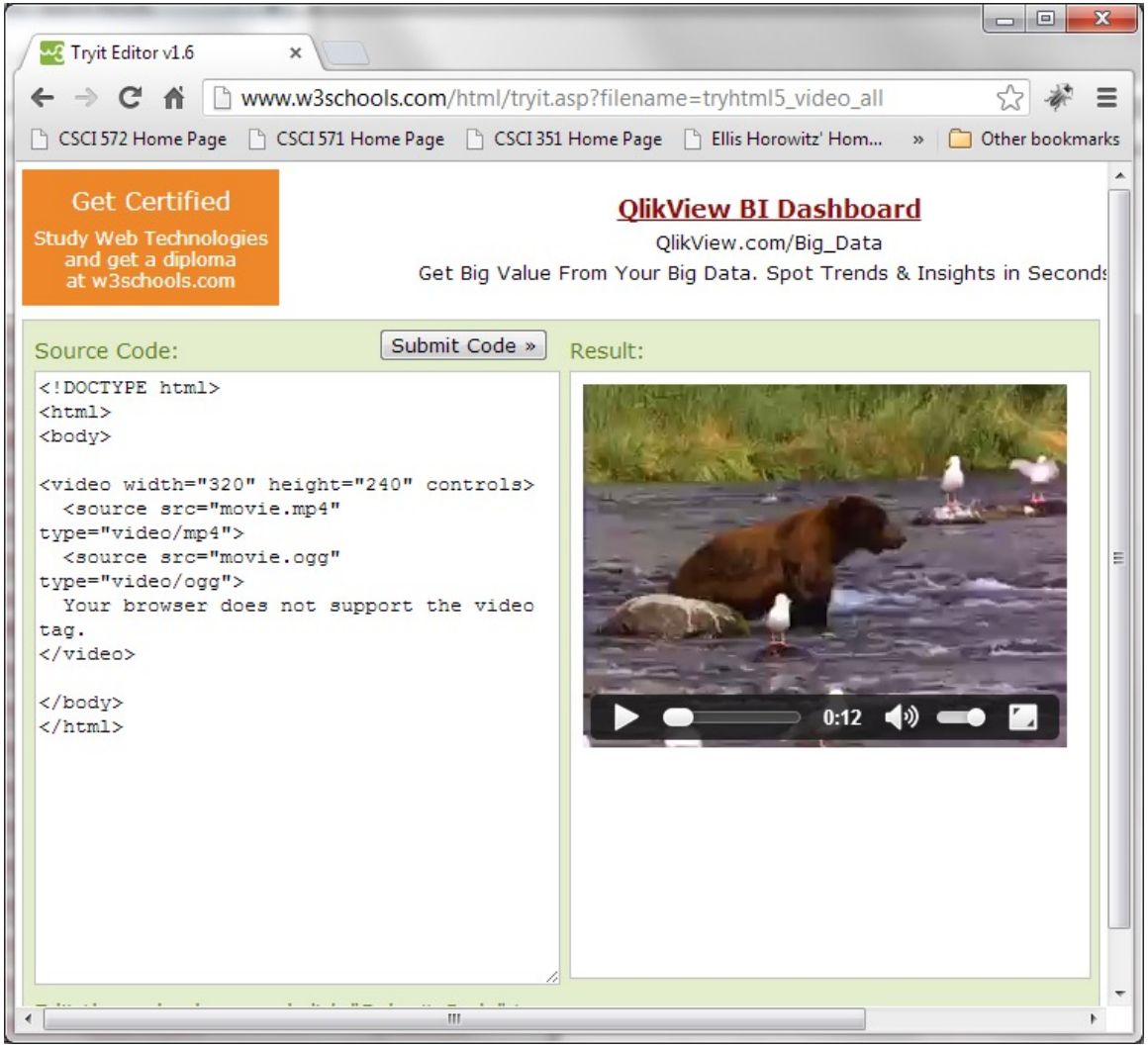
```
<video src="pr6.webm" width="320" height="240" preload="none"></video>
```

- The **autoplay** attribute tells the browser that you want to start downloading the video file as soon as the page loads, *and* you would like it to start playing the video automatically as soon as possible. Mostly blocked by browsers, unless muted.
- Here's an example of a video that will start downloading and playing as soon as possible after the page loads:

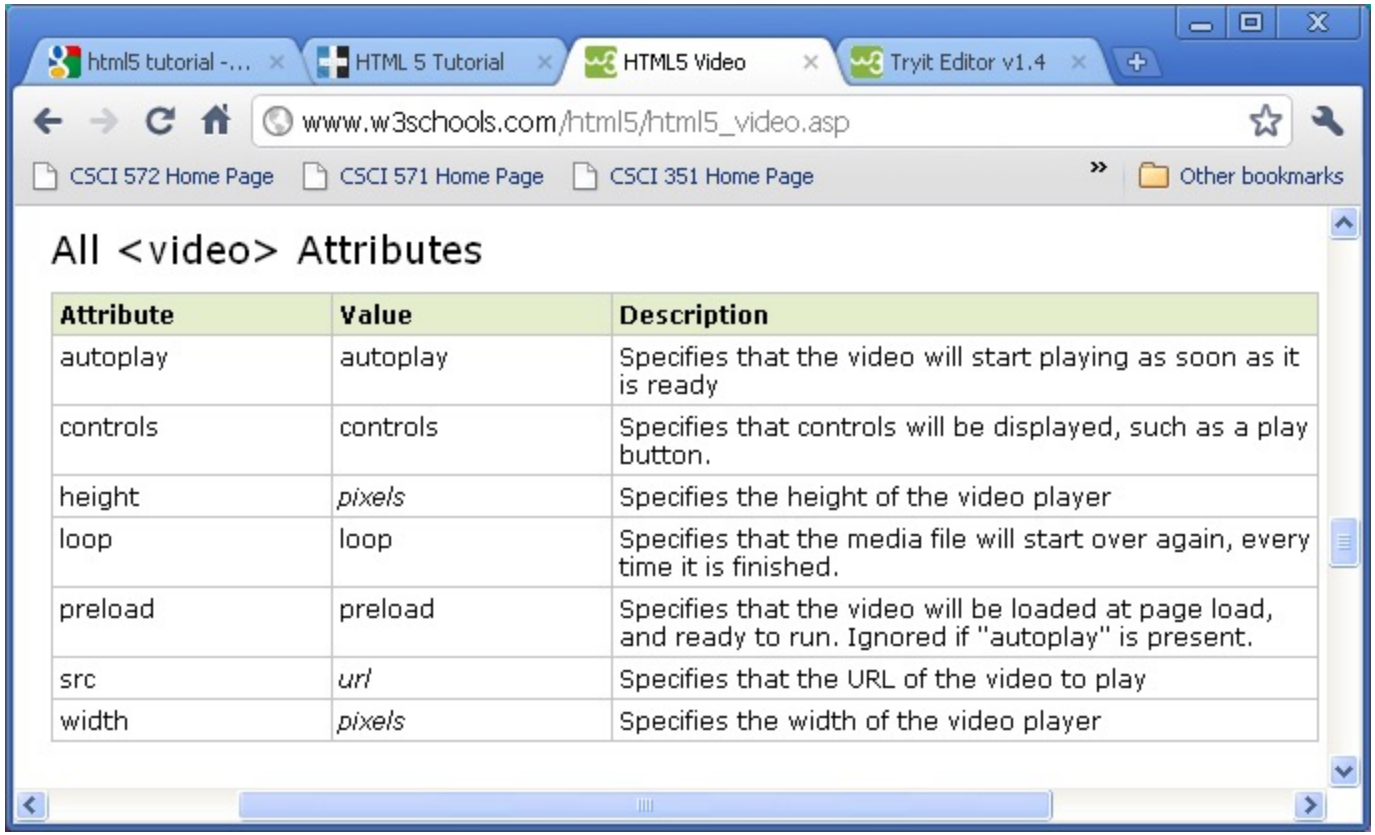
```
<video src="pr6.webm" width="320" height="240" autoplay></video>
```

HTML5 Video Example

http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_video_all



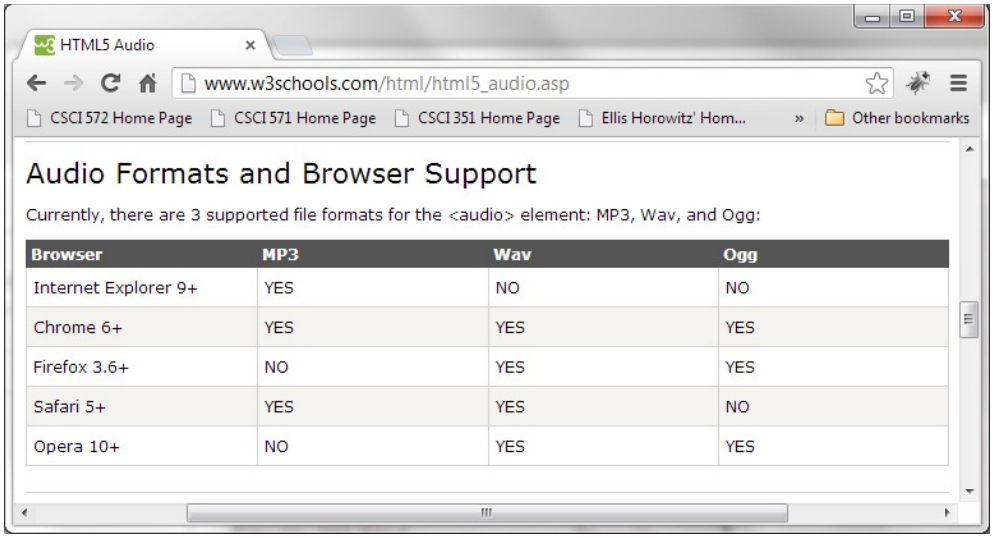
Video Attributes

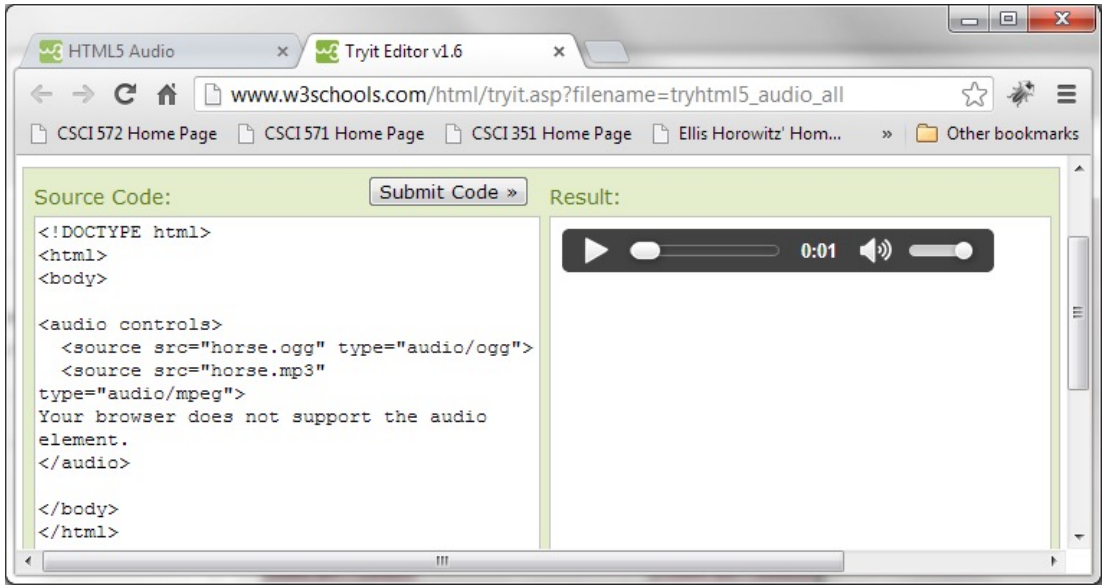


All <video> Attributes

Attribute	Value	Description
autoplay	autoplay	Specifies that the video will start playing as soon as it is ready
controls	controls	Specifies that controls will be displayed, such as a play button.
height	*pixels*	Specifies the height of the video player
loop	loop	Specifies that the media file will start over again, every time it is finished.
preload	preload	Specifies that the video will be loaded at page load, and ready to run. Ignored if "autoplay" is present.
src	*url*	Specifies that the URL of the video to play
width	*pixels*	Specifies the width of the video player

Audio

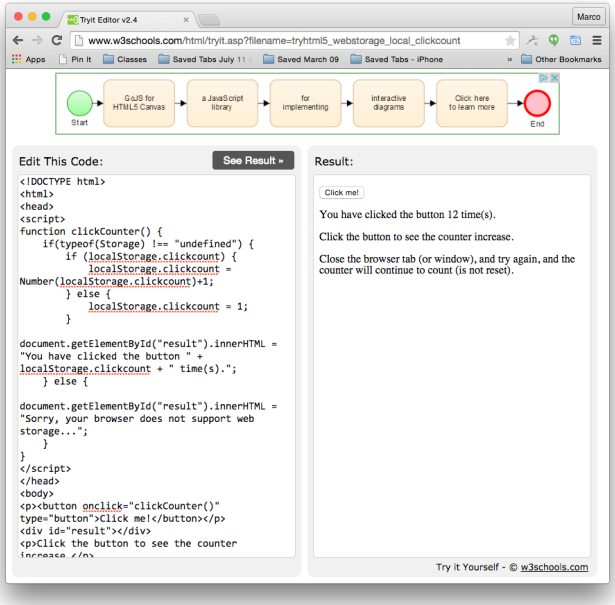

HTML5 Audio
www.w3schools.com/html/html5_audio.asp
CSCI 572 Home Page
CSCI 571 Home Page
CSCI 351 Home Page
Ellis Horowitz' Hom...
Other bookmarks
Audio Formats and Browser Support
Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg:
Browser | MP3 | Wav | Ogg
Internet Explorer 9+ | YES | NO | NO
Chrome 6+ | YES | YES | YES
Firefox 3.6+ | NO | YES | YES
Safari 5+ | YES | YES | NO
Opera 10+ | NO | YES | YES


HTML5 Audio
Tryit Editor v1.6
www.w3schools.com/html/tryit.asp?filename=tryhtml5_audio_all
CSCI 572 Home Page
CSCI 571 Home Page
CSCI 351 Home Page
Ellis Horowitz' Hom...
Other bookmarks
Source Code:
Submit Code »
Result:
<!DOCTYPE html>
<html>
<body>
<audio controls>
<source src="horse.ogg" type="audio/ogg">
<source src="horse.mp3"
type="audio/mpeg">
Your browser does not support the audio
element.
</audio>
</body>
</html>
0:01

localStorage

- localStorage is a client-side key-value database,
  - data is stored in the user's browser and remains there ***even across sessions*** (open/close browser)
  - data is only available when on that machine and in that browser.
  - localStorage is per browser not per computer.
  - localStorage only supports storing of strings
- Below is an example of local storage that counts the number of clicks on a button

```
<!DOCTYPE html>
<html>
<head>
<script>
function clickCounter() {
   if(typeof(Storage) !== "undefined") {
      if (localStorage.clickcount) {
         localStorage.clickcount = Number(localStorage.clickcount)+1;
      } else {
         localStorage.clickcount = 1; }
      document.getElementById("result").innerHTML = "You have clicked the button " +
      localStorage.clickcount + " time(s).";
   } else {
      document.getElementById("result").innerHTML = "Sorry, your
browser does not support web storage...";}
}</script>
</head>
<body>
<p><button onclick="clickCounter()" type="button">Click me!</button></p>
<div id="result"></div>
<p>Click the button to see the counter increase.</p><p>Close the browser tab (or window), and try again, and the counter will continue to count (is not
        reset).</p></body></html>
```



See: http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_webstorage_local_clickcount

Session Storage

- Session storage is designed for scenarios where the user is carrying out a single transaction but could be carrying out multiple transactions in different windows at the same time.
- To address this, this specification introduces the `sessionStorage` IDL attribute (An IDL attribute determines the behavior of script data). Sites can add data to the session storage, and *it will be accessible to any page from the same site opened in that window*. The sessionStorage object is equal to the localStorage object, **except** that it stores the data for only one session. **The data is deleted when the user closes the specific browser tab**.
- For example, a page could have a checkbox that the user ticks to indicate that he wants insurance:

```
<label>
<input type="checkbox" onchange="sessionStorage.insurance =
   checked ? 'true' : ''">
I want insurance on this trip.</label>
```

- A later page could then check, from script, whether the user had checked the checkbox or not:

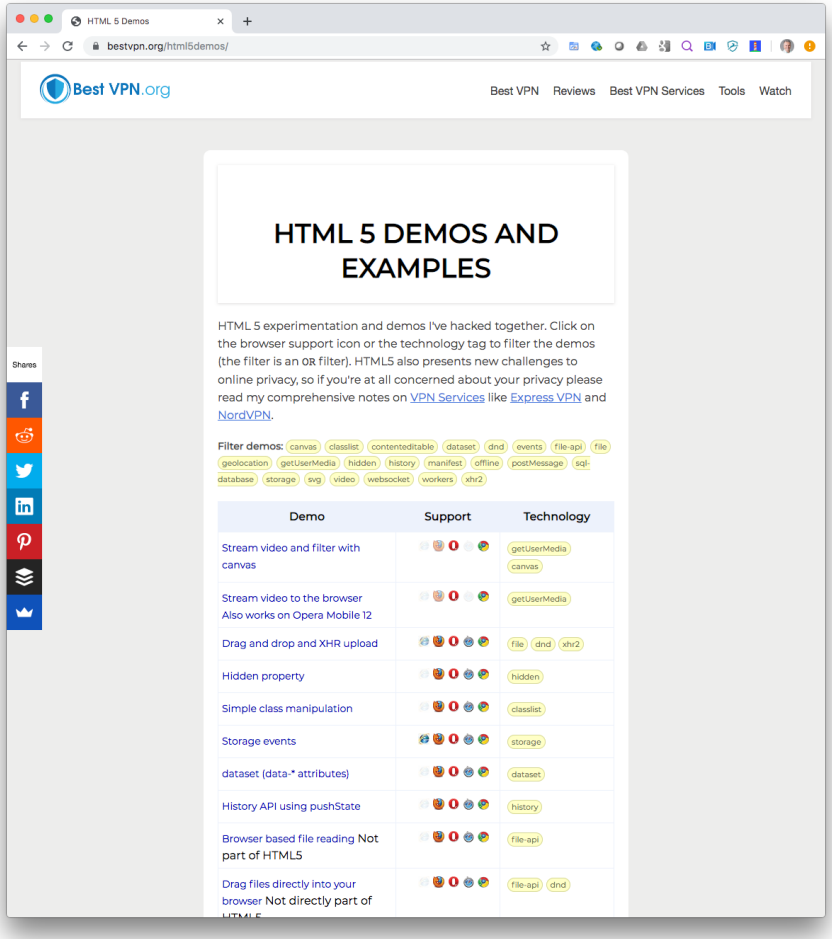
```
if (sessionStorage.insurance) { ... }
```

- If the user had multiple windows opened on the site, each one would have its own individual copy of the session storage object.

View Some New HTML5 Capabilities

- Go to https://bestvpn.org/html5demos/ and select your favorite browser, e.g., Firefox, and try these examples:
  - Drag files directly into your browser
    - Try it with an image
  - Interactive canvas gradients
    - Move mouse across gradient

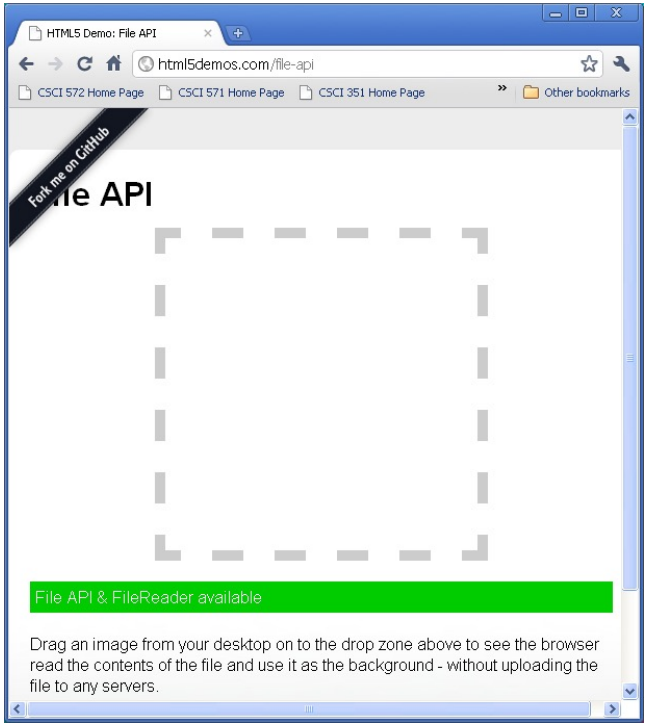
    Then View Source
  - Content editable
    - Edit some text and then restore
  - Geolocation
    - Try it



- More significant examples

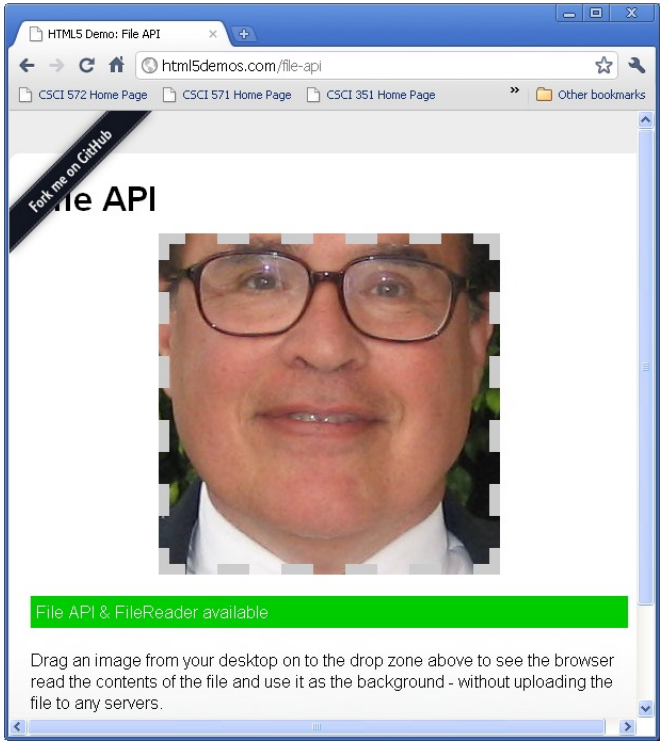
of content editable and geolocation can be found at https://csci571.com/examples.html#html5

Drag an Image from Desktop to drop Zone



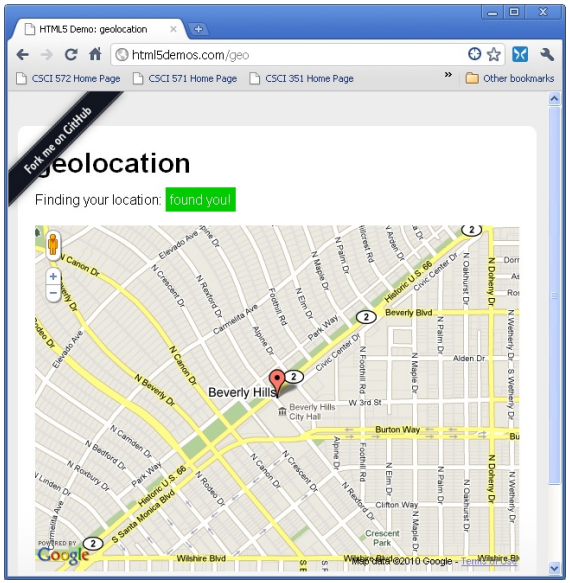
Before

Image

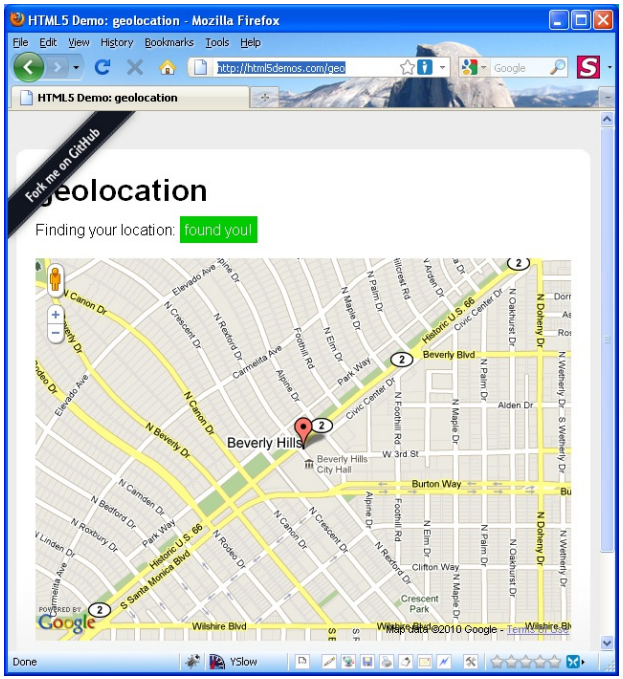


After

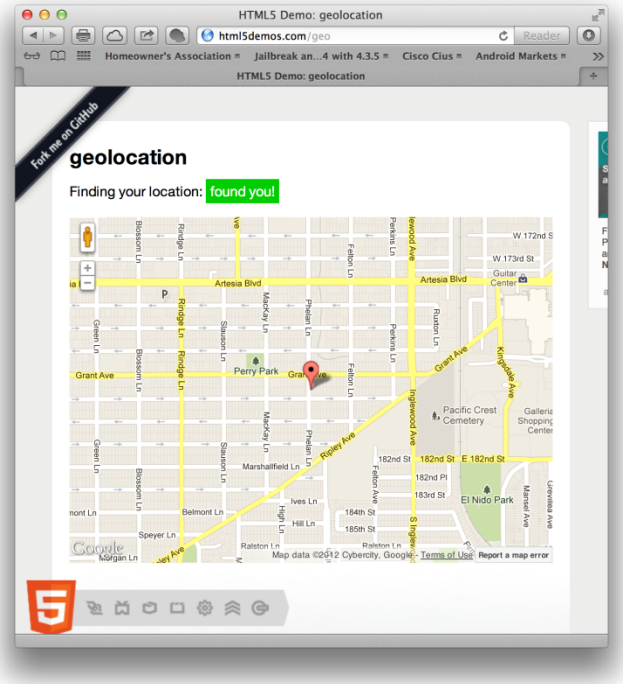
Geolocation Example



Chrome

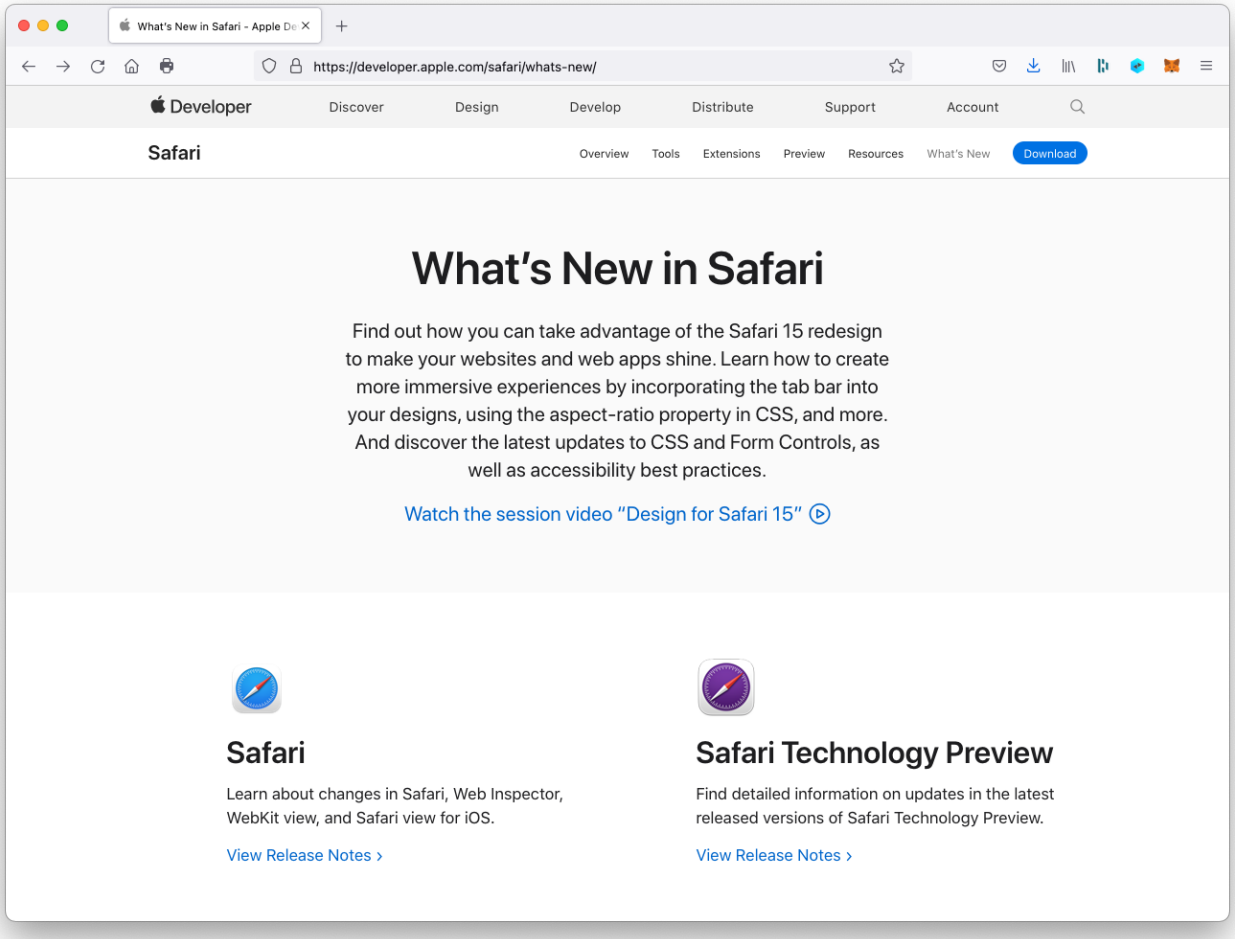


Firefox



Safari

Apple support for HTML5



https://developer.apple.com/safari/whats-new/

HTML5 Also Introduces Many New APIs

http://alebelcor.blogspot.com/2011/10/html5-apis.html

HTML5 Logo

http://www.w3.org/html/logo/

More Articles on the Flash vs HTML5 Controversy

- Adobe to More Aggressively Contribute to HTML5

http://blogs.adobe.com/flashplatform/2011/11/flash-to-focus-on-pc-browsing-and-mobile-apps-adobe-to-more-aggressively-contribute-to-html5.html

- Comparison of speed of Flash and HTML5

http://www.appleinsider.com/articles/10/03/10/flash_html_5_comparison_finds_neither_has_performance_advantage.html

- SproutCore:

http://www.appleinsider.com/articles/10/04/19/sproutcore_debuts_new_html5_web_development_tools.html

https://sproutcore.com/

More HTML5 fun....

- http://www.chromeexperiments.com/
- http://www.creativebloq.com/web-design/examples-of-html-1233547
- https://www.noupe.com/essentials/freebies-tools-templates/40-beautiful-free-html5-css3-templates.html
- http://html5gallery.com
- https://www.juicebox.net/demos/
- http://wowslider.com/html5-gallery-puzzle-collage-demo.html

video element (added many more attributes):
https://html.spec.whatwg.org/#video

- audio element:
https://html.spec.whatwg.org/#audio