HTML5 Graphics, Media and APIs

www.instigate-training-center.am

Ինստիգեյթ
Ուսումնական Կենտրոն

Topics

- HTML5 Canvas
- HTML5 SVG
- HTML5 Video
- HTML5 Audio
- HTML5 Geolocation
- HTML5 Drag/Drop
- HTML5 Web Storage
- HTML5 App Cache
- HTML5 Web Workers
- HTML5 SSE

HTML5 Canvas

- The HTML5 <canvas> element is used to draw graphics, on the fly, on a web page.
- The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.
- Canvas has several methods for drawing paths, boxes, circles, text, and adding images.
- A canvas is a rectangular area on an HTML page, and it is specified with the <canvas> element.
- Always specify an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas.
- All drawing on the canvas must be done inside a JavaScript:
  - Functions of JavaScript used in conves: getContext("2d"), fillRect(), moveTo(), lineTo(), arc(), stroke(), fillText(), strokeText(), createLinearGradient(), addColorStop()
  - methods of the getContext("2d") object is more and we don't need to remember them.

HTML5 Inline SVG

- What is SVG?
  - SVG stands for Scalable Vector Graphics
  - SVG is used to define vector-based graphics for the Web
  - SVG defines the graphics in XML format
  - SVG graphics do NOT lose any quality if they are zoomed or resized
  - Every element and every attribute in SVG files can be animated
  - SVG is a W3C recommendation
- Advantages of using SVG over other image formats (like JPEG and GIF):
  - SVG images can be created and edited with any text editor
  - SVG images can be searched, indexed, scripted, and compressed
  - SVG images are scalable
  - SVG images can be printed with high quality at any resolution
  - SVG images are zoomable (and the image can be zoomed without degradation)

HTML5 Inline SVG

- Differences Between SVG and Canvas
  - SVG is a language for describing 2D graphics in XML.
  - Canvas draws 2D graphics, on the fly (with a JavaScript).
- SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.
- In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.
- Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.

HTML5 Video

- Many modern websites show videos. HTML5 provides a standard for showing them.
- Before HTML5, there was no standard for showing videos/movies.
- Before HTML5, videos could only be played with a plug-in (like flash). However, different browsers supported different plug-ins.
- HTML5 defines a new element which specifies a standard way to embed a video or movie on a web page: the <video> element.
- HTML5 has DOM methods, properties, and events for the <video> and <audio> elements.
- These methods, properties, and events allow you to manipulate <video> and <audio> elements using JavaScript.
- There are methods for playing, pausing, and loading. There are also DOM events that can notify you when the <video> element begins to play, is paused, is ended, etc.

HTML5 Video (cont.)

- <video> tag specifies video, such as a movie clip or other video streams.
- Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg
- Any text between the <video> and </video> tags will be displayed in browsers that do not support the <video> element.
- Optional Attributes:
  - autoplay - Specifies that the video will start playing as soon as it is ready
  - controls- Specifies that video controls should be displayed
  - poster - Specifies an image to be shown while the video is downloading
  - height, width, loop, muted, preload, src

HTML5 Audio

- HTML5 provides a standard for playing audio files.
- Before HTML5, audio files had to be played with a plug-in. However, different browsers supported different plug-ins.
- HTML5 defines a new element which specifies a standard way to embed an audio file on a web page: the <audio> element.
- The control attribute adds audio controls, like play, pause, and volume.
- The <audio> element allows multiple <source> elements. <source> elements can link to different audio files. The browser will use the first recognized format.
- Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg
- <audio> tag defines sound, such as music or other audio streams.
- Attributes: autoplay, controls, loop, muted, preload, src

HTML5 Geolocation

- HTML5 Geolocation API is used to get the geographical position of a user.
- Since this can compromise user privacy, the position is not available unless the user approves it.
- Use the getCurrentPosition() method to get the user's position.
- This page demonstrated how to show a user's position on a map. However, Geolocation is also very useful for location-specific information. Examples:
  - Up-to-date local information
  - Showing Points-of-interest near the user
  - Turn-by-turn navigation (GPS)

HTML5 Drag and Drop

- Drag and drop is a part of the HTML5 standard.
- Drag and drop is a very common feature. It is when you "grab" an object and drag it to a different location.
- In HTML5, drag and drop is part of the standard, and any element can be draggable.
- To make an element draggable, set the draggable attribute to true:
  - <img draggable="true">
- Then, specify what should happen when the element is dragged.
  - ondragstart and setData()
- The ondragover event specifies where the dragged data can be dropped.
- When the dragged data is dropped, a drop event occurs.

HTML5 Web Storage

- With HTML5, web pages can store data locally within the user's browser.
- Earlier, this was done with cookies. However, Web Storage is more secure and faster. The data is not included with every server request, but used ONLY when asked for. It is also possible to store large amounts of data, without affecting the website's performance.
- The data is stored in name/value pairs, and a web page can only access data stored by itself.
- Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.
- HTML5 Web Storage provides two new objects for storing data on the client:
  - window.localStorage - stores data with no expiration date
  - code.sessionStorage - stores data for one session (data is lost when the tab is closed)

HTML5 Application Cache

- HTML5 introduces application cache, which means that a web application is cached, and accessible without an internet connection.
- Application cache gives an application three advantages:
  - Offline browsing - users can use the application when they're offline
  - Speed - cached resources load faster
  - Reduced server load - the browser will only download updated/changed resources from the server
- To enable application cache, include the manifest attribute in the document's <html> tag: <html manifest="demo.appcache">
- Every page with the manifest attribute specified will be cached when the user visits it. If the manifest attribute is not specified, the page will not be cached (unless the page is specified directly in the manifest file).
- The recommended file extension for manifest files is: ".appcache"

HTML5 Application Cache (cont.)

- The manifest file is a simple text file, which tells the browser what to cache (and what to never cache).
- The manifest file has three sections:
  - CACHE MANIFEST - Files listed under this header will be cached after they are downloaded for the first time
  - NETWORK - Files listed under this header require a connection to the server, and will never be cached
  - FALLBACK - Files listed under this header specifies fallback pages if a page is inaccessible
- Once an application is cached, it remains cached until one of the following happens:
  - The user clears the browser's cache
  - The manifest file is modified (see tip below)
  - The application cache is programmatically updated
- Updating the date and version in a comment line is one way to make the browser re-cache your files.

HTML5 Web Workers

- When executing scripts in an HTML page, the page becomes unresponsive until the script is finished.
- A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.
- Before creating a web worker, check whether the user's browser supports it:
  - if(typeof(Worker) !== "undefined")
- When we have the web worker file, we need to call it from an HTML page.
- The following lines checks if the worker already exists, if not - it creates a new web worker object and runs the code in "demo_workers.js":
  - if(typeof(w) == "undefined") {

    w = new Worker("demo_workers.js");}
- When a web worker object is created, it will continue to listen for messages (even after the external script is finished) until it is terminated.

HTML5 SSE

- SSE = Server-Sent Events - One Way Messaging
- A server-sent event is when a web page automatically gets updates from a server.
- This was also possible before, but the web page would have to ask if any updates were available. With server-sent events, the updates come automatically.
- Examples: Facebook/Twitter updates, stock price updates, news feeds, sport results, etc.
- The EventSource object is used to receive server-sent event notifications:
  - var source = new EventSource("server.php");
- Before this need to check browser support for server-sent events:
  - if(typeof(EventSource) !== "undefined")

Ինստիգեյթ
Ուսումնական Կենտրոն

Thank You

Ինստիգեյթ
Ուսումնական Կենտրոն