Școala
informală
de IT

Advanced BOM & HTML

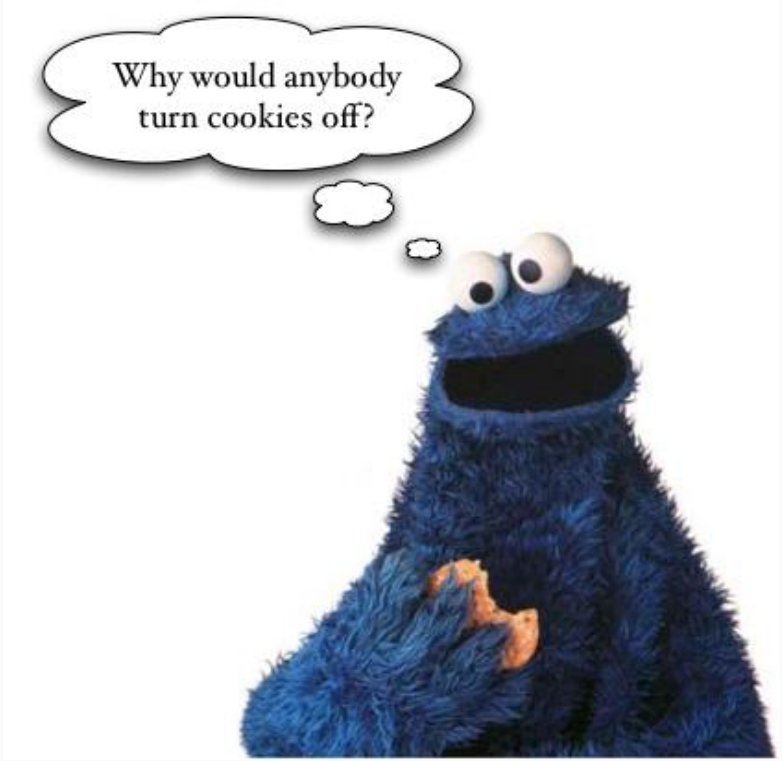
Agenda

- Recap
- More APIs: Cookies, Web Storage API, Geolocation
- Media (Audio, Video)
- SVG & Canvas

Recap

Last session's topics

- Window
- Screen
- Timing
- Location
- History



Cookies

Reminder: HTTP cookie

- = small piece of data that a server sends to the user's web browser, that may store it and send it back with the next request to the same server
- Remembers stateful information for the stateless HTTP protocol
- 3 main purposes:
  - Session management (user logins, shopping carts)
  - Managing user preferences
  - Tracking (analyzing user behavior)

Document.cookie

- Gets and sets the cookies associated with the current document
- Get all cookies:

```
allCookies = document.cookie;
```

- Write a new cookie:

```
document.cookie = newCookie;
```

- Attributes:
  - path, domain, max-age (in seconds), expires (date in GMTString format), secure

Hands on Cookies

1. Set a cookie on a document
2. Read all cookies
3. Add two radio buttons with two available languages (e.g., en-US, ro-RO)
   - The one whose value equals cookie's value should be "pre-selected"
   - When the user selects the other radio button, his option should be preserved in the cookie

Web Storage API

Web Storage API

- provides mechanisms by which browsers can securely store key/value pairs
- 2 mechanisms:
  - Session storage
    - maintains a separate storage area for each given origin that's available for the duration of the page session
    - Window.sessionStorage
  - Local storage
    - does the same thing, but persists even when the browser is closed and reopened
    - Window.localStorage

Web Storage Basics

- Objects stored as key-value pairs
  - Keys and values are always strings!
- The values can be accessed using:
  - getItem()
  - setItem()
  - removeItem()
  - clear()
  - key()
  - length

```
var aValue = storage.getItem(keyName);
storage.setItem(keyName, keyValue);
storage.removeItem(keyName);
storage.clear();
var aKeyName = storage.key(keyIndex);
var aLength = storage.length;
```

- Feature detection in browser

Web Storage Hands-On

- Look at the storage in browser
- Add two radio buttons with two available languages (e.g., en-US, ro-RO).
  - Store the value of this preference in local storage, if it's available
  - The one whose value equals the stored value should be "pre-selected"
  - When the user selects the other radio button, his option should be saved in the local storage
  - Save this preference in a cookie as fallback (use the previous cookie hands on)

Geolocation API

Geolocation API

- Used to programmatically obtain the position of the device
- Note: the user is notified and asked to grant permission
- Uses the navigator.geolocation object with the following methods:
  - getCurrentPosition()
  - watchPosition()
  - clearWatch()

```
navigator.geolocation.getCurrentPosition(success[, error[, options]]);
var watchId = navigator.geolocation.watchPosition(success[, error[,
options]]);
navigator.geolocation.clearWatch(watchId);
```

Media: Audio & Video

Audio & Video on Web Pages - Short History

- Flash & Silverlight
  - Disadvantages:
    - Didn’t work well with HTML/CSS features
    - security issues
    - accessibility issues
- Native solution: <audio> and <video> tags
- Note: OVP (Online Video Providers) like YouTube, Vimeo offer easy ways of hosting and consuming videos

Video Tag

```
<video src="myvideo.webm" controls>
  <p>Your browser doesn't support HTML5 video. Here is a
    <a href="myvideo.webm">link to the video</a> instead.</p>
</video>
```

- src = path to the source of the video
- controls = set of controls for the video, including at minimum a way to start and stop the media, and to adjust the volume
- Fallback content (the <p>) will be displayed if the browser accessing the page doesn't support the <video> element

Video Formats & Features

- Different formats needs to be provided for the video to be played in all browsers
- More features: *autoplay, loop, muted, poster*

```
<video controls width="400" height="400"
       autoplay loop muted
       poster="poster.png">
  <source src="myvideo.mp4" type="video/mp4">
  <source src="myvideo.webm" type="video/webm">
  <p>Your browser doesn't support HTML5 video. Here is a
    <a href="myvideo.mp4">link to the video</a> instead.</p>
</video>
```

Audio Tag

- Very similar to the video element
- Doesn’t support *width, height, poster*

```
<audio controls>
  <source src="myaudio.mp3" type="audio/mp3">
  <source src="myaudio.ogg" type="audio/ogg">
  <p>Your browser doesn't support HTML5 audio. Here is a
    <a href="myaudio.mp4">link to the audio</a> instead.</p>
</audio>
```

Canvas & SVG

Canvas

- <canvas> tag
- exposes a surface where you can create and manipulate rasterized images pixel by pixel using a JavaScript programmable interface
- examples: draw graphs, make photo composition, animations

```
HTML: <canvas id="myCanvas" width="800" height="800"></canvas>
JS:
    var canvas = document.getElementById('myCanvas');
    if (canvas.getContext) {
      var ctx = canvas.getContext('2d');
      ctx.fillStyle = '#c00';
      ctx.fillRect(10, 10, 50, 50);
    }
```

Scalable Vector Graphics

- <svg> tag
- SVG is an XML-based language for describing two-dimensional graphics
- Being scalable, you *increase or decrease a vector image while maintaining its crispness and high quality*

HTML:

```
<svg xmlns="http://www.w3.org/2000/svg" viewbox="0 0 600 600">
  <desc>Red rectangle shape</desc>
  <rect x="10" y="10" width="50" height="50" fill="#c00"  />
</svg>
```

Canvas	SVG
Pixel-based	Object Model-based
Single HTML element similar to <img> in behavior	Multiple graphical elements which become part of the DOM
Visual presentation created and modified programmatically through script	Visual presentation created with markup and modified by CSS or programmatically through script
Event model/user interaction is at the canvas element only; interactions must be manually programmed from mouse coordinates	Event model/user interaction is object-based at the level of primitive graphic elements (lines, rectangles)
API does not support accessibility	SVG markup and object model directly supports accessibility

Homework

Homework

- The user can have a preference whether to see the weather is C or in F.
- Store this in the local storage and use it to display the temperature for the user based on the stored value.
- Use cookies if local storage is not supported in the browser

Resources

- https://www.html5rocks.com/en/tutorials/offline/storage/
- https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API
- https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API
- https://developer.mozilla.org/en-US/docs/Web/API/Geolocation/Using_geolocation
- https://developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia_and_embedding/Video_and_audio_content
- https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial
- https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Getting_Started
- https://www.sitepoint.com/canvas-vs-svg-choosing-the-right-tool-for-the-job/
- https://www.sitepoint.com/how-to-choose-between-canvas-and-svg/