CMSC434 TUTORIAL #4
HTML
CSS
JavaScript
Jquery
Ajax
HTML5
Mobile WebApp
A new standard for HTML (since the last release of HTML4 in 1999)
It has rules and goals...
• HTML, CSS, DOM, and JavaScript
• Reduce the need for external plugins (like Flash)
• Better error handling
• Device independent

HTML5 can do...
• **CANVAS**; Drawing graphic elements on canvas
• **VIDEO, AUDIO**; for media playback (no plug in needed)
• **LOCALSTORAGE**; store data on client machine
• **ARTICLE, FOOTER, HEADER, NAV, SECTION**; content specific tags
• **CALENDAR, DATE, TIME, EMAIL, URL, and SEARCH**; new form controls.
<video>

<table>
<thead>
<tr>
<th>Format</th>
<th>IE</th>
<th>Firefox</th>
<th>Opera</th>
<th>Chrome</th>
<th>Safari</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ogg</td>
<td>No</td>
<td>3.5+</td>
<td>10.5+</td>
<td>5.0+</td>
<td>No</td>
</tr>
<tr>
<td>MPEG 4</td>
<td>9.0+</td>
<td>No</td>
<td>No</td>
<td>5.0+</td>
<td>3.0+</td>
</tr>
<tr>
<td>WebM</td>
<td>No</td>
<td>4.0+</td>
<td>10.6+</td>
<td>6.0+</td>
<td>No</td>
</tr>
</tbody>
</table>

- Ogg = Ogg files with Theora video codec and Vorbis audio codec
- MPEG4 = MPEG 4 files with H.264 video codec and AAC audio codec
- WebM = WebM files with VP8 video codec and Vorbis audio codec

</video>

<audio>

<table>
<thead>
<tr>
<th>Format</th>
<th>IE 9</th>
<th>Firefox 3.5</th>
<th>Opera 10.5</th>
<th>Chrome 3.0</th>
<th>Safari 3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ogg Vorbis</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MP3</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Wav</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

</audio>

</audio>

Your browser does not support the video tag.

Your browser does not support the audio element.
<canvas id="myCanvas" width="200" height="100"></canvas>

...<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.fillStyle="#FF0000";
cxt.fillRect(0,0,150,75);
</script>

- create <canvas> element first – with specific width and height.
- $$\$(\"#myCanvas\")$$ will give an array containing one element. Thus, use $$\$(\"#myCanvas\")[0]$$ instead.

context.moveTo(x,y);
context.lineTo(x,y);
context.lineWidth = 20;
context.strokeStyle = "#ff0000";
context.stroke();

• Full-fledged tutorial of canvas technique [http://www.html5canvastutorials.com/]
```javascript
var centerX = 288;
var centerY = 160;
var radius = 75;
var startingAngle = 1.1 * Math.PI;
var endingAngle = 1.9 * Math.PI;
var counterclockwise = false;
context.arc(centerX, centerY, radius, startingAngle, endingAngle, counterclockwise);
context.lineWidth = 15;
context.strokeStyle = "black";
context.stroke();
```
context.moveTo(188, 150);
    var controlX = 288;
    var controlY = 0;
    var endX = 388;
    var endY = 150;

    context.quadraticCurveTo(controlX, controlY, endX, endY);
context.lineWidth = 10;
context.strokeStyle = "black"; // line color
context.stroke();
context.beginPath();
   context.moveTo(100, 20);
   context.lineTo(200, 160); // line 1
   context.quadraticCurveTo(230, 200, 250, 120); // quadratic curve
   context.bezierCurveTo(290, -40, 300, 200, 400, 150); // bezier curve
   context.lineTo(500, 90); // line 2

context.lineWidth = 5;
context.strokeStyle = "#0000ff";
context.stroke();
context.beginPath(); // begin custom shape
  context.moveTo(170, 80);
  context.bezierCurveTo(130, 100, 130, 150, 230, 150);
  context.bezierCurveTo(250, 180, 320, 180, 340, 150);
  context.bezierCurveTo(420, 150, 420, 120, 390, 100);
  context.bezierCurveTo(430, 40, 370, 30, 340, 50);
  context.bezierCurveTo(320, 5, 250, 20, 250, 50);
context.closePath(); // complete custom shape
context.lineWidth = 5;
context.fillStyle = "#8ED6FF";
context.fill();
context.strokeStyle = "#0000ff";
context.stroke();
context.save(); // save current context to come back later
context.beginPath();
context.arc(centerX, centerY, radius, 0, 2 * Math.PI, false);
context.clip();

// draw blue circle inside clipping region
context.beginPath();
context.arc(centerX - offset, centerY - offset, radius, 0, 2 * Math.PI, false);
context.fillStyle = "#00D2FF"; // light blue
context.fill();

// draw yellow circle inside clipping region
...

// draw red circle inside clipping region
...

// draw border around clipping region
// restores the canvas context to its original state
// before we defined the clipping region
context.restore();
context.beginPath();
context.arc(centerX, centerY, radius, 0, 2 * Math.PI, false);
context.lineWidth = 3;
context.strokeColor = "black";
context.stroke();
```javascript
var destX = 69;
var destY = 50;
var imageObj = new Image();
imageObj.onload = function(){
    context.drawImage(imageObj, destX, destY);
}; // callback function that will run after loading finishes
imageObj.src = "darth-vader.jpg"; // start loading image file
```
Define name-source location pairs of images as javascript object

Call image loader with a custom callback function that will draw images on canvas after loading finishes

Count total # of images first.

Set callback functions for each image loading process, so that it trigger the drawing function when # of loaded images is same or bigger than total # of images.
Canvas WebGL with three.js

three.js examples

http://mrdoob.github.com/three.js/
Canvas WebGL showcase

http://www.html5canvastutorials.com/demos/advanced/html5_canvas_quadratic_motion_animation/
localStorage

```javascript
localStorage.setItem("name", "Hello World!");
//saves to the database, key/value
document.write(localStorage.getItem("name"));
//Hello World!
localStorage.removeItem("name");
//deletes the matching item from the database
```

- client-side database
- key-value storage (like JSON, dictionary or HashMap)
- Firefox, Safari and IE have one’s own localStorage (can’t be shared)
- Different web sites have different storages
- Why not cookie or sessionStorage?
  - localStorage keeps data even after the browser is closed
  - multiple windows share the same storage
  - Cookies are passed on by every request to the server (slow and inefficient)
- Where to use it?
  - Remember/Recall user specific status without any server-side DB
    - bring user to the last page logged out
    - restore previous draft of the task
  - Multiple-session tasks are possible even for guest users
Form Attributes

- Autocomplete shows previous input values matching with the current input value
  - Across different web pages.
  - Input box name specific.

```html
<form action="demo_form.asp" method="get" autocomplete="on">
    First name: <input type="text" name="fname" />
    Last name: <input type="text" name="lname" />
    E-mail: <input type="email" name="email" autocomplete="off" />
    <input type="submit" />
</form>
```

<table>
<thead>
<tr>
<th>Attribute</th>
<th>IE</th>
<th>Firefox</th>
<th>Opera</th>
<th>Chrome</th>
<th>Safari</th>
</tr>
</thead>
<tbody>
<tr>
<td>autocomplete</td>
<td>8.0</td>
<td>3.5</td>
<td>9.5</td>
<td>3.0</td>
<td>4.0</td>
</tr>
<tr>
<td>autofocus</td>
<td>No</td>
<td>4.0</td>
<td>10.0</td>
<td>3.0</td>
<td>4.0</td>
</tr>
<tr>
<td>form</td>
<td>No</td>
<td>4.0</td>
<td>9.5</td>
<td>10.0</td>
<td>No</td>
</tr>
<tr>
<td>form overrides</td>
<td>No</td>
<td>4.0</td>
<td>10.5</td>
<td>10.0</td>
<td>No</td>
</tr>
<tr>
<td>height and width</td>
<td>8.0</td>
<td>3.5</td>
<td>9.5</td>
<td>3.0</td>
<td>4.0</td>
</tr>
<tr>
<td>list</td>
<td>No</td>
<td>4.0</td>
<td>9.5</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>min, max and step</td>
<td>No</td>
<td>No</td>
<td>9.5</td>
<td>3.0</td>
<td>No</td>
</tr>
<tr>
<td>multiple</td>
<td>No</td>
<td>3.5</td>
<td>11.0</td>
<td>3.0</td>
<td>4.0</td>
</tr>
<tr>
<td>novalidate</td>
<td>No</td>
<td>4.0</td>
<td>11.0</td>
<td>10.0</td>
<td>No</td>
</tr>
<tr>
<td>pattern</td>
<td>No</td>
<td>4.0</td>
<td>9.5</td>
<td>3.0</td>
<td>No</td>
</tr>
<tr>
<td>placeholder</td>
<td>No</td>
<td>4.0</td>
<td>11.0</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>required</td>
<td>No</td>
<td>4.0</td>
<td>9.5</td>
<td>3.0</td>
<td>No</td>
</tr>
</tbody>
</table>

C: adds constraints to input value for validation purpose
Mobile browser

• How to know which browser is being used??

Examples of navigator.userAgent:
[iPhone] Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1A543a Safari/419.3
[iPod] Mozilla/5.0 (iPod; U; CPU like Mac OS X; en) AppleWebKit/420.1 (KHTML, like Gecko) Version/3.0 Mobile/3A101a Safari/419.3
[iPad] Mozilla/5.0 (iPad; U; CPU OS 3_2 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) version/4.0.4 Mobile/7B367 Safari/531.21.10
[Android] Android - Mozilla/5.0 (Linux; U; Android 0.5; en-us) AppleWebKit/522+ (KHTML, like Gecko) Safari/419.3

if((navigator.userAgent.match(/iPhone/i)) ||
   (navigator.userAgent.match(/iPod/i)) ||
   (navigator.userAgent.match(/iPad/i))) {
   alert("iOS mobile device!");
} else if((navigator.userAgent.match(/Android/i)) {
   alert("Android mobile device!");
}
Mobile browser

• What-not-to-do on mobile browser
  • No Flash
  • No Cookies
  • No Frames
  • No Tables / Nested-Tables (some browsers don’t support it)
  • Use relative sizes (EM, Percentage) instead of absolute sizes (px, millimeters, or inches)
    • modern browser correctly scale up/down when ctrl+-/- is pressed whether the measure is px or EM.
    • Using EM makes CSS more maintainable – by changing base font size, developer can change all other measures in EM relatively.
    • Using percentage makes CSS more maintainable for different mobile browsers which have various window dimensions.
Mobile browser

• **Viewport metatag**
  
  • The part of the web page currently shown on-screen.
  • User may zoom in/out, pan and rotate viewport.
  • only exists on mobile browsers.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Another option; prevent pinch zoom-in/out

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no"/>
```
QUESTIONS.