



Web Advanced: Javascript APIs

“We will learn JavaScript properly. Then, we will learn useful design patterns. Then we will pick up useful tools for making cool things better.”

FALL 2018

SESSION #13

HTML5 APIs

jaink@newschool.edu

<https://canvas.newschool.edu/courses/1407281>

<https://classroom.github.com/classrooms/4280964>
5-parsons-web-advanced-javascript-fall-2018



RECAP

HTML5 DATA ATTRIBUTES

Allows for a more structured method for passing configs and attributes from html to javascript.

Used by all javascript/jQuery plugins to define settings for the plugin.

```
<div id="slideshow" data-type="carousel"
  data-transition="fade-in" data-auto-start="true">
  <div>Slide 1</div>
  <div>Slide 2</div>
</div>
```

```
let slide = document.getElementById("slideshow");
let type = slide.dataset.type;
let transition = slide.dataset.transition;
let auto_start = slide.dataset.autoStart;
```

```
// alternative generic way to get attributes:
let type = slide.getAttribute("data-type");
```

```
let type = slide.data("data-type");
```

HTML5 GEOLOCATION

Navigator contains current position information for the client and can be accessed through:

`navigator.geolocation`

-> returns a Geolocation object:

```
function youAreHere(position) {  
    console.log("position: ", position);  
}  
  
if(navigator.geolocation) {  
    navigator.geolocation.getCurrentPosition(youAreHere);  
}
```

Watch for changes:

```
function youHaveMoved(position) {  
    console.log("changed position: ", position);  
}  
  
if(navigator.geolocation) {  
    navigator.geolocation.watchPosition(youHaveMoved);  
}
```

Reference:

[https://developer.mozilla.org/en-US/docs/Web/API/Geolocation API](https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API)



HTML5 VIDEO/AUDIO

A clean and extensible approach to embedding media objects:

```
<audio src="assets/08_please.mp3" controls>
  Your browser does not support the audio
  element.
</audio>
```

Pre-html5:

```
<div>
  <object
classid="clsid:02bf25d5-8c17-4b23-bc80-d3488abddc
6b"
codebase="http://www.apple.com/qtactivex/qtplugin
.cab">
    <param name="src" value="movie.mp4" />
    <param name="autoplay" value="false" />
    <param name="controller" value="true" />

    <!--[if !ie] -->
      <object type="video/mp4" data="movie.mp4">
        <param name="controller" value="true" />
        <param name="autoplay" value="false" />
      </object>
    <!--[endif]-->
  </object>
</div>
```

NOW:



HTML5 VIDEO/AUDIO

Media attributes:

```
<video src="assets/Nearness_on_Vimeo.mp4" autoplay>
  Your browser does not support the video element.
</video>
```

```
<video src="assets/Nearness_on_Vimeo.mp4" controls>
  Your browser does not support the video element.
</video>
```

```
<video src="assets/Nearness_on_Vimeo.mp4" loop>
  Your browser does not support the video element.
</video>
```

```
<video src="assets/Nearness_on_Vimeo.mp4"
poster="assets/casa2_hero_1452814189.jpg">
  Your browser does not support the video element.
</video>
```

```
<video src="assets/Nearness_on_Vimeo.mp4"
preload="none">
  Your browser does not support the video element.
</video>
```



HTML5 VIDEO/AUDIO

Controlling media with Javascript:

```
let video = document.getElementsByTagName("video")[0];
video.play();
video.pause();
video.volume = 90;
video.muted = true;
video.loop = true;
```

Events:

```
video.addEventListener("pause", function(event) {
    console.log("video has been paused");
})

video.addEventListener("play", function(event) {
    console.log("video has been paused");
})

video.addEventListener("volumechange", function(event) {
    console.log("video volume has been changed");
})

video.addEventListener("timeupdate", function(event) {
    console.log("video timecode has changed");
})
```




HTML5 CANVAS

Canvas allows graphics to be drawn onto a web page in real time through JavaScript.

```
<canvas id="canvasDrawing1" width="200"  
height="100">
```

This browser doesn't support the canvas element.

```
</canvas>
```

```
let canvas =  
document.getElementById("canvasDrawing1");
```

Context:

An object containing all the methods used to draw onto and manipulate the canvas.

```
// 2D  
let context = canvas.getContext("2d");
```

```
// 3D  
let context3D = canvas.getContext("webgl");
```



HTML5 CANVAS - ADDING SHAPES

Create Shapes

```
// set defaults
context.fillStyle = "#0000cc"; // a blue fill color
context.strokeStyle = "#ccc"; // a gray stroke color
context.lineWidth = 4;
```

```
// draw
context.fillRect(10, 10, 100, 50);
```

```
// draw
context.strokeRect(10, 100, 100, 50);
```

```
// Lines
context.beginPath();
context.moveTo(20, 50);
context.lineTo(180, 50);
context.moveTo(20, 50);
context.lineTo(20, 90);
context.strokeStyle = "#c00";
context.lineWidth = 10;
context.stroke();
```

```
// ARCs
context.arc(200, 200, 30, 0, Math.PI * 2, false);
context.strokeStyle = "#ff0";
context.lineWidth = 4;
context.stroke();
```



HTML5 CANVAS - ADDING SHAPES

```
// text
context.fillStyle = "#cc0033"; // fill color
context.font = "bold 26px sans-serif";
context.fillText("Hello", 20, 200);
```

```
// Image
let img = document.createElement('img');
img.src = 'assets/no_image.gif';
img.addEventListener('load', function() {
    context.drawImage(img, 10, 10 );
});
```

```
//Transform
context.scale(1,2) // works on anything drawn
after
context.rotate(0.1*Math.PI)
context.translate(50,100)
```



HTML5 APIs

- Prefetch API
- Camera API
- Speech Synthesis API
- Geolocation API
- Fullscreen API
- etc...

A nice list:

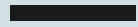
<https://github.com/diegocard/awesome-html5#virtual-reality>



EXAMPLES



Assignment



Next Steps

1

Plugins and Modules