Advanced HTML5 Media Controllers

In Safari on iOS and Mac OS X

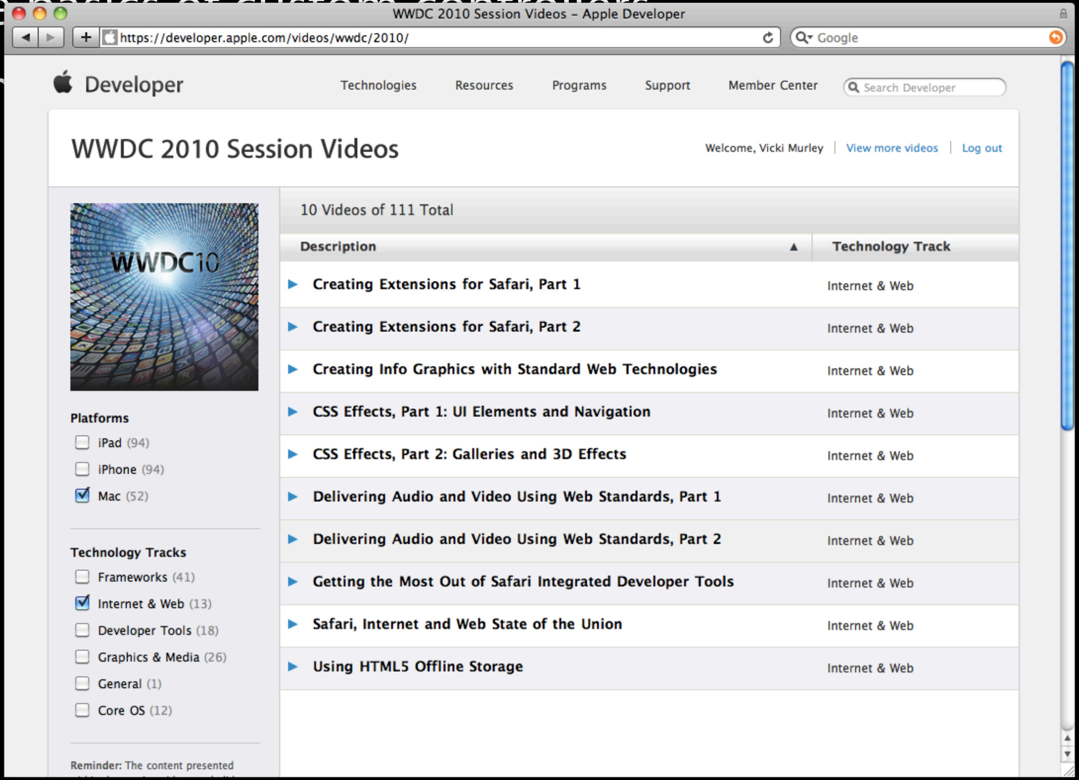
Session 502

Jer Noble
Safari and WebKit Engineer

Last Year, at WWDC…

https://developer.apple.com/videos/wwdc/2010/

- Covered the basics of custom controllers
- Available o



WWDC 2010 Session Videos – Apple Developer

https://developer.apple.com/videos/wwdc/2010/

Developer — Technologies | Resources | Programs | Support | Member Center | Search Developer

WWDC 2010 Session Videos

Welcome, Vicki Murley | View more videos | Log out

10 Videos of 111 Total

Description	Technology Track
Creating Extensions for Safari, Part 1	Internet & Web
Creating Extensions for Safari, Part 2	Internet & Web
Creating Info Graphics with Standard Web Technologies	Internet & Web
CSS Effects, Part 1: UI Elements and Navigation	Internet & Web
CSS Effects, Part 2: Galleries and 3D Effects	Internet & Web
Delivering Audio and Video Using Web Standards, Part 1	Internet & Web
Delivering Audio and Video Using Web Standards, Part 2	Internet & Web
Getting the Most Out of Safari Integrated Developer Tools	Internet & Web
Safari, Internet and Web State of the Union	Internet & Web
Using HTML5 Offline Storage	Internet & Web

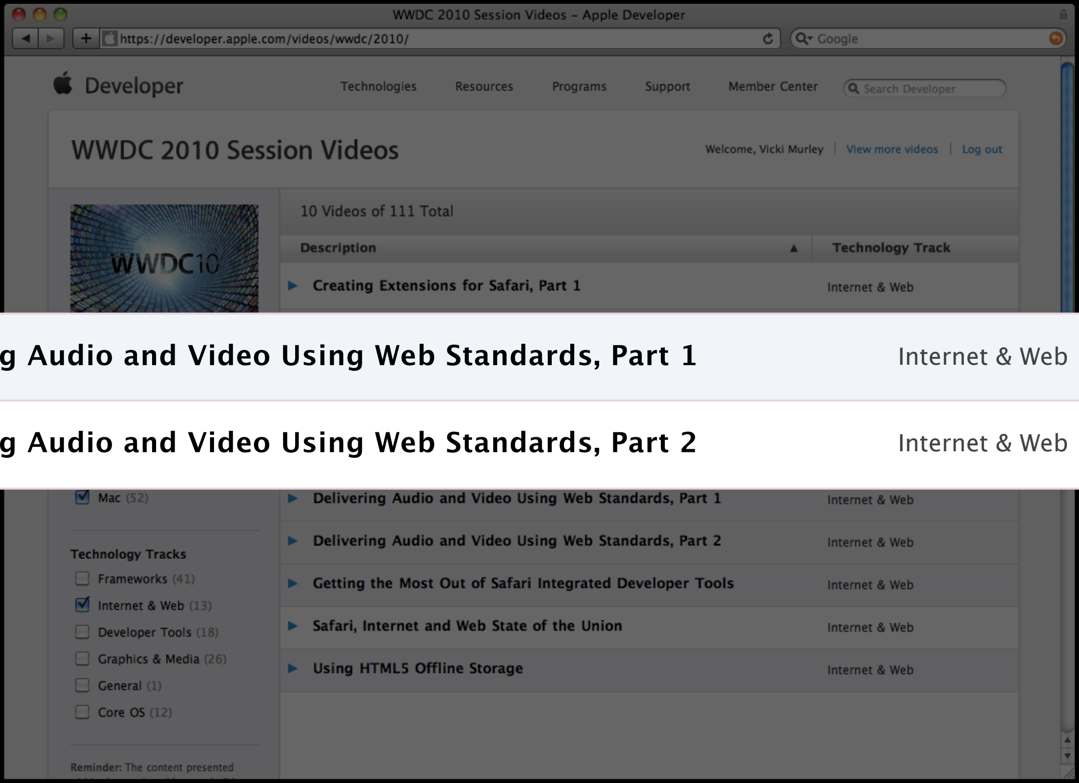
Platforms
- ☐ iPad (94)
- ☐ iPhone (94)
- ✓ Mac (52)

Technology Tracks
- ☐ Frameworks (41)
- ✓ Internet & Web (13)
- ☐ Developer Tools (18)
- ☐ Graphics & Media (26)
- ☐ General (1)
- ☐ Core OS (12)

Reminder: The content presented

Last Year, at WWDC…

https://developer.apple.com/videos/wwdc/2010/


WWDC 2010 Session Videos – Apple Developer

https://developer.apple.com/videos/wwdc/2010/

Developer — Technologies · Resources · Programs · Support · Member Center

WWDC 2010 Session Videos

Welcome, Vicki Murley | View more videos | Log out

10 Videos of 111 Total

Description	Technology Track
Creating Extensions for Safari, Part 1	Internet & Web
Delivering Audio and Video Using Web Standards, Part 1	Internet & Web
Delivering Audio and Video Using Web Standards, Part 2	Internet & Web
Getting the Most Out of Safari Integrated Developer Tools	Internet & Web
Safari, Internet and Web State of the Union	Internet & Web
Using HTML5 Offline Storage	Internet & Web

Mac (52)

Technology Tracks
- Frameworks (41)
- Internet & Web (13)
- Developer Tools (18)
- Graphics & Media (26)
- General (1)
- Core OS (12)

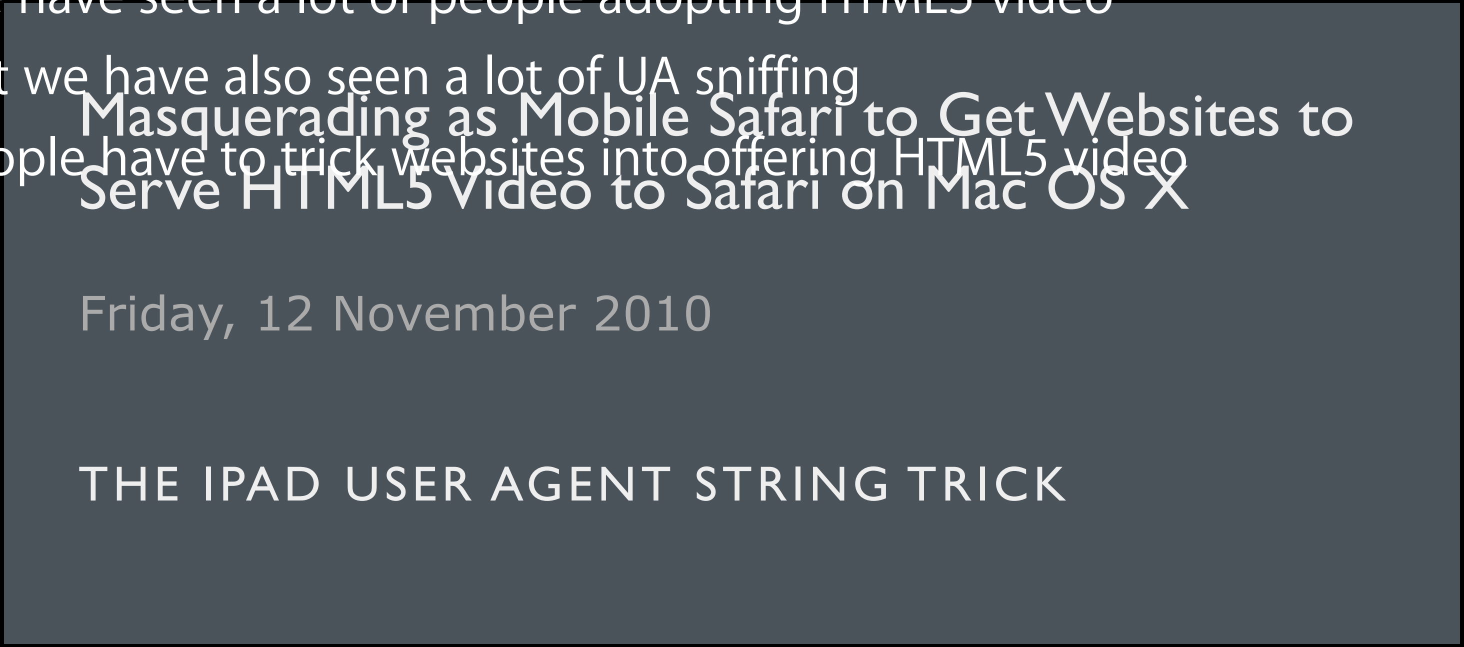
Reminder: The content presented

Delivering Audio and Video Using Web Standards, Part 1	Internet & Web
Delivering Audio and Video Using Web Standards, Part 2	Internet & Web

Last Year, at WWDC…

Since Last Year

- We have seen a lot of people adopting HTML5 video
- But we have also seen a lot of UA sniffing
- People have to trick websites into offering HTML5 video



Masquerading as Mobile Safari to Get Websites to Serve HTML5 Video to Safari on Mac OS X

Friday, 12 November 2010

THE IPAD USER AGENT STRING TRICK

Why Is This Happening?

Authors want…

- Controls with a specific look and feel
- Controls to work on all platforms
- Controls that do things the default controls cannot
- All this—and more—is possible with HTML5

What You’ll Learn

- How to create simple custom controls
- Controls for everyone
- Special effects with <video> elements

Your Custom <video> Controls

Getting started

The Basics

Media elements

- Methods

  `play()`

  `pause()`

- Properties

  `currentTime`

  `duration`

- Events

  `play`

  `pause`

  `timeupdate`

e.g.,

```
video = document.getElementById('video')
function onpause() {
  playButton.innerText = 'play';
}
video.addEventListener('pause', onpause,
  false);
  ontimeupdate, false);
```

The Basics

Play/Pause

```
<button onclick="togglePlay()">
<script>
  function togglePlay() {
    if (video.paused || video.ended)
      video.play();
    else
      video.pause();
  }
</script>
```

The Basics

Play/Pause

```
<script>
  function updatePlayButton() {
    if (video.paused || video.ended)
      playButton.value = "play";
    else
      playButton.value = "pause";
  }
  video.addEventListener("play", updatePlayButton, false);
  video.addEventListener("pause", updatePlayButton, false);
  video.addEventListener("ended", updatePlayButton, false);
</script>
```

The Basics

Scrub

```
<input type="range" min="0" max="1" step="any"
  onchange="setTime(this.value)">
<script>
function setTime(value) {
  video.currentTime = video.startTime + (value * video.duration);
}
</script>
```

The Basics

Scrub

```
<script>
  function updateTimeSlider() {
    timeSlider.value = (video.currentTime - video.startTime)
      / video.duration;
  }
  video.addEventListener("timeupdate", updateTimeSlider, false);
</script>
```

The Basics

Volume

- Caveat
  - On iOS, there is only one volume setting
  - The system volume can’t be changed from JavaScript

The Basics
Volume

```
<input type="range" min="0" max="1" step="any"
  onchange="setVolume(this.value)" style="display:none">
<script>
  function setVolume(value) { video.volume = value; }
  function volumeChanged() {
    volumeSlider.style.removeProperty('display');
    volumeSlider.value = video.volume;
  }
  video.addEventListener("volumechange", volumeChanged, false);
  video.volume = 0.5;
</script>
```

The Basics

Full-screen video

- Methods

```
webkitEnterFullScreen();
webkitExitFullScreen();
```

- Events

```
webkitbeginfullscreen
webkitendfullscreen
```

The Basics

Full-screen video

- Works in Safari on iOS, Mac OS X, and Windows
- Great controls for every platform
- Videos on iPhone will always play in full screen

The Basics

New

Full-screen any element

- Take any arbitrary element into a full-screen mode
- Complete DOM access
- See the proposed specification at: https://wiki.mozilla.org/Gecko:FullScreenAPI

The Basics

Element

New

- Methods

```
webkitRequestFullScreen();
```

e.g.,

```
div = document.getElementById('myVideo')
div.webkitRequestFullScreen();
```

The Basics

Document

New

- Properties
  - `webkitCurrentFullScreenElement`
  - `webkitIsFullScreen`
  - `webkitFullScreenKeyboardInputAllowed`
- Events
  - `webkitfullscreenchange`
- Methods
  - `webkitCancelFullScreen()`

e.g.,

```
if (div == document.
  webkitCurrentFullScreenElement) {
  document.webkitCancelFullScreen();
}
  }, true);
```

The Basics

Styles

New

- Pseudo classes

```
-webkit-full-screen
-webkit-full-screen-document
```

e.g.,

```
:-webkit-full-screen-document {
  overflow: scroll;
}
  position: absolute;
  left: 50%;
  bottom: 0;
  margin: -250px;
}
```

The Basics

New

Full-screen any element

- Requirements
  - User interaction
  - <iframe>s must have `webkitallowfullscreen` attribute

The Basics

New

Full-screen any element

```
<button onclick="toggleFullScreen()" value="Full-screen">
<script>
  function toggleFullScreen() {
    if (document.webkitCurrentFullScreenElement == video
      document.webkitCancelFullScreen();
    else
     video.webkitRequestFullScreen();
  }
</script>
```

The Basics

New

Full-screen any element

```
<script>
  function fullScreenChanged() {
    if (document.webkitCurrentFullScreenElement == video
      fullScreenButton.value = "Exit Full-screen";
    else
      fullScreenButton.value = "Enter Full-screen";
  }
  document.addEventListener("webkitfullscreenchange",
    fullScreenChanged, false);
</script>
```

The Basics

New

Full-screen any element

- Fall-back

```
<script>
  function toggleFullScreen() {
   if (typeof(element.webkitRequestFullScreen) == function)
     newToggleFullScreen();
   else
     oldToggleFullScreen();
 }
</script>
```

The Basics

Full-screen any element

New

- Caveat
  - This spec is still in development
  - Most keyboard input is disabled
  - No event or error if a full-screen request is denied

Demo

Eric Carlson
Safari/WebKit Engineer

Your Custom <video> Controls

Accessibility

Creating an Accessible Video Experience

Why?

- Reach a larger audience
- Legal compliance
- It’s the right thing to do!

Creating an Accessible Video Experience

Touch events

- Tablets are all like, “What’s a mouse?”
- Touch events

  `touchstart`

  `touchend`

  `touchmove`

  `touchcancel`
- Gesture events

  `gesturestart`

  `gesturechange`

  `gestureend`

Creating an Accessible Video Experience

VoiceOver

- Screen reader built into OS X and iOS
- Support is built into Safari as well
- Accessible Rich Internet Applications (ARIA) specification

Creating an Accessible Video Experience

VoiceOver

- Nonaccessible

```
<div onclick="togglePlay()">Play</div>
<div style="width: 25%" id="progress"></div>
```

- Accessible

```
<div onclick="togglePlay()" role="button">Play</div>
<div style="width: 25%" id="progress" role="progressBar"></div>
```

Creating an Accessible Video Experience

VoiceOver

- Many more roles are available, e.g.,

```
role="application"
role="slider"
```

- ARIA specification can be found at:

  http://www.w3.org/TR/wai-aria/

Demo

Eric Carlson
Safari/WebKit Engineer

Your Custom <video> Controls

Subtitles

Creating an Accessible Video Experience

Subtitles

- Active work in the World Wide Web Consortium
- Good news, everyone
  - The WHATWG has proposed a subtitle file format!
  - WebVTT
  - http://www.whatwg.org/specs/web-apps/current-work/webvtt.html
  - http://www.w3.org/TR/html5/rendering.html
  - No browser support yet
  - But…

Creating an Accessible Video Experience

Subtitles

- JavaScript "polyfill" libraries
  - Captionator—https://github.com/cgiffard/Captionator
  - Playr—http://www.delphiki.com/html5/playr/
  - Videojs—http://videojs.com/
- Other JavaScript subtitle libraries
  - Popcorn.js—http://popcornjs.org/

Creating an Accessible Video Experience

Subtitles

- <track> element

```
<video controls>
   <source src="video.m4v" type="video/m4v">
   <source src="video.webm" type="video/webm">
</video>
```

Creating an Accessible Video Experience

Subtitles

- <track> element

```
<video controls>
   <source src="video.m4v" type="video/m4v">
   <source src="video.webm" type="video/webm">
   <track kind="subtitles" src="english.vtt" srclang="en" label="English">
   <track kind="subtitles" src="spanish.vtt" srclang="es" label="Spanish">
</video>
```

Creating an Accessible Video Experience

TextTrack

- Properties
  - `kind`
  - `label`
  - `language`
  - `readyState`
  - `mode`
  - `cues`
  - `activecues`
- Events
  - `cuechange`

e.g.,

```
if (track.activeCues.length > 0)
  setCurrentCue(track.activeCues[0]);
```

Creating an Accessible Video Experience

TextTrackCue

- Properties
  - `track`
  - `id`
  - `startTime`
  - `endTime`
  - `pauseOnExit`
- Events
  - `enter`
  - `exit`
- Methods
  - `getCueAsSource()`
  - `getCueAsHTML()`

e.g.,

```
var cue = track.activeCues[0];
div.innerText = cue.getCueAsSource();
  lastCue.pauseOnExit = true;
}
```

Creating an Accessible Video Experience

MutableTextTrack

- Methods

  `addCue()`

  `removeCue()`

Creating an Accessible Video Experience

Text tracks

- Not just about captions
- Searchable
- Translatable
- Timing is key

Demo

Eric Carlson
Safari/WebKit Engineer

More Information

Vicki Murley
Safari Technologies Evangelist
vicki@apple.com

Documentation
Safari Dev Center
http://developer.apple.com/devcenter/safari/

Apple Developer Forums
http://devforums.apple.com

Related Sessions

HTTP Live Streaming Update	Nob Hill Tuesday 4:30PM
Understanding and Optimizing Web Graphics	Marina Wednesday 3:15PM
Combining Web Accessibility and Automation on iOS	Nob Hill Friday 10:15AM

Labs

HTML5 Audio and Video Lab	Internet and Web Lab A Tuesday 4:30PM
HTTP Live Streaming Lab	Graphics, Media & Games Lab D Wednesday 9:00AM
Safari Open Lab Lab	Internet and Web Lab B Wednesday 9:00AM

Summary

- Awesome controls
- Full screen
- Subtitles
- Video for everyone

Q&A

WWDC2011