The HTML5 & CSS3 Landscape

Chris Mills, Opera Software, all the way from “sunny” Manchester, UK
Slides available

At my.opera.com/ODIN
(search for “chrismills” tag)
I work for Opera
Open web standards evangelist
Technologist
Tech writer

and GENERAL DOGSBODY
What we'll cover
HTML5 history
HTML5 purpose
HTML5 things we can use today
CSS3 purpose
CSS3 things we can use today
HTML5 history

HTML5 purpose

HTML5 things we can use today

CSS3 purpose

CSS3 things we can use today
A brief history of HTML

HTML first proposed 1989-91
HTML2 first standardised in 1995
HTML 4.01 standardised in 1999
Corrections submitted 2001
blah blah blah...
HTML5 history

HTML5 started 2004 by WHAT-WG
Adopted by W3C 2008
Still being argued about
Still being developed by both!
What does this tell us??

What wisdom can we glean from this?
History is boring!

This technology has been around for a long time!
HTML5 purpose

CSS3 purpose

HTML5 things we can use today

CSS3 things we can use today
Evolving...

There is nothing wrong with HTML 4
...Evolved!

But HTML5 is much more feature-rich!
HTML5 doesn't replace HTML4

It fills up holes
Adds new markup + APIs
Adds more semantics
Competes with proprietary tech
Isn't backwards incompatible
Competition in mind
Ian Hickson has already said as much. HTML5 will directly compete with other web application technologies, like Flash and Silverlight.
Competition in mind

"Silverlight has only one calorie — not evil enough..."
HTML5 features

More accurate semantics (eg `<header>`, `<footer>`)  
Better forms (built in validation!)  
`<video>`  
`<canvas>`
HTML5 features

Drag and drop
Web workers
Web storage, app cache, webdb
...and more
HTML5 things we can use today

HTML5 history
HTML5 purpose

CSS3 things we can use today
CSS3 purpose
New syntax: better semantics
HTML5 doctype

<!DOCTYPE html>
Typical blog structure

```html
<div id="header"

<div id="sidebar"

<div id="content"

<div class="post"

<div class="post"

<div class="post"

<div id="footer"
```
HTML5 blog structure

<header>
<nav>
<section id="content">
<article>
<article>
<article>
<footer>
Unambiguous & machine readable

<time datetime="2010-06-27">27 June 2010</time>
<time datetime="2010-06-27">Chris's 32nd birthday</time>
<time datetime="2010-06-27T020:00Z">8PM on my birthday</time>
<time datetime="2010-06-27T020:00+09:00">8PM on my birthday—in Tokyo</time>
Other syntax rules
Abstracts more away from the developer
Attribute quotes not usually needed
Even the `<head>`, `<body>`, etc. are optional ;-)
HTML5 forms
Previously called “Web forms 2.0”
More powerful form elements
Built-in validation
More standard archetypes
Slider

<input type=range>
Calendar widget

<input type=date>
URL picker, E-mail input

<input type=url>

<input type=email>
Client-side validation
Was horrible in HTML4...
function validate() {
    var str = "";
    var elements = document.getElementsByTagName('input');

    // loop through all input elements in form
    for(var i = 0; i < elements.length; i++) {

        // check if element is mandatory; ie has a pattern
        var pattern = elements.item(i).getAttribute('pattern');
        if (pattern != null) {
            var value = elements.item(i).value;

            // validate the value of this element, using its defined pattern
            var offendingChar = value.match(pattern);

            // if an invalid character is found or the element was left empty
            if(offendingChar != null || value.length == 0) {

                // add up all error messages
                str += elements.item(i).getAttribute('errorMsg') + "\n" +
                        "Found this illegal value: " + offendingChar + "' \n";

                // notify user by changing background color, in this case to red
                elements.item(i).style.background = "red";
            }
        }
    }

    if (str != "") {
        // do not submit the form
        alert("ERROR ALERT!!\n" +str);
        return false;
    } else {
        // form values are valid; submit
        return true;
    }
}
HTML5 built-in validation

<input type=email required>
Autofocus

<input type=email required autofocus>
HTML5 `<canvas>`

Scriptable graphics
Standard API for drawing
Supported in most browsers
The basics

<canvas id="canvas" width="400" height="300">
...fallback...
</canvas>
The basics

```javascript
var ctx = document.getElementById('canvas').getContext('2d');

ctx.fillStyle
ctx.fillRect
```
Example time!
nihilologic.dk has cool stuff on it
dev.opera.com has good articles
HTML5 `<video>` (& `<audio>`) New tags, plus new API for controlling audio and video!
The old school way

<object width="425" height="344">
<param name="movie" value="http://www.example.com/v/LtfQg4KkR88&hl=en&fs=1"></param>
<param name="allowFullScreen" value="true"></param>
<embed src="http://www.example.com/v/LtfQg4KkR88&hl=en&fs=1"
  type="application/x-shockwave-flash"
  allowfullscreen="true" width="425" height="344"></embed>
</object>
The badass sexy new way...

<video></video>
...more functions

<video src="video.ogv"
  controls
  autoplay
  loop
  poster="poster.jpg"
  preload="none"
  width="320" height="240">
  <a href="video.ogv">Download movie</a>
</video>
Native `<video>` is awesome

Works well with open standards
Built-in keyboard accessibility
API for customising controls, etc.
DOESN'T require plugins
Circumvents EOLAS patent BS
<video> problems

Disagreements on what formats to use — Ogg Theora, H264?

Still need to provide fallbacks
Different sources

<video width=640 height=480 controls>
<source src="bruce_henny.ogv" type="video/ogg"/>
<source src="bruce_henny.mp4" type="video/mp4"/>
If you're not using a browser that can display either the open Ogg Theora or the patent-encumbered H.264 codec, there's not much to see here.
</video>
<video> plays nicely with CSS, JavaScript, etc.

Just another block-level element. So you can do what you want with it.

API allows easy customization
<video> accessibility

Built-in captioning?
Currently not ;-(
You can build a workaround though
Hello, Good Evening and Welcome
Tonight on the Jeremy Kyle show
...
Hello, Good Evening and Welcome.

Tonight on the Oprah Winfrey show ...
Hello, Good Evening and Welcome.
function timeupdate() {
    var v = document.querySelector('video');
    var now = v.currentTime;  
}

<video width=600 src=synergy.ogv
    ontimeupdate=timeupdate()>

Browser support?

Supported across most major browsers (forms only in Opera)…
Browser support?
Browser support?

Fake-able in IE using JS:
Dean Edwards' HTML5 library
Excanvas
SVG Web and Raphael JS for SVG
etc.
CSS3 purpose

HTML5 history
HTML5 purpose
HTML5 things we can use today

CSS3 purpose
CSS3 things we can use today
CSS3...

Introduces more powerful functionality
Standard design patterns
Less maintenance
Less time spent in Photoshop
CSS3 things we can use today

HTML5 history

HTML5 purpose

HTML5 things we can use today

CSS3 purpose

CSS3 things we can use today
text-shadow

It is very easy to set text shadows using CSS 3.

text-shadow: #444 2px 2px 2px;

Let’s have a look at a slightly demonic example!

text-shadow: 0 0 4px white,
0 -5px 4px #ff3,
2px -10px 6px #fd3,
-2px -15px 11px #f80,
2px -25px 18px #f20;
box-shadow: 10px 10px 15px #000000;
CSS3 opacity

```
p { opacity: 0.9; }
p { opacity: 0.7; }
```
CSS3 colours: rgba(a)

#p1 { background-color: rgba(255,0,0,0.95); }  
#p2 { background-color: rgba(255,0,0,0.85); }  
#p3 { background-color: rgba(255,0,0,0.75); }  
#p4 { background-color: rgba(255,0,0,0.65); }
CSS3 colours: hsl(a)

```css
#p5 { background-color: hsl(0, 100%, 50%); }
#p6 { background-color: hsl(0, 100%, 60%); }
#p7 { background-color: hsl(0, 100%, 70%); }
#p8 { background-color: hsl(0, 100%, 80%); }
```
border-radius

Finally, Web 2.0 is easy!!

01. border-radius: 10px;

01. border-radius: 10px 0px 10px 0px;

(Starts from top-left corner)
Transitions

Offer animation-like abilities
Set a default state for the element
Choose property & duration
Then set state to transition to
Transition default state

p#transition1 {
  background-color: #ff0000;
  -o-transition-property: background-color;
  -o-transition-duration: 2s;
}
Transitioned state

p#transition1:hover {
  background-color: #ffffff;
}

Transitions: easing

Allows you to control the pattern of acceleration/deceleration. More natural feel.

-o-transition-timing-function: ease-in;
Transitions: delay
Add a delay before the transition starts.

-o-transition-delay: 1s;
Multiple transitions

Multiple transitions, each with their own start time.

-o-transition-property: background-color, width, height;
-o-transition-duration: 4s, 8s, 5s;
Transforms (2D)
Transforming element position, size, etc.: moving, rotating, skewing...
Setting transform origin
For example what point does your element rotate around?

-o-transform-origin: 3em bottom;
Moving elements

In X and Y directions

-o-transform: translateX(50px);
-o-transform: translateY(100px);
Resizing elements
By a set scale factor

-o-transform: scale(2.5);
Skewing elements
Squishy distortion!

-o-transform: skew(10deg, 20deg);
Rotating elements
Around the origin point

-o-transform: rotate(30deg);
Combining transforms

Do multiple things in one declaration

```css
-o-transform: scale(2) rotate(45deg) translate(80px);
```
Combining transitions with transforms...
...is where it starts to get really fun.
background-clip

background-clip: border-box;
background-clip: padding-box;
background-clip: content-box;
border-image

Apply background images just to borders

border-image: url(border.png) 27 27 27 27 round round;
Web Fonts
Download custom fonts along with your web pages
Solve the web typographer's nightmare?
Include the font

@font-face {
  font-family: "My font";
  src: url("http://www.myweb.com/fonts/myfont.ttf")
       format("truetype");
}

Use it in your page as normal

p { font-family: "My font gothic"; }
Web Fonts issues

Good free fonts are available, but... Many are not licensed for the Web. Some also mean large downloads. Some solutions are being explored (such as TypeKit).
Media queries

You know what media types are

Media queries take the idea further

Apply CSS depending on device attributes
Device attributes

Browser window width/height
Device width/height
Resolution
Aspect ratio
Monochromacity
etc.
Essential for “One Web”

Most obvious use case is varying layout for different screen sizes.
CSS3 attribute selectors #1

<a href="mailto:cmills@opera.com">E-mail link</a>

```css
a[href^="mailto:"] {
    background: url(i/mail.jpeg) no-repeat right center;
    padding-right: 30px;
}
```

E-mail link
CSS3 attribute selectors #2

<a href="http://amazon.co.uk">British link</a>

a[href$=".co.uk"] {  
  background: url(i/uk.png) no-repeat left center;  
  padding-left: 35px;  
}
CSS3 attribute selectors #3

<a href="#" title="this title has chris in it">link about Chris</a>

a[title*="chris"] {  
  background: url(i/heart.jpeg) no-repeat left center;
  padding-left: 30px;
}

link about Chris
Attribute selector + :before

<a href="#" title="this title has chris in it">link about Chris</a>

a[title*="chris"]:before {
  content: url(i/heart.jpeg);
}
:nth-child

Before

tr.even  {background-color: red;}
  <tr>...
  <tr class="even">...

Now

tr:nth-child(even)  {background-color: red;}
  <tr>...
  <tr>...
  <tr>...
...aaaand there's more!

Multiple background images
Multiple column layout
CSS animations
3D transforms
etc.
Training resources available

Opera web standards curriculum: www.opera.com/wsc

Opera developer site: dev.opera.com
Training resources available

WaSP InterAct: interact.webstandards.org

Course structures, rubrics, assignments, etc.
All you need to teach the Web.
“The book of the film”

“Interact with Web Standards”: interactwithwebstandards.com
Holistic view of web design
Written for education
Thanks!

cmills@opera.com
@chrisdavidmills

Check out dev.opera.com
Check out html5doctor.com
Check out css3.info