



Firefox



# HTML5 Everywhere

Making a Web page work everywhere

---

Telerik Software Academy

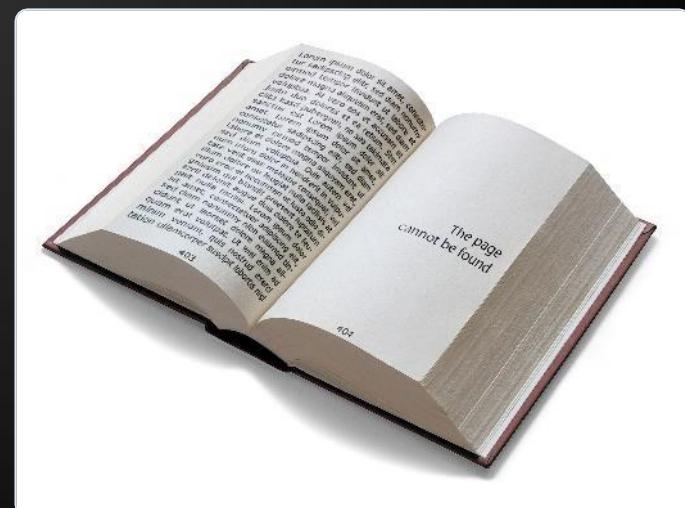
Learning & Development Team

<http://academy.telerik.com>



# Table of Contents

- ◆ HTML5 and the Web today
  - ◆ What is HTML5?
  - ◆ Browser support
- ◆ Backward and mobile
  - ◆ Support for older browsers and mobile
- ◆ Shims and Shims
- ◆ Polyfills
- ◆ Modernizr
- ◆ CSS boiler plates



# HTML5 and The Web Today

Making the fine web



# HTML5 and The Web Today

- ◆ The Web today is vast
  - ◆ Lots of ways to reach the web (mobile, PC, TV)
  - ◆ Lots of operating systems
    - ◆ Windows, Mac OSX, Android, Firefox OS, Linux
  - ◆ Lots of browsers with version specifics
    - ◆ Google Chrome, Mozilla Firefox, Apple Safari, Internet Explorer, Opera
  - ◆ Lots of web browser engines
    - ◆ Gecko ,Presto\*, Trident, WebKit, Blink
  - ◆ Engines act differently on each OS or browser

- ◆ Browsers dive into HTML5 with different steps
  - ◆ Most major browsers for desktop and mobile support primary HTML5 specifics
- ◆ Some browser vendors thought it through
  - ◆ Firefox and Chrome update silently
  - ◆ Opera and Safari offer to download and update
  - ◆ Internet Explorer is hardest to update
    - ◆ Update from browser (can be forbidden)
    - ◆ Use Windows Update

# HTML5 Support (2)

- ◆ Yet waiting for users to update their browser is not reliable
  - ◆ Sometimes users are afraid or unwilling to update
  - ◆ Sometimes administrators restrict updates due to corporate policies
  - ◆ Sometimes the OS doesn't support newer versions
    - ◆ The best XP users get is Internet Explorer 8!

- ◆ What then?

- ◆ Use something that is supported everywhere?
  - ◆ Like HTML4/XHTML
  - ◆ Not good enough
- ◆ Don't support Internet Explorer?
  - ◆ Still not OK...
  - ◆ IE holds good ~25% of the browsers share for 2013
- ◆ Make your site look just as good on older browsers, and good enough on the oldest
  - ◆ Using a bit of JavaScript

# Backward Compatibility

Looking good on most browsers

# Backward Compatibility

- ◆ How to make a web page look good on most browsers?
- ◆ Two common approaches
  - Limit the HTML features so every browser can understand them
  - Use the top features of modern web design and do a little JavaScript to leverage the platform

# Backward Compatibility (2)

- ◆ How to use the features of HTML 5 in browsers back to IE6?
  - Need to use something to flatten the browser capabilities
- ◆ HTML5 browser support is far from complete and some users still tend to use IE7/IE8
  - There are lots of gaps in there
- ◆ Shims, Shims and Polyfills to the rescue!



# Shims, Shivs and Polyfills

Fill the gaps in browsers

# Shims and Shibs

- ◆ Shims and shibs are a small libraries that fill gaps in a platform
  - Mainly used when the API changes and causes compatibility issues
  - The older API can still be supported by a thin compatibility layer on top of the newer code

- ◆ Polyfills are a similar concept to shims/shivs
  - They provide fallback functionality for features still not implemented natively in the browser
  - When the users update their browser they will have the same experience, but use the native implementation



# Shims, Shibs and Polyfills

- ◆ Shibs and Shims make HTML5 things happen in less-supported browsers
- ◆ They fill the gaps in:
  - ◆ HTML - markup features, new tags, meta tags, attributes, etc...
  - ◆ CSS - style properties, gradients, fonts embedding, data-urls, border radius etc...
  - ◆ JavaScript - storages, web workers, canvas, SVG, etc...

- ◆ **html5shiv.js is the most used shiv for HTML5 elements**
  - It enables styling for HTML5 elements in older versions of IE (IE8-)
  - <https://code.google.com/p/html5shiv/>

# html5shiv

Live Demo



- ◆ CSS 3 Pie is a polyfill for Internet Explorer
  - ◆ It enables CSS3 features in older versions of IE
- ◆ How does it work?
  - ◆ Download pie.js and pie.htc
  - ◆ Include pie.js into the Web page
  - ◆ Add the following to the styles

```
behavior: url(scripts/css3-pie/pie.htc);
```



# CSS3 Pie Shim

Live Demo

# Array.indexOf() shim

- ◆ Some browsers do not support Array.indexOf()
  - ◆ Need a shim to make it happen

```
if (!Array.prototype.indexOf) {  
    function arrayIndexOf (searchElement, fromIndex) {  
        for (var i = fromIndex || 0; i < this.length; i++) {  
            if (this[i] === searchElement) {  
                return i;  
            }  
        }  
        return -1;  
    }  
    Array.prototype.indexOf = arrayIndexOf;  
}
```

# Array.indexOf Shim

Live Demo

- ◆ Playing audio or video with pure HTML
  - ◆ Available since HTML5
- ◆ The most common approach is to use external plugin to fallback the audio/video support
  - ◆ Like Flash or Silverlight

# Audio Polyfill

Live Demo



**Modernizr**  
FRONT-END DEVELOPMENT DONE RIGHT



# What is Modernizr.js?

JS Library to Detect Native HTML5 Features

- ◆ Modernizr is a JavaScript library that detects HTML5 and CSS3 features in the browser
  - ◆ <http://modernizr.com/download/>
- ◆ Modernizr has three primary tasks
  - ◆ Enables html5shiv if necessary
  - ◆ Detects html 5 support through adding classes to the HTML element
    - ◆ Class js for "js is supported" and "no-js" otherwise
  - ◆ YepNope loading
    - ◆ If a feature is not supported - load a polyfill

[Download Modernizr 2.6.2](#)

Use the [Development version](#) to develop with and learn from. Then, when you're ready for production, use the build tool below to pick only the tests you need.

CSS3

- @font-face
  - background-size
  - border-image



GENERATE

## HTML5

- applicationCache
  - Canvas
  - Canvas Text



[DOWNLOAD](#)

Misc.

- Geolocation API
  - SVG
  - Touch Events

```
/* Modernizr 2.6.2 (Custom Build) | MIT & BSD
 * Build: http://modernizr.com/download/#-shiv-cssclasses-load
 */
>window.Modernizr=function(a,b,c){function u(a){}.cssText=a}function v(a,b){return u(prefixes.join(a+";"+(b||"")))}function w(a,b){return typeof a===""?b:function x(a,b){return!!~(a+").indexOf(b))}function y(a,b,d){for(var e in a){var f=b[a[e]];if(f!=c){return d==!=!1?a[e]:w(f,"function")?f.bind(d)||b:f;}}return!1}var d="2.6.2",e={},f=0,g=b.documentElement,h="modernizr",i=b.createElement(h),j=i.style,k={};i.toString,m={},n={},o=p[],q=o.slice,r={};hasOwnProperty:t||s."undefined"&&!t(s.call."undefined")?t=function(a,b){return s.call(a,b)}:t=function(a,b){return b in
```

## ■ Don't Minify Source

# Installing and Configuring Modernizr

# Installing Modernizr

## ◆ Steps to install Modernizer:

1. Go to <http://modernizr.com/download/>
2. Select features you want to use
3. Generate and download your customized Modernizr JS code

### Download Modernizr 2.6.2

Use the [Development version](#) to develop with and learn from. Then, when you're ready for production, use the build tool below to pick only the tests you need.

**CSS3** TOGGLE

- @font-face
- background-size
- border-image

**HTML5** TOGGLE

- applicationCache
- Canvas
- Canvas Text

**Misc.** TOGGLE

- Geolocation API
- SVG
- Touch Events

**GENERATE!** **DOWNLOAD** Custom Build

# Modernizr: Detecting HTML5 Features

```
function get_location() {
    if (Modernizr.geolocation) {
        navigator.geolocation.getCurrentPosition(show_map);
    } else {
        // no native support; maybe try a fallback?
    }
}
```

# Detecting HTML5 Features

- ◆ Modernizer can check for HTML5 / CSS3 features through JavaScript:

```
if (Modernizr.canvas) {  
    alert('HTML5 canvas is supported');  
} else {  
    alert('HTML5 canvas is NOT supported');  
}
```

- ◆ Features detected by Modernizr:
  - ◆ <http://modernizr.com/docs/#s2>

```
function get_location() {  
    if (Modernizr.geolocation) {  
        navigator.geolocation.getCurrentPosition(show_map);  
    } else {  
        // no native support; maybe try a fallback?  
    }  
}
```

# Modernizr: Detecting HTML5 Features

Live Demo

# Modernizr: Detecting CSS3 Features & Fallbacks

## CSS3 PROPERTIES

			
	CHROME	FIREFOX	OPERA
	5	3.6	10
RGBA	✓	✓	✓
HSLA	✓	✓	✓
Multiple Backgrounds	✓	✓	✗

```
9  label {  
10     display: block;  
11     position: relative;  
12 }  
13  
14 .pointerevents label:before {  
15     background: #5d5d5d;  
16     background: -moz-linear-gradient(top, #868686  
4%, #646464 50%, #505050 69%, #404040 97%), -moz-  
linear-gradient(top, #bdbdbd 6%, #888888 63%,  
#707070 78%, #5d5d5d 98%);  
17     background: -o-linear-gradient(top, #868686  
4%, #646464 50%, #505050 69%, #404040 97%), -o-  
linear-gradient(top, #bdbdbd 6%, #888888 63%,
```

# Modernizr: Detecting CSS3 Features & Fallbacks

- ◆ On document load Modernizr detects which features are supported
  - ◆ Adds classes "feature" / "no-feature" for the features to the HTML element
- ◆ Example features:
  - ◆ **canvas, rgba, sessionstorage, etc.**
    - ◆ If the features are supported
    - ◆ **no-canvas, no-rgba, no-sessionstorage, etc.**
      - ◆ If the features are not supported

- ◆ If CSS gradients are not supported, use a fallback gradient with PNG repeated by X:

```
<script src="modernizr.js"></script>
<style>
  .box {
    width: 150px;
    height: 150px;
    border: 1px solid black;
  }
  .cssgradients .box {
    // css gradients code
  }
  .no-cssgradients .box {
    background: url(gradient.png) 0 0 repeat-x;
  }
</style>
```

# Modernizr: Detecting CSS3 Features & Fallbacks

Live Demo

```
9  label {
10    display: block;
11    position: relative;
12  }
13
14 .pointerevents label:before {
15   background: #5d5d5d;
16   background: -moz-linear-gradient(top, #868686
17   4%, #646464 50%, #505050 69%, #404040 97%), -moz-
linear-gradient(top, #bdbdbd 6%, #888888 63%,
#707070 78%, #5d5d5d 98%);
18   background: -o-linear-gradient(top, #868686
19   4%, #646464 50%, #505050 69%, #404040 97%), -o-
linear-gradient(top, #bdbdbd 6%, #888888 63%,
```

# Modernizr Load

## Yep / Nope Loading of JS Polyfills for Missing HTML5 Features

### Modernizr.load() tutorial

`Modernizr.load` is a resource loader (CSS and JavaScript) that was made to specifically work side-by-side with Modernizr. It's optional in your build, but if you are loading polyfills, There's a good chance it can save you some bandwidth and boost performance a bit.

`Modernizr.load` syntax is generally very easy to understand, so we'll start with a basic example:

```
Modernizr.load({
  test: Modernizr.geolocation,
  yep : 'geo.js',
  nope: 'geo-polyfill.js'
});
```

- ◆ Modernizr can test for features and load resources depending on their support
  - ◆ Used to load polyfills for unsupported features

```
<script src="modernizr.js"></script>

<script>
  Modernizr.load({
    test: Modernizr.audio,
    nope: 'http://api.html5media.info/1.1.5/
html5media.min.js'
  });
</script>
```

## Modernizr.load() tutorial

`Modernizr.load` is a resource loader (CSS and JavaScript) that was made to specifically work side-by-side with Modernizr. It's optional in your build, but if you are loading polyfills, There's a good chance it can save you some bandwidth and boost performance a bit.

`Modernizr.load` syntax is generally very easy to understand, so we'll start with a basic example:

```
Modernizr.load({
  test: Modernizr.geolocation,
  yep : 'geo.js',
  nope: 'geo-polyfill.js'
});
```

# Modernizr Load

Live Demo

# CSS Boiler Plates

Resetting made simple

- ◆ Lots of browsers over there
  - ◆ Each browser defines its own default styles
  - ◆ Need to reset everything we use

```
*{margin:0; padding:0}  
html, body, div, header...{margin:0; padding:0}
```

- ◆ Boiler plates deliver a ready-to-use CSS style resets
  - ◆ <http://html5boilerplate.com/>
  - ◆ <http://www.getskeleton.com>

# Questions?

1. \*Make the web page task1.html with styles task1.css (or task1.less) to work on all desktop browsers, including IE7

- You are not allowed to change the original CSS(LESS) and HTML. You can only expand it
- Use shivs, shims or polyfills

2. \*Make a web page that detects the current geolocation

- The web page should work fine in all desktop browsers, including IE7

3. \*Create a web page that contains the following input types
  - range, number, email, date, color, datetime, month, search, tel, time, url, week
  - And it must work on all desktop browsers (IE7+)
  - Research about Webshims Lib