

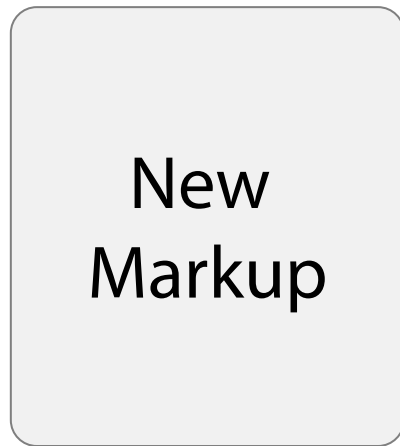
What is HTML5?

Related Technologies

W3C HTML5 Specification

Initial WHATWG HTML5 Specification

What is HTML5?



+



=



New Elements

article
aside
audio
bdi
canvas
datalist
details
embed
figcaption
figure

footer
header
main
math
mark
meter
nav
output
progress
section

source
summary
svg
time
track
video
wbr

article



```
<article></article>
```

aside



```
<aside></aside>
```

audio
source
track



```
<video>  
  <source src="vid.mp4">  
  <source src="vid.ogv">  
  <source src="vid.webm">  
  <track  
    kind="subtitles"  
    label="English Subtitles"  
    src="en.vtt"  
    srclang="en">  
</video>
```

canvas



```
<canvas></canvas>
```



The diagram consists of a large light gray rounded rectangle in the background. In the foreground, there is a smaller light gray rounded rectangle on the left containing HTML code. To its right is a white rounded rectangle with the word 'datalist' in black text. On the far left, there is a dark gray code icon consisting of a less-than sign, a forward slash, and a greater-than sign.

datalist



```
<input
  type="text"
  list="colors">
<datalist id="colors">
  <option src="Blue">
  <option src="White">
</datalist>
```




details
summary



```
<details>  
  <summary>Events</summary>  
  <p>Come join us  
    this summer...</p>  
</details>
```

embed



```
<embed  
  src="intro.swf">
```

figure
figcaption



```
<figure>  
    
</figure>  
<figcaption>  
  Screenshot of Menu  
</figcaption>
```

header



```
<header></header>  
<footer></footer>
```

main



```
<main></main>
```



math



```
<math>  
  <!-- MathML elements -->  
</math>
```

mark



```
<mark>  
  this text is often highlighted  
</mark>
```

meter



```
<meter  
  min="0"  
  max="100"  
  value="70">70 out of 100</meter>
```



nav



```
<nav></nav>
```

output



```
<output></output>
```

progress



```
<progress  
  value="50"  
  max="0"></progress>
```



section



```
<section></section>
```

time



```
<time  
  datetime="2013-07-08T13:25:23-8:00"  
  pubdate>  
  July 7th, 2013  
</time>
```

wbr



reallybig<wbr>word

New Elements

article
aside
audio
bdi
canvas
datalist
details
embed
figcaption
figure

footer
header
main
math
mark
meter
nav
output
progress
section

source
summary
svg
time
track
video
wbr

New JavaScript APIs

New JavaScript APIs

Canvas

IndexedDB

Selection

Contacts

Media Capture

Server-Sent Events

File API

Microdata

Web Notifications

Forms

Messaging

Web Sockets

Geolocation

Offline Web Applications

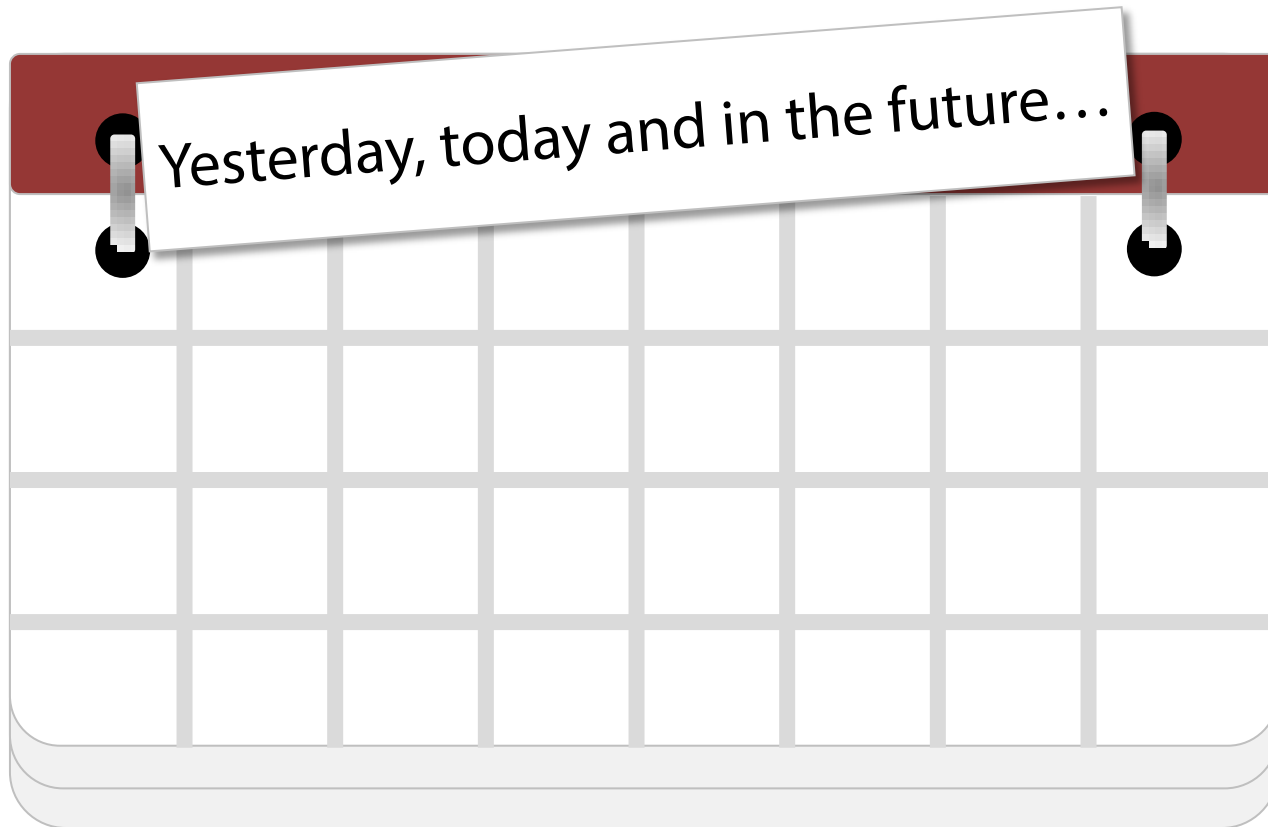
Web Storage

Web Workers

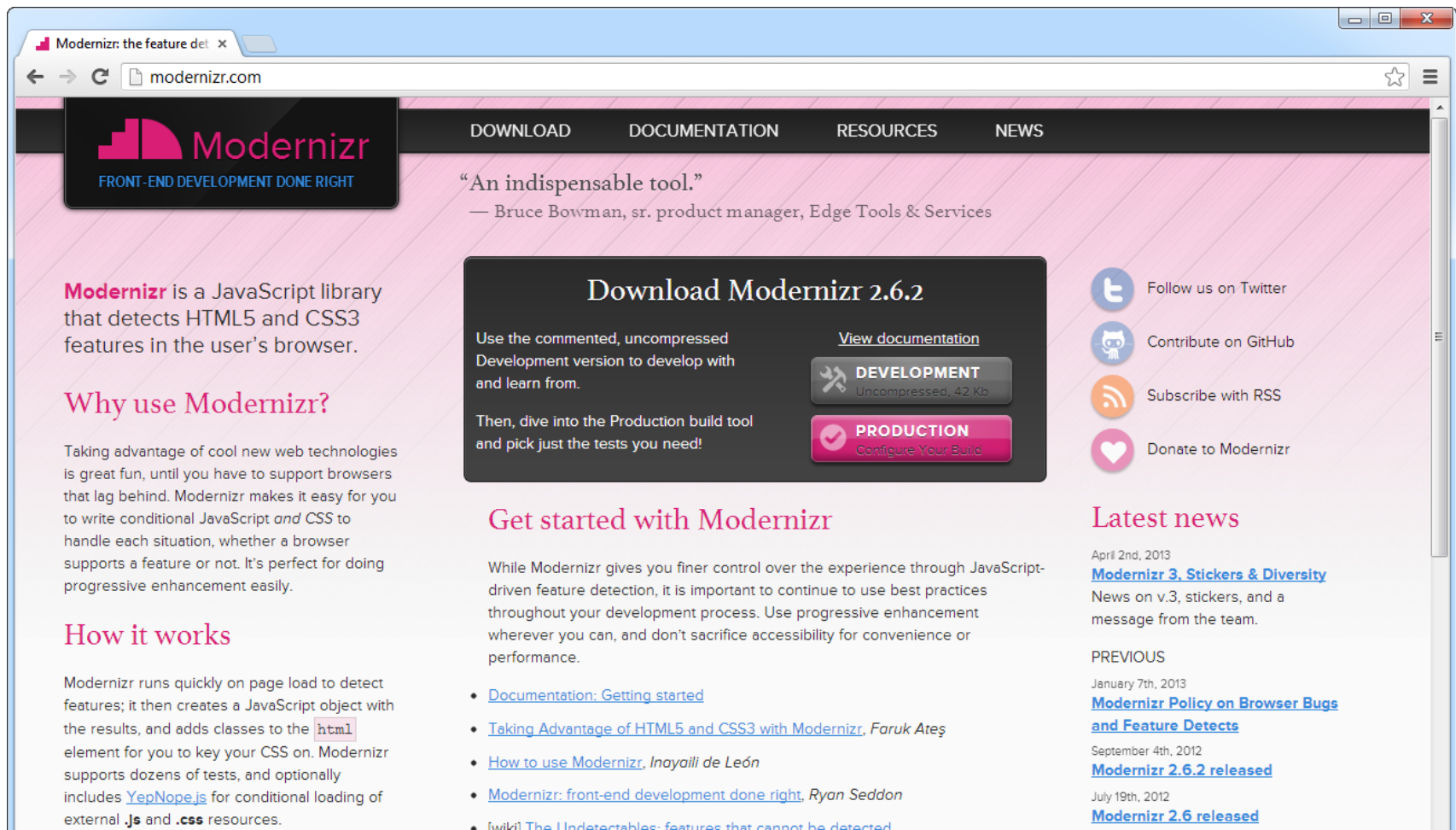
XMLHttpRequest Level 2

When is HTML5 'done'?

When is HTML5 'done'?



Detecting HTML5



The screenshot shows the Modernizr website in a browser window. The browser's address bar displays 'modernizr.com'. The website has a dark header with the Modernizr logo and navigation links: DOWNLOAD, DOCUMENTATION, RESOURCES, and NEWS. The main content area has a pink background with diagonal lines. On the left, there's a section titled 'Modernizr' with the tagline 'FRONT-END DEVELOPMENT DONE RIGHT'. Below this, it explains that Modernizr is a JavaScript library for detecting HTML5 and CSS3 features. A 'Why use Modernizr?' section follows, highlighting its use for progressive enhancement. The 'How it works' section describes the process of detecting features and adding classes to the HTML element. In the center, a dark box promotes 'Download Modernizr 2.6.2', offering both a development version (commented, uncompressed) and a production build (configured). To the right, there are social media links for Twitter, GitHub, RSS, and a donation button. At the bottom right, a 'Latest news' section lists recent updates, including the release of Modernizr 3 and Modernizr 2.6.2.

Modernizr: the feature detector

modernizr.com

Modernizr
FRONT-END DEVELOPMENT DONE RIGHT

DOWNLOAD DOCUMENTATION RESOURCES NEWS

“An indispensable tool.”
— Bruce Bowman, sr. product manager, Edge Tools & Services

Modernizr is a JavaScript library that detects HTML5 and CSS3 features in the user's browser.

Why use Modernizr?

Taking advantage of cool new web technologies is great fun, until you have to support browsers that lag behind. Modernizr makes it easy for you to write conditional JavaScript and CSS to handle each situation, whether a browser supports a feature or not. It's perfect for doing progressive enhancement easily.

How it works

Modernizr runs quickly on page load to detect features; it then creates a JavaScript object with the results, and adds classes to the `html` element for you to key your CSS on. Modernizr supports dozens of tests, and optionally includes [YepNope.js](#) for conditional loading of external `.js` and `.css` resources.

Download Modernizr 2.6.2

Use the commented, uncompressed Development version to develop with and learn from. [View documentation](#)

DEVELOPMENT
Uncompressed, 42 Kb

Then, dive into the Production build tool and pick just the tests you need!

PRODUCTION
Configure Your Build

Get started with Modernizr

While Modernizr gives you finer control over the experience through JavaScript-driven feature detection, it is important to continue to use best practices throughout your development process. Use progressive enhancement wherever you can, and don't sacrifice accessibility for convenience or performance.

- [Documentation: Getting started](#)
- [Taking Advantage of HTML5 and CSS3 with Modernizr](#), Faruk Ateş
- [How to use Modernizr](#), Inayaili de León
- [Modernizr: front-end development done right](#), Ryan Seddon
- [\[wiki\] The Undetectables: features that cannot be detected](#)

Latest news

April 2nd, 2013
[Modernizr 3, Stickers & Diversity](#)
News on v.3, stickers, and a message from the team.

PREVIOUS

January 7th, 2013
[Modernizr Policy on Browser Bugs and Feature Detects](#)

September 4th, 2012
[Modernizr 2.6.2 released](#)

July 19th, 2012
[Modernizr 2.6 released](#)

Follow us on Twitter

Contribute on GitHub

Subscribe with RSS

Donate to Modernizr