elementos HTAL5

Detecção

Browser vs Feature

HTML5 5 PLEASE

Use the new and shiny responsibly.

Look up HTML5, CSS3, etc features, know if they are ready for use, and if so find out how you should use them – with polyfills, fallbacks or as they are. tell me more ▶

box-shadow

link to this search >

Explore

supported by IE10+ • IE9+ • IE8+ • IE7+ • no IE

not supported by mobile devices • older mobile devices

requiring prefixes • polyfill • fallback

that you should use • use with caution • avoid

that are css • html • api • js

all features

0

css

box-shadow

caution

box-shadow is safe to use most of the time, except when you are applying it on a large area as that may cause significant performance impact when interacting with the page (on scroll and hovers, but also transitions and animations).

Polyfills are unnecessary for box-shadow. They are too costly and slow down page load. Users without a modern browser will not see a shadow, but aren't checking your site in multiple browsers anyhow.

Box shadows can also be applied safely without prefixes.

html5please.com



TOP



```
yepnope({
  test : Modernizr.geolocation,
  yep : 'normal.js',
  nope : ['polyfill.js', 'wrapper.js']
});
```

yepnope is an asynchronous conditional resource loader that's super-fast, and allows you to load only the scripts that your users need.

What's new in yepnope 1.5?

- · Scripts with the same url won't execute twice, but their callbacks fire in the correct order.
- The complete function behaves much more like the callback function with respect to 'recursive yepnope'.
- CSS load callbacks were taken out by default and put in an official plugin. Too few people used it. Old code will still work, callbacks will
 just fire immediately on CSS (unless css load plugin is included).
- We exposed yepnope, inject Is and yepnope, inject Css in order to give you direct access to the injection functions.

```
yepnope.injectJs( scriptSource [, callback ] [, elemAttributes ] [, timeout ]);

yepnope.injectJs( scriptSource [, callback ] [, elemAttributes ] [, timeout ]);

yepnope.injectJs( scriptSource [, callback ] [, elemAttributes ] [, timeout ]);

yepnope.injectJs( scriptSource [, callback ] [, elemAttributes ] [, timeout ]);

yepnope.injectJs( scriptSource [, callback ] [, elemAttributes ] [, timeout ]);
```

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.is for conditional loading of external .js and .css resources.

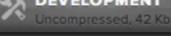
Check out the full list of features that Modernizr detects, or learn more about conditional resource loading with Modernizr.

Download Modernizr 2.6.2

Use the commented, uncompressed Development version to develop with and learn from.

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

View documentation DEVELOPMENT





Follow us on Twitter



Contribute on GitHub



Subscribe with RSS



Donate to Modernizr

Get started with Modernizr

While Modernizr gives you finer control over the experience through JavaScriptdriven 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 Ates
- How to use Modernizr, Inayaili de León
- Modernizr: front-end development done right, Ryan Seddon
- [wiki] The Undetectables: features that cannot be detected
- [wiki] Cross-browser Polyfills

Also check out our Resources section.

Tip: check our Modernizr test suite to quickly test your current browser's features

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 Bug and Feature Detects

September 4th, 2012

Modernizr 2.6.2 released

July 19th, 2012

Modernizr 2.6 released







Estrutura

- header, footer, main
- article, section, aside
- nav, hgroup, ...

Formulário

- tipos
- atributos
- validadores

Carregamento

- meter
- progress
- details
- summary

Áudio

- autoplay
- controls
- loop
- muted
- preload

Áudio

- MP3 (Chrome, Safari, IE9)
- WAV (Chrome, Firefox, Safari, Opera)
- OGG (Chrome, Firefox, Opera)

Vídeo

- autoplay
- controls
- loop
- muted
- poster
- preload

Vídeo

- MP4 (Chrome, Safari, IE9)
- WebM (Chrome, Firefox, Opera)
- OGG (Chrome, Firefox, Opera)

Chefão

Montar um site usando apenas elementos HTML5