

Modern Web Development

mozilla

A new world...

- HTML5 != Desktop
- Exciting new opportunities
- New browsers with one goal
- Taking this show on the road

Demo time: HTML5 magic built-in

Memo time:

- ➊ **HTML5 is backwards compatible and robust by design**
- ➋ **That doesn't mean we should rely on that - it just means we can move forward safely**
- ➌ **HTML5 has a lot of in-built client side functionality we do not need to write in JavaScript ourselves.**

Developing for the unknown





Progressive enhancement

Progressive Enhancement

- Start with something that works
- Then hijack it to become better
- Test for support, then add on
- No broken promises

Graceful degradation



mozilla

Graceful degradation

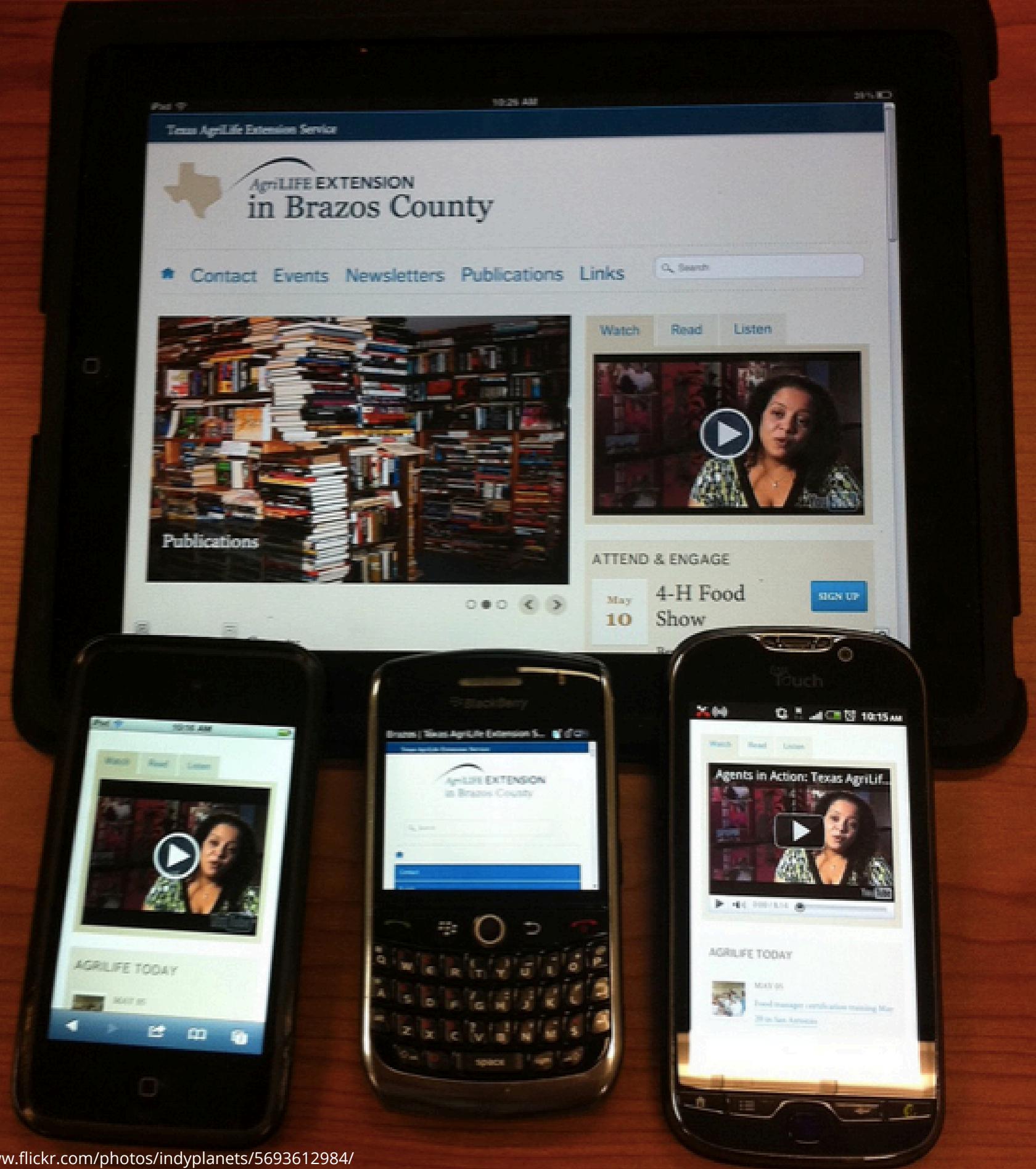
- Start with a complex matter
- Develop a fallback for other environments
- Maintain both
- Broken promises

Demo time: **Progressive Enhancement** **vs. Graceful Degradation**

Memo time:

- ➊ Be flexible and nothing will cause you grief in the future
- ➋ Don't make end users suffer for your mistakes
- ➌ Understand the basics
- ➍ Don't bother with PE when you are in a fixed environment

Responsive Design



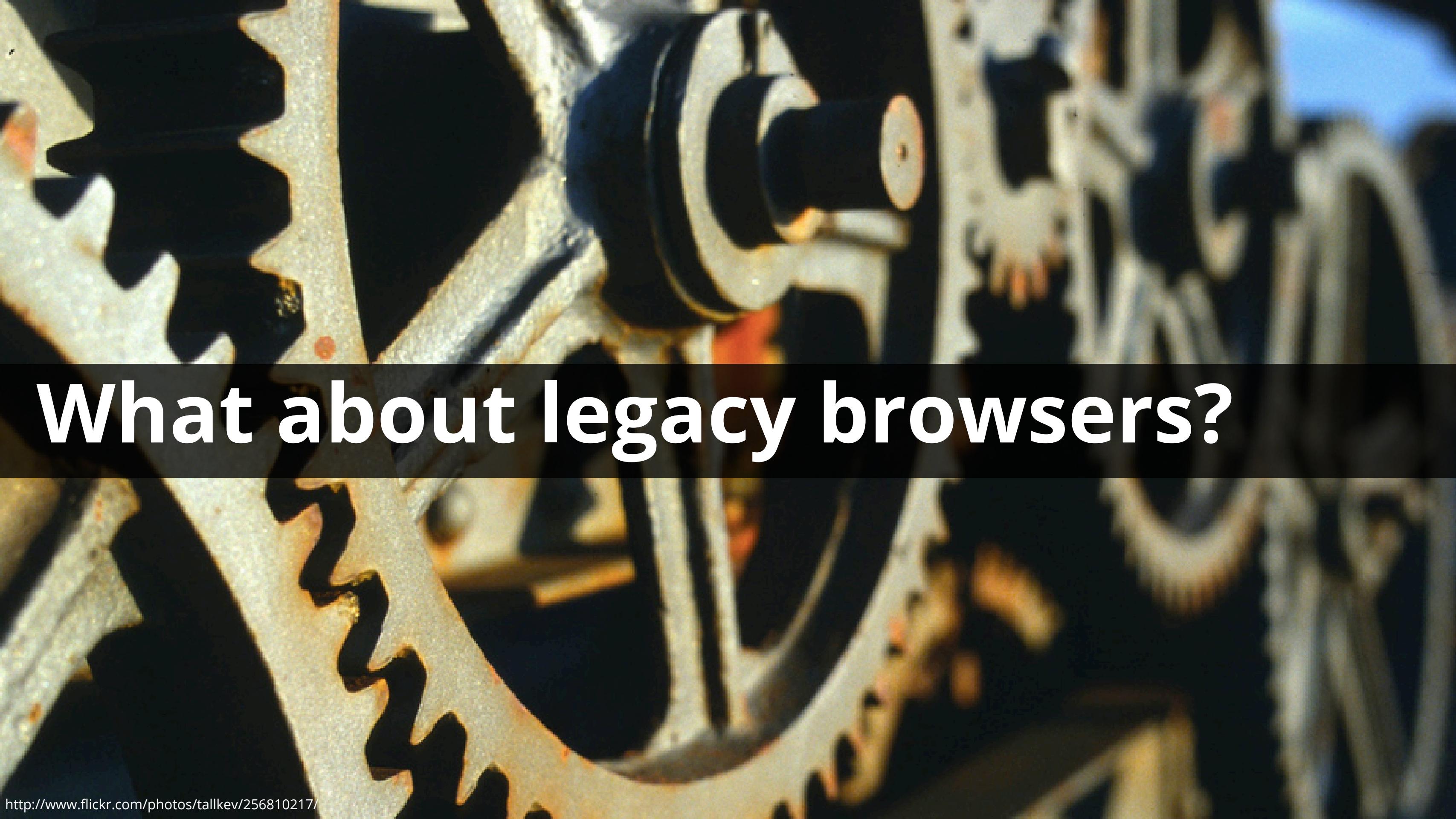
Mediaqueries

- Apply CSS for certain scenarios
- Also available in JavaScript
- One design, many views

Demo time: Responsive design demos

Memo time:

- Mediaqueries allow for granular design changes
- No need for JS sniffing
- Possible performance issues :(
- matchMedia() can help
- Queries for connection speed needed



What about legacy browsers?



Polyfills

“Modernizr is an essential part of my toolkit of files.”

— Andy Clarke, *Stuff & Nonsense*

Modernizr is an open-source JavaScript library that helps you build the next generation of HTML5 and CSS3-powered websites.

Why use Modernizr?

Taking advantage of the new capabilities of HTML5 and CSS3 can mean sacrificing control over the experience in older browsers.

Modernizr 2 is your starting point for making the best websites and applications that work exactly right no matter what browser or device your visitors use.

Thanks to the new Media Query tests and built-in [YepNope.js](#) micro-library as `Modernizr.load()`, you can now combine feature detection with media queries and conditional resource loading. That gives you the power and flexibility to optimize for every circumstance.

Check out the [full list of features](#) that Modernizr detects, or learn more about [conditional resource loading](#).

<http://modernizr.com/>

Download Modernizr 2.5.3

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](#)
- [wiki] [Cross-browser Polyfills](#)

Also check out our [Resources section](#).

Tip: check out [Modernizr test suite](#) to quickly test your current browser's features.

HTML5 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 ▶](#)

type to filter

Explore
features

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](#)

css

+ [animations](#)

caution with fallback

html

+ [<audio>](#)

use with polyfill

css

+ [background-image options](#)

use with fallback

css

+ [border-image](#)

use with fallback

css

+ [border-radius](#)

use

<http://html5please.com/>

mozilla

HTML5 ★ BOILERPLATE

A rock-solid default for HTML5 awesome.

- I'm new here: plz explain why it's good, first.

[DOWNLOAD BOILERPLATE 3.0.2 UPDATED FEB 19TH](#)

DOWNLOAD
BOILERPLATE

OR

DOWNLOAD
BOILERPLATE "STRIPPED"

NO COMMENTS, JUST THE BIZNIS.

OR

CUSTOMIZE
BOILERPLATE

100% HIPSTER.

KEEP THE HINTS AND LINKS

[READ THE DOCS](#) ★ [CONTRIBUTE ON GITHUB](#) ★ [FOLLOW ON TWITTER](#)

HTML5 BOILERPLATE IS 3.0!

MONDAY, FEB 6TH, 2012

The key feature of this update is making Boilerplate smaller. Most of the changes have been working towards that. We have significant work done to our build script, so much so that we thought it deserved its own repository. Meanwhile, here is what's new.

CHANGELOG

3.0 gets some major updates. Here are all the commits since last release.

Should you upgrade existing sites? Short answer: nah, you're good.

<http://html5boilerplate.com/>

SIGNIFICANT CHANGES

mozilla



320 and Up

The ‘tiny screen first’ responsive boilerplate

[320 and Up on Github](#)

[Download](#)

This is the new ‘320 and Up’

A lot's changed since I wrote [the original ‘320 and Up’](#), my ‘tiny screen first’ responsive web design boilerplate. Back then we were just getting started with responsive web design and many sites, including mine, and frameworks and boilerplates like HTML5 Boilerplate, structured their CSS3 Media Queries from the desktop down, rather than for small screens up.

(Oh how we laughed when we realised our mistake.)

To put things right, I wrote ‘320 and Up’. It worked as an extension to [HTML5 Boilerplate](#) or a set of standalone files. ‘320 and Up’ has been used by designers and developers all over the web and I’ve used versions of it on every website I’ve worked on since I wrote it.

What's in the new '320 and Up'?

Five CSS3 Media Query increments: 480, 600, 768, 992 and 1382px

<http://stuffandnonsense.co.uk/projects/320andup/>

mozilla

Rule of thumb:

Giving legacy browsers a working solution is better than a full one that means a lot of work and doesn't perform.

Be future friendly!



Adaptive Web Design

Crafting Rich Experiences with Progressive Enhancement

by **Aaron Gustafson**

Foreword by **Jeffrey Zeldman**

“ Adaptive Web Design *not only provides the clearest, most beautiful explanation of progressive enhancement I’ve ever read, it’s also packed full of practical know-how pumped directly into your neocortex through Aaron’s warm and friendly writing style. If you aren’t already using progressive enhancement to build websites, you soon will be.*

— Jeremy Keith, Author, *HTML5 for Web Designers*

Don't believe Jeremy?
Read a sample for yourself

DOWNLOAD CHAPTER 1

FREE

FUTURE ★ FRIENDLY



**In today's incredibly exciting yet overwhelming world
of connected digital devices, these are the truths we
hold to be self-evident:**

Disruption will only accelerate. The quantity and diversity of connected devices—many of which we haven't imagined yet—will explode, as will the quantity and diversity of the people around the world who use them. **Our existing standards, workflows, and infrastructure won't hold up.** Today's onslaught of devices is already pushing them to the breaking point. They can't withstand what's ahead. **Proprietary solutions**

<http://futurefriend.ly>

mozilla