



Resources

Cory House
bitnative.com
Twitter: @housecor



pluralsight 
hardcore developer training

[HOME](#)[ARTICLES](#)[PRESENTATIONS](#)[RESOURCES](#)[SANDBOX](#)

WebComponents.org

a place to discuss and evolve web component best-practices

WHAT?

WebComponents.org is where pioneers and community-members of the Web Components ecosystem (like *Polymer*, *X-tags*, and other interested parties) document web components best practices so that others can follow the same path instead of needlessly striking out on their own.

SPECS



WEB COMPONENTS

This document is a non-normative reference, which provides an overview of Web Components. It summarizes the normative information in the respective specifications in easy-to-digest prose with illustrations.



CUSTOM ELEMENTS

This specification describes the method for enabling the author to define and use new types of DOM elements in a document.



HTML IMPORTS

HTML Imports are a way to include and reuse HTML documents in other HTML documents.

ARTICLES



WEB COMPONENTS BEST PRACTICES

[Web Components](#) (WC) are a new set of web platform features that enable developers to build applications in a declarative, composable way. The following is an initial list of best practices we advocate component authors consider to ensure their elements are good citizens in the Web Component ecosystem.

[Read More >](#)

[see all articles](#)

BROWSER SUPPORT

CHROME OPERA FIREFOX SAFARI IE

WebComponents.org



Shadow DOM 101

Table of Contents

- + Introduction
- + Hello, Shadow World
- + Separating Content from Presentation
- + Extra Credit: Advanced Projection
- + You Pass Shadow DOM 101

Localizations

- + 日本語
- + 한국어
- + 中文 (简体)
- + Contribute another



By [Dominic Cooney](#)

Published: January 4th, 2013

Updated: December 18th, 2013

Comments: [43](#)

Introduction

Web Components is a set of cutting edge standards that:

1. Make it possible to build widgets
2. ...which can be reused reliably
3. ...and which won't break pages if the next version of the component changes internal implementation details.

Does this mean you have to decide when to use HTML/JavaScript, and when to use Web Components? No! HTML and JavaScript can make interactive visual stuff. Widgets are interactive visual stuff. It makes sense to leverage your HTML and JavaScript skills when developing a widget. The Web Components standards are designed to help you do that.




Custom Elements

a web components gallery for modern web apps

 Tweet 1,419

 Like 1.1k

 +1 381

 Star 385

What are Web Components?

Web Components are a collection of standards which are working their way through the W3C. They enable truly encapsulated and reusable components for the web. And if you think HTML5 changed the web, wait to see what Web Components will do.

For lots more information about it, including articles and presentations, check out webcomponents.org.

— [Zeno Rocha](#), project lead.

Submit your own

Got a great idea for a custom element? Awesome! There are boilerplates for [Polymer](#), [X-Tag](#), and [VanillaJS](#) that you can fork and get up and running with a simple component.

When you're ready to go, submit it to the form below and it'll appear on this site for others to play and use!

Submit

Latest elements

[google-doc](#)

A web component to import Google Doc content into a

Featured elements

[ajax-form](#)

HTML forms on performance-enhancing drugs

Most popular elements

[x-gif](#)

A custom element for flexible GIF playback



Learn when new web components are available: [Subscribe](#)



<input type="checkbox"/>	Title	Date	Type
	<input type="text"/>	<input type="text"/>	<input type="button" value="Please select"/>
<input type="checkbox"/>	gym	2014-04-14	private
<input type="checkbox"/>	breakfast	2014-04-14	private
<div><< Prev 1 2 3 Next >></div>			



Action

Declarative way of mapping users actions to custom element methods

aha-programable [DEMO](#)

A custom element to calculate a piece of js code and use the result as the content.

aha-table [DEMO](#)

A Polymer element for a searchable, sort-able, paginate-able, inline-editable,

aha-tree [DEMO](#)

A PolymerElement to convert a JSON to a tree.



```
function $initHighlight(block, cls) {
  try {
    if (cls.search(/\bno\-highlight/)
        return process(block, true, {
          'class':'';
        })
  } catch (e) {
    /* handle exception */
  }
  for (var i = 0 / 2; i < classes.length; i++) {
    if (checkCondition(classes[i]))
      return /\d+[\s/]/g;
  }
}
```

akyr-al-code [DEMO](#)



akyr-al-element-query [DEMO](#)



akyr-al-layout [DEMO](#)

The Extensible Web Manifesto

#extendthewebforward

We—the undersigned—want to change how web standards committees create and prioritize new features. We believe that this is critical to the long-term health of the web.

We aim to tighten the feedback loop between the editors of web standards and web developers.

Today, most new features require months or years of standardization, followed by careful implementation by browser vendors, only then followed by developer feedback and iteration. We prefer to enable feature development and iteration in JavaScript, followed by implementation in browsers and standardization.

To enable libraries to do more, browser vendors should provide new low-level capabilities that expose the possibilities of the underlying platform as closely as possible.

They should also seed the discussion of high-level APIs through JavaScript implementations of new features (such as Mozilla's X-Tags and Google's Polymer).



Go Forth and Componentize!

