Web Components: Background, Opportunities and Challenges

Presented by: Karl Groves and Alice Boxhall

Look at Chromium Dashboard to see what the ‘path of desire’ is.

The essential problem we’re trying to address is semantics

Web components:

* Reusable
* Encapsulated
* Composeable
* Extensible

<template> - a container with no semantics and is not rendered

Custom elements – a whole spec that boils down to three things:

* Name (“unicorn bike”)
* Prototype (object.prototype)
* Lifecycle callbacks – a set of functions that are guaranteed to be called:
  + createdCallBack()
  + attachedCallBack()
  + attributeChangedCallBack(attrName, oldVal, newVal)
  + deletedCallBack()

HTML imports

* Content is not rendered but
* JavaScript context is sharead and
* Document references the importing document

Shadow DOM

* Gives the encapsulation (content in the shadow don’t leak out into the rest of the document)
* ID refs are scoped within the shadow

Web Components are still a under discussion

* What level of encapsulation show Shadow DOM provide?
* How do we extend native elements?

Browser support for existing specs is … inconsistent (Safari and IE don’t support any of the aspects, and Firefox only supports one aspects but have stated they don’t plan on support imports which are very important for web components)

Web Components are the new jQuery Plugins (but 90% aren’t accessible and aren’t written well)

“I’m writing a new talk titled ‘demystifying web components accessibility.’ It has one slide. It says ‘same as regular ole web accessibility.’ Seriously. Stop over thinking it.

Extensible web manifesto