how to train your dragon
web standard
i built this
how to train your dragon

web standard
first, you need a dragon problem
...then wait until it’s everyone’s problem
(this is very easy if you have dragons)
complain about it around browser people
complain about it around browser people
(browser people loooove problems)
YAY! now it’s the browser’s problem!
standards are a compromise between all browser vendors
we need web standards so that browsers don’t copy each other’s bugs
also, standardizing things is really not new
railroads use standard tracks
hot dogs
don’t
hot dogs

don’t
don’t be like hot dogs.
some standards aren’t successful
2011: <input type="date">
2011: `<input type="date">`

2017: sigh
“I never want to implement another date picker again”
— rob wormald, 2017
web standards hindsight is 20/20
you need to standardize the right thing not just any thing
2013: extensible web manifesto
“The standards process should focus on adding new low-level capabilities"
4 things browser vendors want in a standards proposal
the browser already does the thing
the browser *already* does the thing

the thing explains *existing* browser magic
• the browser already does the thing
• the thing explains existing browser magic
• the thing can be extended
the browser already does the thing
the thing explains existing browser magic
the thing can be extended
people already used and liked the thing
<input type="date">

- the browser already does the thing
- the thing explains existing browser magic
- the thing can be extended
- people already used and liked the thing
<input type="date">

- the browser already does the thing
- the thing explains existing browser magic
- the thing can be extended
- people already used and liked the thing
<input type="date">

- the browser already does the thing
- the thing explains existing browser magic
- the thing can be extended
- people already used and liked the thing
the browser already does the thing
the thing explains existing browser magic
the thing can be extended
people already used and liked the thing
<input type="date">

the browser already does the thing
the thing explains existing browser magic
the thing can be extended
people already used and liked the thing
you need to standardize the *right* thing not just any thing
you need to standardize
the a component model not just a component
you might say a web...component...model
congratulations.
you have become
alex russell
2012ish: learn from what all component models have in common
2012ish: learn from what all component models have in common
standardize ideas
not implementations
a useful lifecycle
a useful lifecycle
extensibility
- a useful lifecycle
- extensibility
- (maybe) implementation encapsulation
- a useful lifecycle
- extensibility
- (maybe) implementation encapsulation
- a space suit to breathe in space
☐ a useful lifecycle
☐ extensibility
☐ (maybe) implementation encapsulation
☐ a space suit to breathe in space (DOM)
this was exactly the web components proposal
<shiny-button></shiny-button>
<shiny-button></shiny-button>

- (maybe) implementation encapsulation
- a space suit to breathe in space (DOM)
<shiny-button></shiny-button>

class ShinyButton extends HTMLElement {
    constructor() {
    }
    connectedCallback() {
    }
}

customElements.define('shiny-button', ShinyButton)
a useful lifecycle
extensibility

class ShinyButton extends HTMLElement {
    constructor() {
    }
    connectedCallback() {
    }
}
customElements.define('shiny-button', ShinyButton)
the browser already does the thing
the thing explains existing browser magic
the thing can be extended
people already used and liked the thing
the browser already does the thing

the thing explains existing browser magic

the thing can be extended

people already used and liked the thing
the browser already does the thing
the thing explains existing browser magic
the thing can be extended
people already used and liked the thing
the browser already does the thing
the thing explains existing browser magic
the thing can be extended
people already used and liked the thing
because they’re low level
you can build frameworks on top of them
because they’re *low level* browsers can optimize them
whew.
2017:
why is progress so slow?
realtalk: browsers are made by humans and they need to sleep
the browser already does the thing
the thing explains existing browser magic
the thing can be extended
people already used and liked the thing
people don’t trust a standard unless it’s shipped on browsers
browsers don’t ship a standard unless it’s been used by people
we should be waaaaaaaaaaaaay more stoked about polyfills
polyfills teleport your browser into the future
unlike diamonds, polyfills aren’t forever
unlike diamonds, polyfills aren’t forever
the browser already does the thing
the thing explains existing browser magic
the thing can be extended
people already used and liked the thing
the browser *already* does the thing
the thing explains *existing* browser magic
the thing can be *extended*
people already *used and liked* the thing
after 7 years,
your standard will actually be 6 *months* old!
1st email
1st polyfill
polymer 0.1
v0 spec ships
questions we get asked a lot
“why isn’t it more successful?”
“why isn’t it more successful? usually means
“why aren’t we using it?”
you might not need a standard component model
(and that’s ok!)
big companies need a standard component model
(to share the same widgets)
component libraries need a standard component model (to share the widgets with everyone)
frameworks need
a standard component model
(to share the widgets with everyone)
“is it viable without polymer?”
“is it viable without **polymer**?

usually means

we’ve done bad dev rel-ing
new API documented

best practices explained

libraries do best practices

framework interop utopia
"do web components toss view-source overboard?"
if the browser understands the **semantics** of your element it can add tools for it
what’s next?
democratize web standards
shaping web standards shapes the web platform