

20 years of CSS: maturity or senility?

20 години CSS: мъдрост или оглупяване?

Daniel Glazman

@glazou

CSS Working Group, Co-chair

Bulgaria Web Summit 2015

Sofia, Bulgaria



They invented CSS
(and they saw it was good...)



Then things became serious...

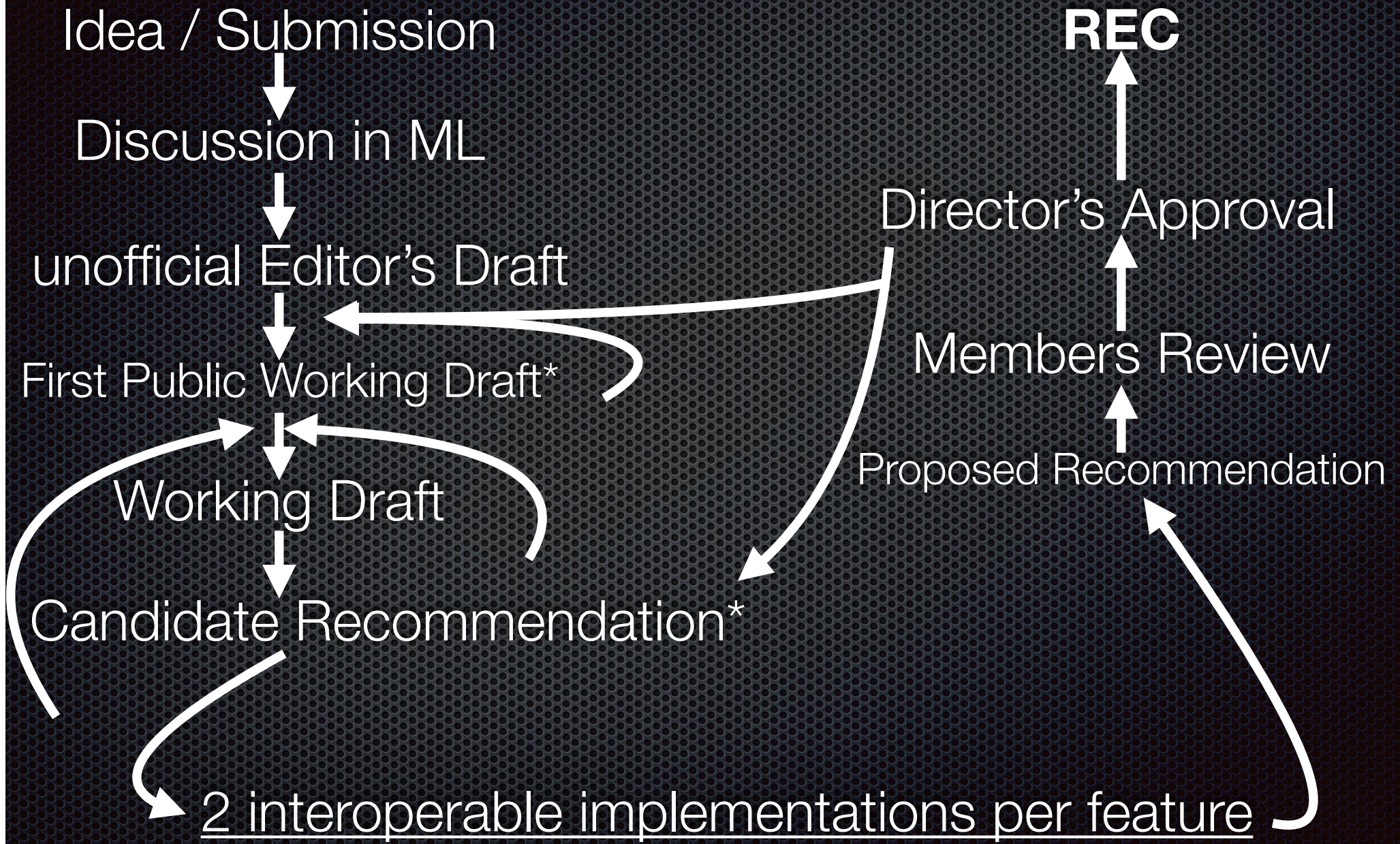
(photo of CSS WG daily life)

CSS WG	2008	2013	%
Members	32	98	+206%
Active Members	<10	35	+250%
Specifications	30	65	+116%



Growth in circles

How long can we live with it? How long can **you** live with it?



Variables

- CSS Custom Properties for Cascading Variables aka User-Defined Inherited Properties (you dip?)
- 16 years for a LCWD

```
body {  
  --corpcolor: orange;  
}  
  
#foo {  
  color: var(--corpcolor);  
}
```


Why not just #define?

<troll>

- ✦ Faaaaaaaaaaaar too simple...
- ✦ Do you really think we're Web designers?
- ✦ We love fighting for 15 years and then blame W3C Process for being slow...
- ✦ Don't you love our innovative -- notation?

</troll>

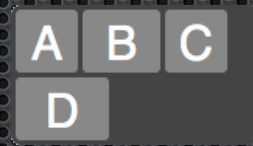


The Mother of All Hacks

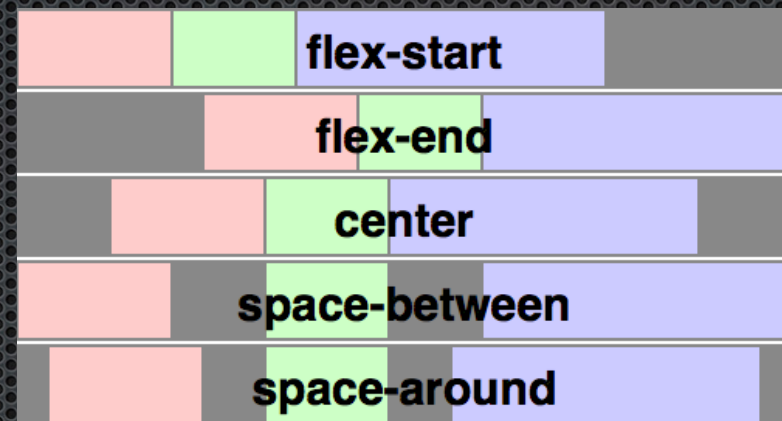
(if you never asked the question, then you don't work with CSS at all)

Flexible Box Model

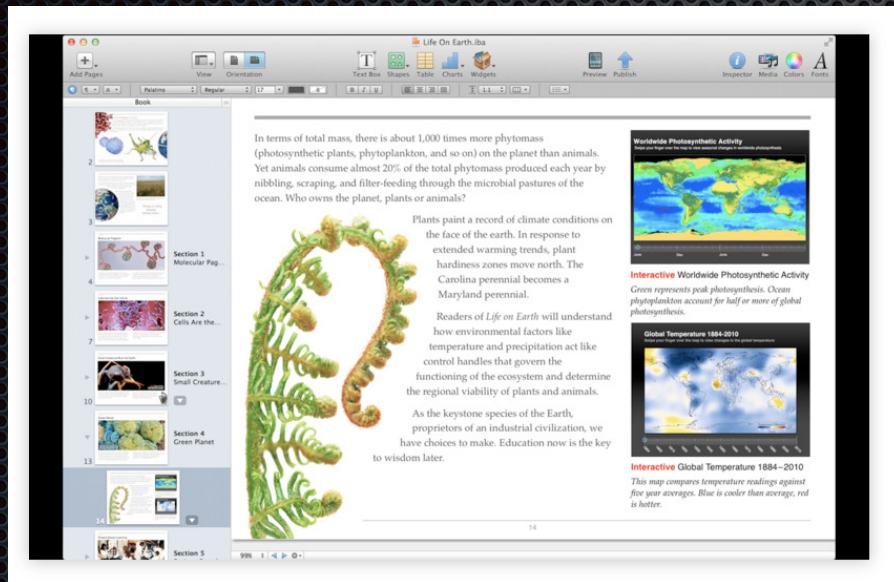
- ✦ Align; size; distribute space.
- ✦ But also flows and wrapping...
- ✦ More complex than in XUL or XAML because there is no element to indicate space...
- ✦ LCWD



An endless opportunity to bikeshed...



Magazine layout



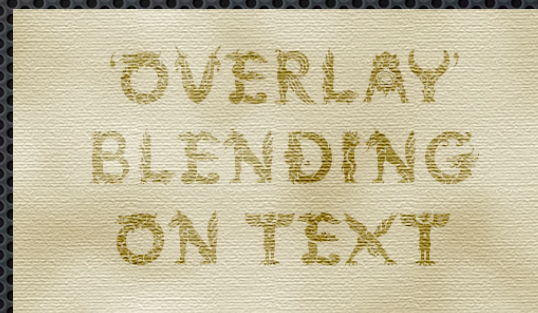
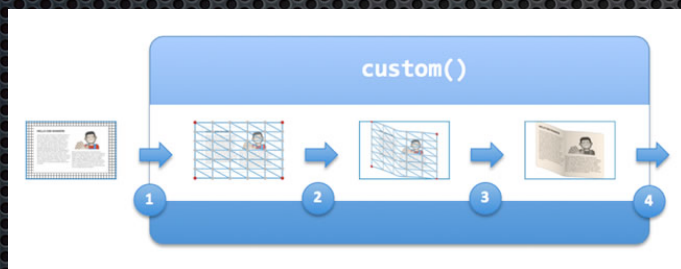
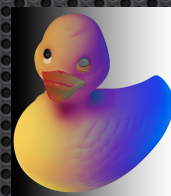
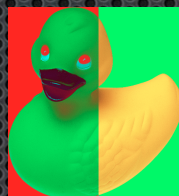
A quick brown fox jumps over the lazy dog. A quick brown the lazy dog. A quick brown fox jumps over the lazy dog. fox jumps over the lazy dog. fox jumps over the lazy dog.

Heading

A quick brown fox jumps over the lazy dog. A quick brown fox jumps over the lazy dog. fox jumps over the lazy dog.



Filters and friends



Selectors 4

- `:matches()` or « even the examples in the spec don't show why we need it »
- `:not(foo > bar, tut + tat)`
- `a:has(> img)` that raises many parser and OM issues
- `tr:nth-child(even of :not([hidden]))`
`tr:nth-child(2n+1)` and `*:nth-child(2n+1 of tr)` mean the same AND with same specificity

Poorly-designed but useful to some

- 'will-change' property

I was thinking: if will-change applies to more than K elements in a document at one time (e.g. K=10), ignore it completely in that document.

- Shadow DOM styling

3.2	Shadow DOM Selectors
3.2.1	Selecting Into the Light: the <code>:host</code> , <code>:host()</code> , and <code>:host-context()</code> pseudo-classes
3.2.2	Selecting Into the Dark: the <code>::shadow</code> pseudo-element
3.2.3	Selecting Shadow-Projected Content: the <code>::content</code> pseudo-element
3.2.4	Selecting Through Shadows: the <code>'/deep/'</code> combinator

- Fast process with too few people leads to weak solutions



Towards world domination

(but be sure we'll bikeshed that too)

Extensibility

- ✦ This is a world of polyfills and that feels wrong...
- ✦ Polyfills are super-hacky because the CSS Parser is not reachable, the Selectors model is not reachable, the Layout engine is not reachable
- ✦ We must improve that situation

Parsing and OM

- It should be possible to instantiate a CSS parser
- That parser should be able to parse sheets or parts of sheets like selectors or just values
- We need an Object Model for Selectors, a better OM for Values
- We need an OM for complex notations like `:nth-child(3n+1 of li:important)` or `linear-gradient()` so you don't have to reparse it
- Create your own properties and attach them to a Layout extension
(Chris Wilson will buy you wine if you reuse `hasLayout()` as a new API)

Media Queries and Pseudo-classes

- ✦ Let the Web Author define its own MQs and pseudos
- ✦ Attach a user-defined MQ or Pseudo-class to a JavaScript returning a boolean
- ✦ Let's also have an OM for that

Access to the Box Tree

- ✦ All rendering engines have always hidden the Box Tree
- ✦ Necessary to create serious layout experiments
 - ✦ Define what is a Box; make the tree reachable
 - ✦ Read-only asynchronous APIs
 - ✦ Allow to create new boxes!

Participation



- One place **only**: www-style@w3.org
(see <https://lists.w3.org/Archives/Public/www-style/> for subscription)

Thank you!

Daniel Glazman

@glazou

Bulgaria Web Summit 2015

Sofia, Bulgaria

<http://disruptive-innovations.com/zoo/slides/20150418-BWS/BWS2015.key>

