Pushing CSS to new frontiers

Daniel Glazman 20151204



Unicorns and Dinosaurs

- first proposal October 1994
- CSS 1 REC December 1996
- CSS 2 REC May 1998
- first CSS 3 RECs summer 2011
- 60+ documents on CSS WG's radar today



	CSS	Rest of the OWP
1. APIs	minimal	API frenzy
2. Syntactic sugar	18 years to get CSS Variables*	drastic changes
3. Domain of impact	Style only	ubiquity
4. Extensibility	none	major

State of the Onion

*Went to CR today!



1. APIS





The CSS Object Model

- The current CSS Object model deeply sucks
- Weak
- Was never fully implemented
- Strong resistance from vendors to some of the details they agreed on (Views for instance)
- But the CSS OM remains the ground layer for all polyfills, all editors, and many Web apps



- No Selector OM
- Can't climb up the Cascade
- No access to containing block's source element
- CSSValues weakly implemented
- No control over serialization

Selector
div:hover > p.warning:first-line
the first line of
elements of type p not of class warning
children of elements of type not hovered by the pointer
+ – Cancel OK



- No Selector OM
- Can't climb up the Cascade
- No access to containing block's source element
- CSSValues weakly implemented
- No control over serialization

The largest CSS confere

December 4, 2015

REGISTER NOW

spector	<u>></u>	Console	🕕 Deb	bugger	{ } Style Edit	or 🕅 Shader Editor
ı.btn.btn-p	orice	> Q			Rules	Computed
<pre>conference in Europe <h3></h3></pre>		e	Q Filter Styles			
			<pre>element { }</pre>			
December 	er 4,	2015 ·	PARIS	.page-	<pre>-index .btn- osition: abs</pre>	price 🖨 { solute;
<pre><a <="" brof="/tickets" class="btn btn-price" pre="" target="_blank"></pre>		rice"	height: 72px; line-height: 68px;			
role="l	butto	n">		C	olor: • #31	2C24 !important;
/div>				t f	ext-transfor	rm: uppercase;
11V> V>				f	ont-size: 24	4px;
class=	sect	ion sect	ion-	▶ b ▶ p b	order-radius adding: 0 50 ottom: -35p>	s: 72px;)px; <;



- No Selector OM
- Can't climb up the Cascade
- No access to containing block's source element
- CSSValues weakly implemented
- No control over serialization





- No Selector OM
- Can't climb up the Cascade
- No access to containing block's source element
- CSSValues weakly implemented
- No control over serialization

REGISTER NOW					
er	<pre>{ } Style Editor</pre>	Shader Editor			
	Rules	Computed			
ξFi	lter Styles				
<pre>text-align: center; vertical-align: middle; cursor: pointer; background-image: none; border: 1px solid transparent; white-space: nowrap; padding: 6px 12px; font-size: 14px; line-height: 1.42857; border-radius: 4px; vebkit ween coloct. percent.</pre>					

dot CSS

- No Selector OM
- Can't climb up the Cascade
- No access to containing block's source element
- CSSValues weakly implemented
- No control over serialization

tion »	
ts	serializing CSS
	To serialize any stylesheet use:
	<pre>print sheet.cssText</pre>
3	Also most other objects have a similar prosent of selectorText) but all use the global serialized
	<pre>>>> sheet = cssutils.parseString('a, b</pre>
	<pre>>>> print sheet.cssRules[0].cssText a, b { color: green }</pre>
Go	<pre>>>> print sheet.cssRules[0].selectorTex a, b</pre>
iodule,	>>> print sheet.cssRules[0].selectorLis b
	Preferences
	Quite a few preferences of the cssutils seria

dot CSS

2. Syntactic Sugar

SYNTACTIC SUGAR CAUSES CANCER OF THE SEMICOLON.

QUOTEHD.COM

Alan Perlis



Preprocessors

Most professional CSS authors use a preproc

- CSS preprocs fill gaps in CSS
- But « CSS is not a programming language »...



You've waited too long

- Rule nesting
- Mixins
- Inheritance
- Iterators
- Real #define constants?
- But the corresponding Object Model?



3. Domain of impact





Why limit ourselves to Style?

- *pointer* to an arbitrary node in a tree
- CSS syntax is still *notepadable*

• A selector is a great way to represent a fragment

Many codes currently rely on XPath to keep a



Example: transformations

- XSLT is very powerful... AND very complex...
- could easily challenge it
- **Example**: automatic generation of a Table of Contents

SASS-like syntax + "transformation" properties



```
/* Delete all children of existing TOC if it exists */
ol.toc {
 delete-children: *;
}
/* Make sure each header has an ID */
h1,h2,h3:not([id]) {
 modify: [id="toc" random()];
}
section > (h1) {
 ol.toc { /* create the list item for that H1 */
    append: li > a[href="#" attr(id, &&1)][[content(&&1)]] + ol;
(h1) + section > (h2),
(h2) + section > (h3),
(h3) + section > (h4),
(h4) + section > (h5),
(h5) + section > (h6) {
    append: li > a[href="#" attr(id, &&2)][[content(&&2)]] + ol;
}
/* Clean up empty nested lists */
ol.toc li {
  delete-children: ol:empty;
}
```

li:has(a[href="#" attr(id, &1)]) > ol { /* create the list item for that Hn */



4. Extensibility





CSS is a black box

- extensible
- unchanged since... ahem... 1998
- Highly time to shake that landcape

• CSS is the only bit of the whole OWP that is not

CSS is the only bit of the OWP with APIs almost



Custom Property/Values

- syntax, inheritance and initial value
- Attempts to also have user-defined pseudoelements and more
- All from JS, not sure this is the right model

"Register" new properties/values based on name,



- Allowing new properties/values is not enough
- tokens
- What else?

Parsing

Polyfills could have access to the CSS Lexer and

• Parse a string based on a CSS-like grammar?



• Do better than CSSValue in DOM 2 Style • Have it implemented, this time...

CSS Values



Box Tree, Painting, Scroll

- text and other information about boxes and fragments
- changes
- API to define specific scroll mechanisms

• API that gives developers access to geometry,

• API which allows developers to paint a part of an element in response to geometry / computed style



Conclusion

 It's also about making the above match production environments' requirements and needs

• CSS is not only about Styling a document tree



http://disruptive-innovations.com/zoo/slides/20151204-dotCSS a better CSSValue proposal: http://is.gd/TZbyaq

@glazou

Thanks!



