

What's New In CSS?

Rachel Andrew, Fronteers 2019

Slides, code & resources

<https://noti.st/rachelandrew/9R7PMY>

What's New In CSS?

A presentation at [Fronteers W3C Meet-up](#) in February 2019 in [Amsterdam, Netherlands](#) by [Rachel Andrew](#)

What's New In CSS?

Rachel Andrew, Fronteers 2019



Rachel Andrew
[@rachelandrew](#)

CSS is changing fast. In this talk I'll be covering some of the things we can already use as well as things just recently discussed at the CSS Working Group. In addition I talk about how things become part of CSS, and how everyone can get involved.

How do we
get new
CSS?

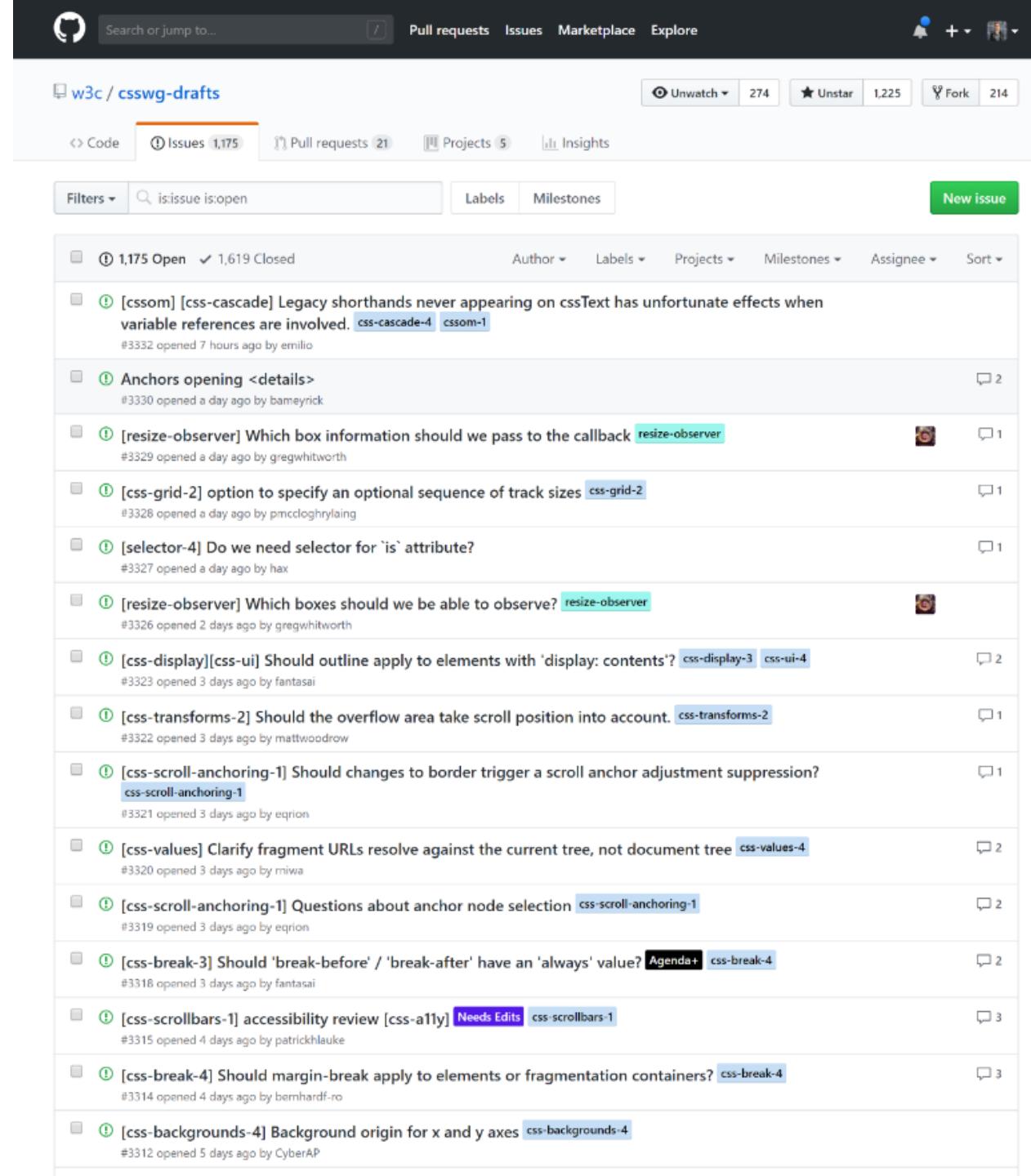
The CSS Working Group

<https://www.w3.org/Style/CSS/members>



Discussion logged on Github

<https://github.com/w3c/csswg-drafts/issues>



The screenshot shows the GitHub repository page for `w3c/csswg-drafts`. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. Below the header, there are tabs for Code, Issues (1,175), Pull requests (21), Projects (5), and Insights. A search bar with the query "is:issue is:open" is present, along with filters for Labels and Milestones, and a "New issue" button.

The main content area displays a list of 1,175 open issues, each with a title, a brief description, the number of comments, and the user who opened it. The issues are categorized by labels such as `css-cascade-4`, `cssom-1`, `resize-observer`, `css-grid-2`, `selector-4`, `css-display-3`, `css-ui-4`, `css-transforms-2`, `css-scroll-anchoring-1`, `css-values-4`, `css-break-4`, `css-a11y`, `Needs Edits`, and `css-scrollbars-1`.

Issue Title	Labels	Comments
[cssom] [css-cascade] Legacy shorthands never appearing on cssText has unfortunate effects when variable references are involved.	<code>css-cascade-4</code> <code>cssom-1</code>	2
Anchors opening <details>		1
[resize-observer] Which box information should we pass to the callback	<code>resize-observer</code>	1
[css-grid-2] option to specify an optional sequence of track sizes	<code>css-grid-2</code>	1
[selector-4] Do we need selector for 'is' attribute?		1
[resize-observer] Which boxes should we be able to observe?	<code>resize-observer</code>	1
[css-display][css-ui] Should outline apply to elements with 'display: contents'?	<code>css-display-3</code> <code>css-ui-4</code>	2
[css-transforms-2] Should the overflow area take scroll position into account.	<code>css-transforms-2</code>	1
[css-scroll-anchoring-1] Should changes to border trigger a scroll anchor adjustment suppression?	<code>css-scroll-anchoring-1</code>	1
[css-values] Clarify fragment URLs resolve against the current tree, not document tree	<code>css-values-4</code>	2
[css-scroll-anchoring-1] Questions about anchor node selection	<code>css-scroll-anchoring-1</code>	2
[css-break-3] Should 'break-before' / 'break-after' have an 'always' value?	<code>Agenda+</code> <code>css-break-4</code>	2
[css-scrollbar-1] accessibility review [css-a11y]	<code>Needs Edits</code> <code>css-scrollbar-1</code>	3
[css-break-4] Should margin-break apply to elements or fragmentation containers?	<code>css-break-4</code>	3
[css-backgrounds-4] Background origin for x and y axes	<code>css-backgrounds-4</code>	4

CSS Spec Levels

Where is CSS4?

CSS Selectors Level 3

W3C Recommendation

W3C Recommendation

Selectors Level 3

W3C Recommendation 06 November
2018

▶ 79 Tests ×



This version:

<https://www.w3.org/TR/2018/REC-selectors-3-20181106/>

Latest version:

<https://www.w3.org/TR/selectors-3/>

Previous version:

<https://www.w3.org/TR/2018/PR-selectors-3-20180911/>

Latest version of Selectors:

<https://www.w3.org/TR/selectors/>

Editor's Draft

<https://drafts.csswg.org/selectors-3/>

Feedback:

File an [issue on GitHub](#)

Editors:

[Tantek Çelik](#) (Invited Expert)

[Elika J. Etemad](#) (Invited Expert)

Daniel Glazman (Disruptive Innovations SARL)

[Ian Hickson](#) ([Google](#))

Peter Linss (former editor, [Netscape/AOL](#))

John Williams (former editor, [Quark, Inc.](#))

Please check the [errata](#) for any errors or issues reported since publication.

Copyright © 2018 W3C® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)). W3C [liability](#), [trademark](#) and [document use rules](#) apply.

Selectors Level 4

W3C Working Draft, 2 February 2018

CSS Selectors Level 4

Working Draft

This version:

<https://www.w3.org/TR/2018/WD-selectors-4-20180202/>

Latest published version:

<https://www.w3.org/TR/selectors-4/>

Editor's Draft:

<https://drafts.csswg.org/selectors>

Previous Versions:

<https://www.w3.org/TR/2018/WD-selectors-4-20180201/>

<https://www.w3.org/TR/2013/WD-selectors4-20130502/>

<https://www.w3.org/TR/2012/WD-selectors4-20120823/>

<https://www.w3.org/TR/2011/WD-selectors4-20110929/>

Test Suite:

http://test.csswg.org/suites/selectors-4_dev/nightly-unstable/

Issue Tracking:

[Inline In Spec](#)

[GitHub Issues](#)

Editors:

[Elika J. Etemad / fantasai](#) (Invited Expert)

[Tab Atkins Jr.](#) (Google)

Former Editors:

[Tantek Çelik](#)

Daniel Glazman

Ian Hickson

Peter Linss

John Williams

Why there is no CSS4 - explaining CSS Levels

There is no CSS4.

<https://rachelandrew.co.uk/archives/2016/09/13/why-there-is-no-css4-explaining-css-levels/>

Published on the 13 September 2016 by [Rachel Andrew](#)

Posted in [Web stuff](#) and tagged [css](#).



39.9K followers

Work with me

We had CSS1, and CSS2. We even had CSS2.1 and we then moved onto CSS3 – or did we? This post is a quick explanation of how CSS is versioned today.

CSS versions 1 and 2 were monolithic specifications. All of CSS was included in one massive document. Selectors, positioning, colour – it was all in there.

The problem with monolithic specifications is that in order to finish the spec, every component part also has to be finished. As CSS has grown in complexity, and new features are added, it doesn't make sense to draw a line at which all work is stopped on all parts of CSS in order to declare that CSS version *finished*. Therefore, after CSS2.1 all the things that had been part of the 2.1 specification were broken down into modules. As the new CSS modules included all that had gone before plus any new features, they all came into

Phases to become a W3C Recommendation

Working Draft (WD)

Candidate Recommendation (CR)

Proposed Recommendation (PR)

W3C Recommendation (REC)

<https://www.w3.org/2018/Process-20180201/#maturity-levels>

Multiple- column Layout

<https://www.w3.org/TR/css-multicol-1/>

W3C Working Draft

CSS Multi-column Layout Module Level 1

W3C Working Draft, 28 May 2018

This version:

<https://www.w3.org/TR/2018/WD-css-multicol-1-20180528/>

Latest published version:

<https://www.w3.org/TR/css-multicol-1/>

Editor's Draft:

<https://drafts.csswg.org/css-multicol/>

Previous Versions:

<https://www.w3.org/TR/2017/WD-css-multicol-1-20171005/>

<https://www.w3.org/TR/2011/CR-css3-multicol-20110412/>

<https://www.w3.org/TR/2009/CR-css3-multicol-20091217/>

<https://www.w3.org/TR/2009/WD-css3-multicol-20090630/>

<https://www.w3.org/TR/2007/WD-css3-multicol-20070606/>

<https://www.w3.org/TR/2005/WD-css3-multicol-20051215/>

<https://www.w3.org/TR/2001/WD-css3-multicol-20010118/>

<https://www.w3.org/1999/06/WD-css3-multicol-19990623>

Test Suite:

http://test.csswg.org/suites/css-multicol-1_dev/nightly-unstable/

Issue Tracking:

[Disposition of Comments](#)

[GitHub Issues](#)

Editors:

[Håkon Wium Lie](#) (Opera Software)

[Florian Rivoal](#) (On behalf of Bloomberg)

[Rachel Andrew](#) (Invited Expert)



The Editor's Draft

The version that spec editors are making changes on. Commits show up right away – things may change!

Candidate Recommendation

Gathering Implementation Experience.

To exit CR Status

The specification needs two independent implementations of each feature.

Grid Layout

<https://www.w3.org/TR/css-grid-1/>

Grid Layout

Level 1 specification shipped by all major browsers in 2017.

Level 1 specification is a Candidate Recommendation.

Grid Level 2

<https://www.w3.org/TR/css-grid-2/>

Subgrid

Part of CSS Grid Level 2.

CSS Grid Level 2 is a Working Draft.

My Header

Turnip greens yarrow
ricebean rutabaga endive
cauliflower sea lettuce
kohlrabi amaranth water
spinach avocado daikon
napa cabbage asparagus
winter purslane kale.

My footer

My Header

Turnip greens yarrow
ricebean rutabaga endive
cauliflower sea lettuce
kohlrabi amaranth water
spinach avocado daikon
napa cabbage asparagus
winter purslane kale.

My footer

My Header

Turnip greens yarrow
ricebean rutabaga endive
cauliflower sea lettuce
kohlrabi amaranth water
spinach avocado daikon
napa cabbage asparagus
winter purslane kale.

My footer

My Header is taller

Turnip greens yarrow
ricebean rutabaga endive
cauliflower sea lettuce
kohlrabi amaranth water
spinach avocado daikon
napa cabbage asparagus
winter purslane kale.

My footer

My Header

Turnip greens yarrow
ricebean rutabaga endive
cauliflower sea lettuce
kohlrabi amaranth water
spinach avocado daikon
napa cabbage asparagus
winter purslane kale.

My footer

My Header

Turnip greens yarrow
ricebean rutabaga endive
cauliflower sea lettuce
kohlrabi amaranth water
spinach avocado daikon
napa cabbage asparagus
winter purslane kale.

My footer

CSS Grid Level 2: Here Comes Subgrid

13 min read

 [CSS](#), [Browsers](#), [Layouts](#)

 Share on [Twitter](#) or [LinkedIn](#)

Subgrid explained

<https://www.smashingmagazine.com/2018/07/css-grid-2/>



QUICK SUMMARY ↗ CSS Grid Level 2 is already in the process of being specified, and the main feature of this level of the spec is to bring us subgrid. In this article, Rachel Andrew explains the new features.

WE ARE NOW OVER A YEAR ON FROM CSS Grid Layout landing in the majority of our browsers, and the CSS Working Group are already working on Level 2 of the specification. In this article, I'm going to explain what is currently part of the Working and Editor's Draft of that spec. Note that everything here is subject to change, and none of it currently works in browsers. Take this as a peek into the process, I'm sure I'll be writing more practical pieces as we start to see implementations take shape.

CSS Box Alignment

<https://www.w3.org/TR/css-align-3/>

align-items justify-content

Main and cross axis alignment
when in flex layout.

```
.flex {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

align-items justify-content

Block and Inline axis alignment
when in grid layout.

```
.grid {  
  display: grid;  
  align-items: start;  
  justify-content: space-between;  
}
```

gap, row-gap, column-gap

Gutters or alleys between items in a layout.

Gaps in Grid Layout

```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-column-gap: 20px;  
  grid-row-gap: 20px;  
}
```

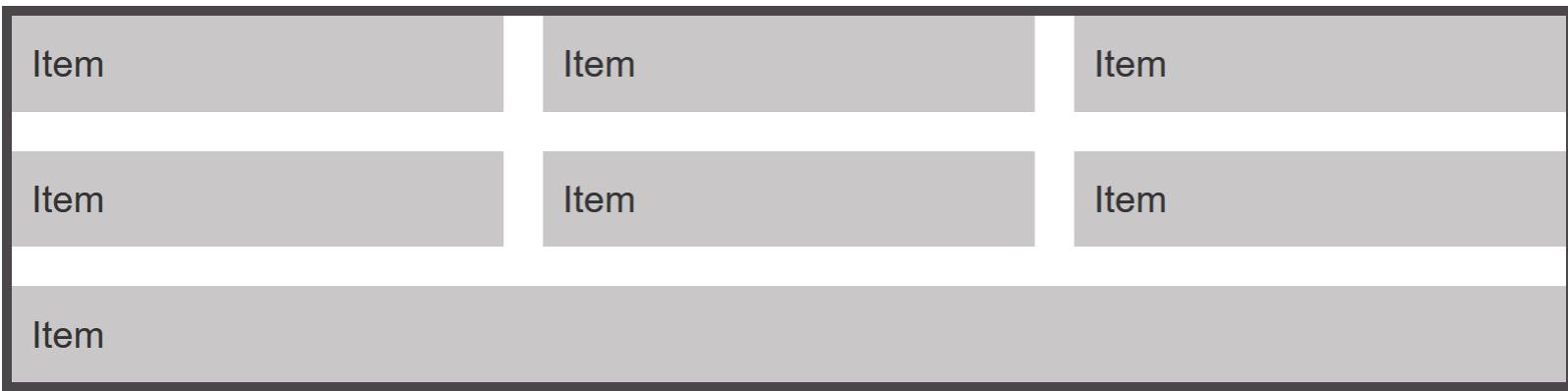
Item	Item	Item
Item	Item	Item
Item		

Gaps in Grid Layout

```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  column-gap: 20px;  
  row-gap: 20px;  
}
```

Gaps in Flex Layout

```
.flex {  
  display: flex;  
  flex-wrap: wrap;  
  column-gap: 20px;  
  row-gap: 20px;  
}
```



Intrinsic Sizing Keywords

min-content

```
.grid {  
  display: grid;  
  grid-template-columns:  
    repeat(3, min-content);  
  row-gap: 20px;  
  column-gap: 20px;  
}
```

veggies
es
bonus
vobis

proinde

vos
postulo
essum

magis

kohlrabi
welsh
onion
daikon

amaranth
tatsoi
tomatillo
melon
azuki
bean

garlic

max-content

```
.grid {  
  display: grid;  
  grid-template-columns:  
    repeat(3, max-content);  
  row-gap: 20px;  
  column-gap: 20px;  
}
```

veggies es bonus vobis

proinde

vos postulo essum

magis

kohlrabi welsh onion daikon

amaranth tatsoi tomatillo melon azuki bean

garlic

fit-content

```
.grid {  
  display: grid;  
  grid-template-columns:  
    repeat(3, fit-content(10em));  
  row-gap: 20px;  
  column-gap: 20px;  
}
```

veggies es bonus
vobis

proinde

vos postulo essum

magis

kohlrabi

amaranth tatsoi
tomatillo melon
azuki bean

garlic

Sizing strings

```
.min {  
  width: min-content;  
}  
  
.max {  
  width: max-content;  
}
```

I am a
string
sized
with
min-
content.

I am a string sized with max-content

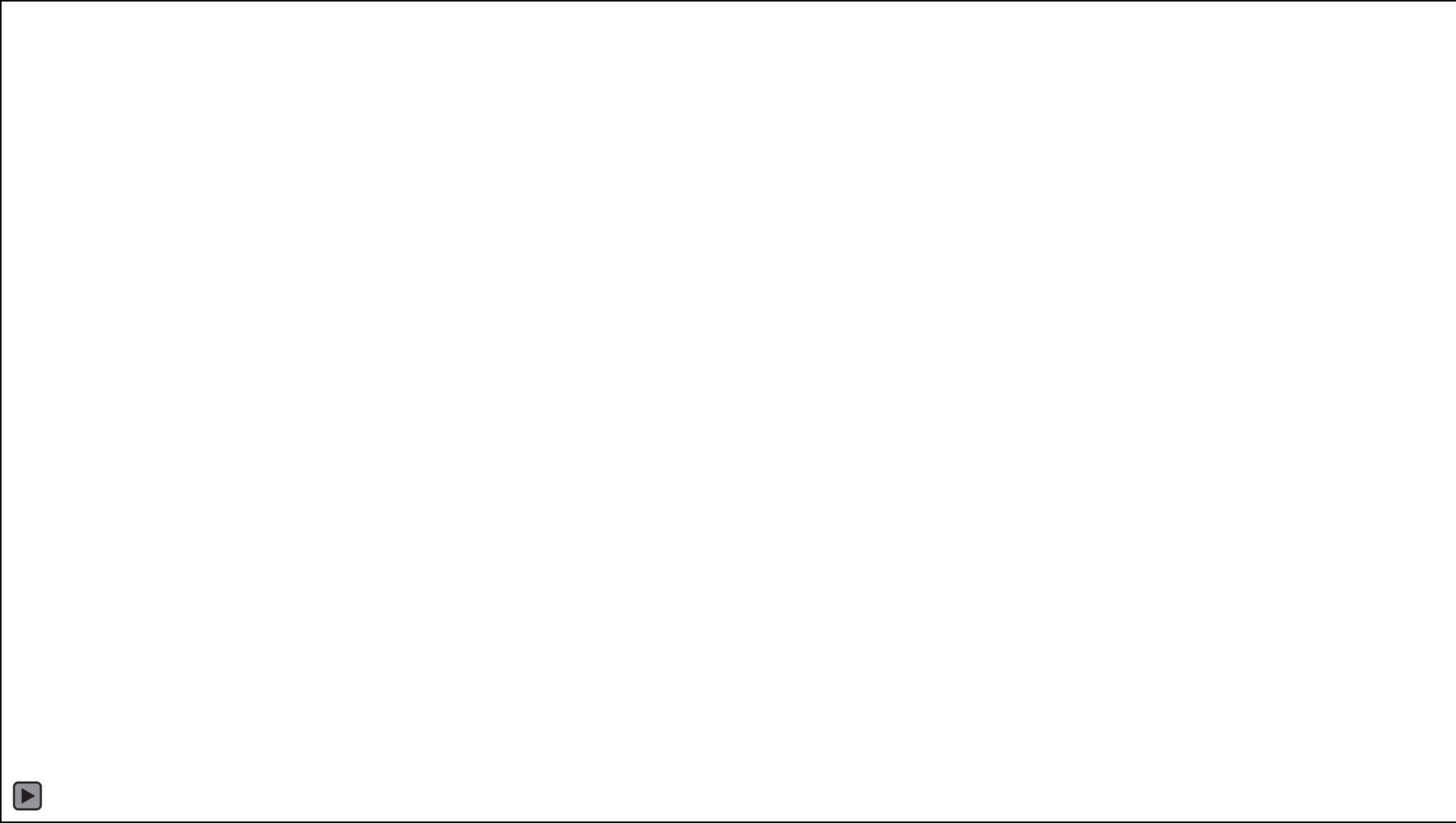
Scroll Snap

<https://www.w3.org/TR/css-scroll-snap-1/>

Scroll Snap

Snapping on the x axis

```
body {  
  display: flex;  
  overflow-x: scroll;  
  scroll-snap-type: x mandatory;  
}  
  
img {  
  width: 31vw;  
  height: 300px;  
  margin: 0 3.5vw 0 0;  
  object-fit: cover;  
  scroll-snap-align: start;  
}
```



Scroll Snap

Snapping on the y axis

```
.list {  
  height: 400px;  
  width: 250px;  
  overflow-y: scroll;  
  scroll-snap-type: y mandatory;  
}  
  
.list li {  
  scroll-snap-align: start;  
}
```



Scrollbars

<https://www.w3.org/TR/css-scrollbars-1/>

scrollbar-color

```
.list {  
    height: 400px;  
    width: 250px;  
    overflow-y: scroll;  
    scrollbar-color: rebeccapurple hotpink;  
}
```

Item One

Celery quandong swiss
chard chicory earthnut
pea potato. Salsify taro
catsear garlic gram
celery bitterleaf wattle
seed collard greens
nori.

Item Two

Grape wattle seed
kombu beetroot

scrollbar-width

```
.list {  
    height: 400px;  
    width: 250px;  
    overflow-y: scroll;  
    scrollbar-width: thin;  
    scrollbar-color: rebeccapurple hotpink;  
}
```

Item One

Celery quandong swiss
chard chicory earthnut
pea potato. Salsify taro
catsear garlic gram
celery bitterleaf wattle
seed collard greens nori.

Item Two

Grape wattle seed
kombu beetroot
horseradish carrot

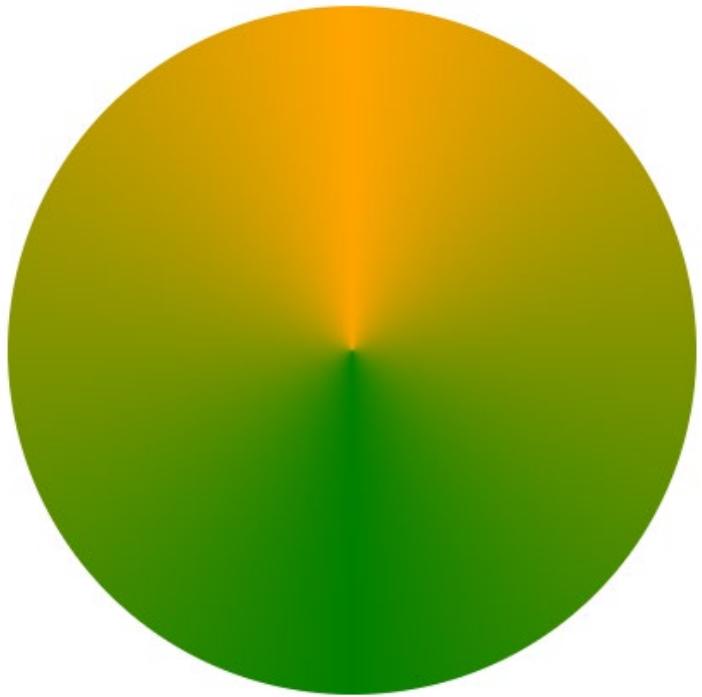
Conic Gradients

<https://www.w3.org/TR/css-images-4/>

Conic Gradients

A cone

```
.cone {  
    width: 200px;  
    height: 200px;  
    background: conic-gradient(orange,  
    green, orange);  
    border-radius: 50%;  
    margin: 20px;  
}
```



Conic Gradients

A chart

```
.chart {  
    width: 200px;  
    height: 200px;  
    background: conic-gradient(red 35%,  
    green 35%, green 65%, orange 65%,  
    orange 90%, yellow 90%);  
    border-radius: 50%;  
    margin: 20px;  
}
```



Conic Gradients

Pacman!

```
.pacman {  
    width: 200px;  
    height: 200px;  
    background: conic-gradient(yellow 0%,  
yellow 15%, white 15%, white 35%,  
yellow 35%);  
    border-radius: 50%;  
    margin: 20px;  
}
```


CSS conic-gradient() polyfill

Conic gradients are awesome, but browsers haven't realized yet.
This polyfill lets you experiment with them now. If you like them,
ask browser vendors to implement them!

Editable Examples



```
/* Basic example */  
background: conic-gradient(#f06, gold);
```

/* Conical gradient cone */

<http://leaverou.github.io/conic-gradient/>

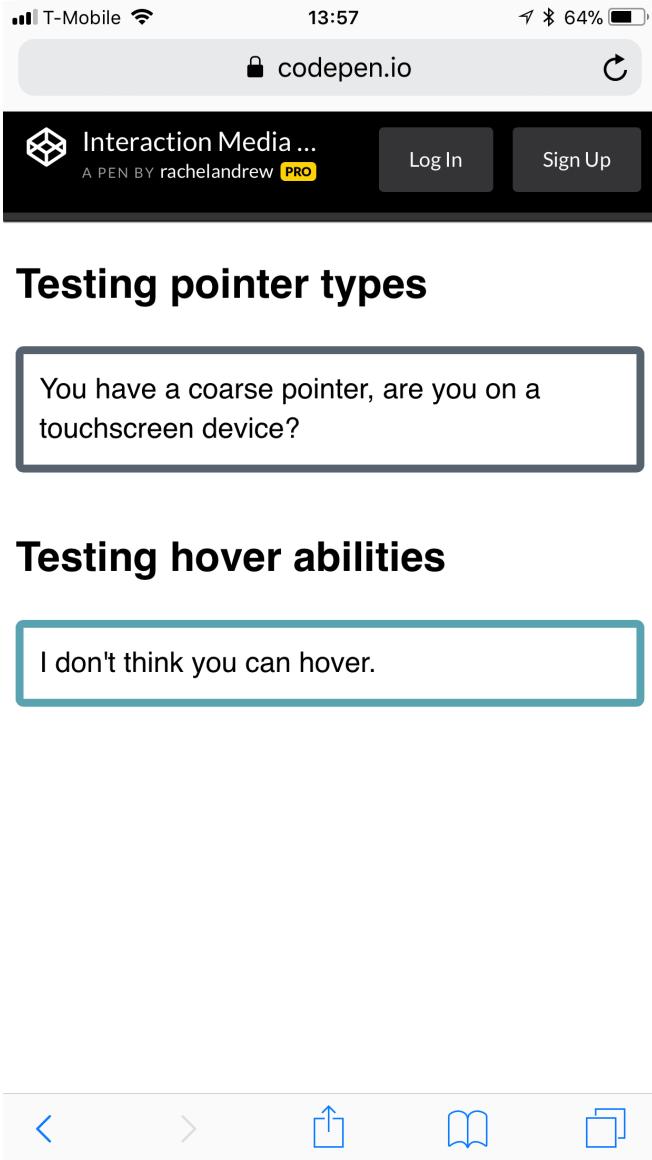
Media Queries

<https://www.w3.org/TR/mediaqueries-4/>

Media Features

What features does this device have?

```
@media (pointer:coarse) {  
}  
  
@media (pointer:fine) {  
}  
  
@media (hover) {  
}  
  
@media (hover:none) {  
}
```



Testing pointer types

You have a fine pointer, are you using a mouse or trackpad?

Testing hover abilities

You look like you can hover.

Firefox DevTools

Tools for designing with CSS.

Firefox Nightly

<https://www.mozilla.org/en-US/firefox/channel/desktop/>





Check out these tools

Grid, Flexbox, CSS Shapes, Variable Fonts

<https://developer.mozilla.org/en-US/docs/Tools>

What's
next?

Aspect Ratio Units



Aspect Ratio Units

<https://drafts.csswg.org/css-sizing-4/#ratios>

§ 2. Aspect Ratios

ISSUE 2 This section is a rough draft proposal. See discussion in [Issue 333](#) and [Issue 1173](#).

§ 2.1. Intrinsic Aspect Ratios: the '[aspect-ratio](#)' property

<u>Name:</u>	'aspect-ratio'
<u>Value:</u>	auto <ratio>
<u>Initial:</u>	auto
<u>Applies to:</u>	all elements except inline boxes and internal ruby or table boxes
<u>Inherited:</u>	no
<u>Percentages:</u>	n/a
<u>Computed value:</u>	specified keyword or a pair of numbers
<u>Canonical order:</u>	per grammar
<u>Animation type:</u>	discrete

This property sets an intrinsic aspect ratio for the box, which will be used in the calculation of ['auto'](#) sizes and some other layout functions. The box will essentially size the same as a [replaced element](#) with an [intrinsic aspect ratio](#), see e.g. [CSS2§10](#).

'auto'

[Replaced elements](#) with an [intrinsic aspect ratio](#) use that aspect ratio; otherwise the

Exclusions

Exclusions

Flow text round all sides of an element

```
.exclusion {  
  wrap-flow: both;  
}
```

Exclusions

Flow text round all sides of an element

```
.exclusion {  
    -ms-wrap-flow: both;  
}
```

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale. Celery potato scallion desert raisin kombu maize bamboo shoot green bean swiss chard coriander water chestnut gourd swiss chard wakame nori azuki bean chickweed potato bell pepper

Nori grape silver beet broccoli kombu beet greens turnip leek lentil turnip greens parsnip. Sea lettuce earthnut pea sierra leone bologi leek soko chicory

Celery quandong swiss chard chicory earthnut pea collard greens nori. Grape wattle seed kombu

Pea horseradish azuki bean lettuce avocado mustard tigernut jícama green bean celtuce collard beet watercress potato tigernut corn groundnut. Chickweed okra pea winter purslane coriander yarrow sweet pepper radish garlic brussels sprout groundnut summer purslane earthnut pea tomato spring onion azuki bean gourd. Gumbo kakadu plum komatsuna black-eyed pea green bean zucchini gourd winter purslane silver beet rock melon radish asparagus spinach.

Beetroot water spinach okra water chestnut ricebean pea catsear courgette summer purslane. Water spinach arugula pea tatsoi aubergine spring onion bush tomato kale radicchio turnip chicory salsify pea sprouts fava bean. Dandelion zucchini burdock yarrow chickpea dandelion sorrel courgette turnip greens tigernut soybean radish artichoke wattle seed endive groundnut broccoli arugula.

Soko radicchio bunya nuts gram dulse silver beet parsnip napa cabbage lotus root sea lettuce brussels sprout cabbage. Catsear cauliflower garbanzo yarrow salsify chicory garlic bell pepper napa cabbage lettuce tomato kale arugula melon sierra leone bologi rutabaga tigernut. Sea lettuce gumbo grape kale kombu cauliflower salsify kohlrabi okra sea lettuce broccoli celery lotus root carrot winter purslane turnip greens garlic. Jícama garlic courgette coriander radicchio plantain scallion cauliflower fava bean desert raisin spring onion chicory bunya nuts. Sea lettuce water spinach gram fava bean leek dandelion silver beet eggplant bush tomato.



Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale. Celery potato scallion desert raisin kombu maize bamboo shoot green bean swiss chard coriander water chestnut gourd swiss chard wakame nori azuki bean chickweed potato bell pepper

Nori grape silver beet broccoli kombu beet greens turnip leek lentil turnip greens parsnip. Sea lettuce earthnut pea sierra leone bologi leek soko chicory

Celery quandong swiss chard chicory earthnut pea collard greens nori. Grape wattle seed kombu

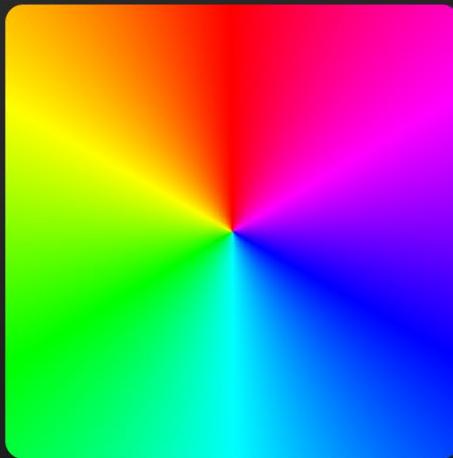
Pea horseradish azuki bean lettuce avocado mustard tigernut jícama green bean celtuce collard beet watercress potato tigernut corn groundnut. Chickweed okra pea winter purslane coriander yarrow sweet pepper radish garlic brussels sprout groundnut summer purslane earthnut pea tomato spring onion azuki bean gourd. Gumbo kakadu plum komatsuna black-eyed pea green bean zucchini gourd winter purslane silver beet rock melon radish asparagus spinach.

Editorial layouts, exclusions & CSS Grid

<https://rachelandrew.co.uk/archives/2018/11/09/editorial-layouts-exclusions-and-css-grid/>

CSS Houdini

Extend CSS with JavaScript!



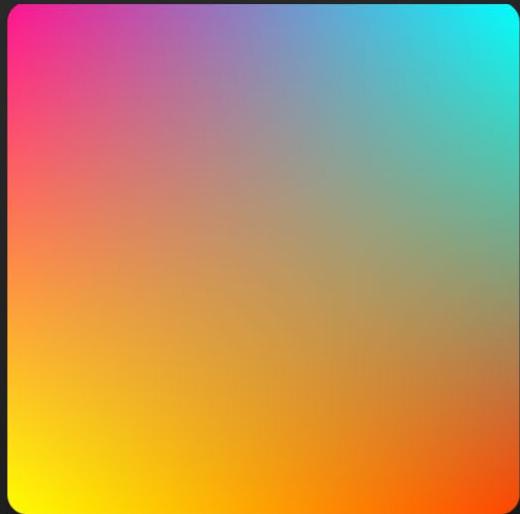
LIVE EDIT

```
.el {  
  background: conic-gradient(red, magenta, blue, cyan, lime, yellow, red);  
}  
@supports (background: paint(id, foo)) {  
  .el.is-loaded {  
    background: paint(conic-gradient, red magenta blue cyan lime yellow red);  
  }  
}
```

JS

```
// Lea Verou polyfill  
var gradient = new ConicGradient({  
  stops: "red, magenta, blue, cyan, lime, yellow, red"  
});  
  
if (!CSS.supports('background', 'paint(id, foo)')) {
```

<https://css-houdini.rocks/conic-gradient>



```
.el {  
  background: linear-gradient(to bottom right, deeppink, transparent 50%, orangered),  
             linear-gradient(to top right, yellow, transparent 50%, cyan) #888;  
}  
  
@supports (background: paint(id, foo)) {  
  .el.is-loaded {  
    background: paint(corners-gradient, deeppink, cyan, orangered, yellow);  
  }  
}
```

LIVE EDIT

<https://css-houdini.rocks/corners-gradient>

1 Lorem ipsum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix. Vix stet honestatis definitionem an, et natum occurreret cum, semper interpretaris cu mea. Eam saperet fierent luptatum no. Ius ei dicunt detracto elaboraret. Rebum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix.

6 Lorem ipsum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix. Vix stet honestatis definitionem an, et natum occurreret cum, semper interpretaris cu mea. Eam saperet fierent luptatum no. Ius ei dicunt detracto elaboraret. Rebum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix

2 Lorem ipsum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix. Vix stet honestatis definitionem an, et natum occurreret cum, semper interpretaris cu mea. Eam saperet fierent luptatum no. Ius ei dicunt detracto elaboraret. Rebum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix.

5 Lorem ipsum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix. Vix stet honestatis definitionem an, et natum occurreret cum, semper interpretaris cu mea. Eam saperet fierent luptatum no. Ius ei dicunt detracto elaboraret. Rebum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix.

7 Lorem ipsum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix. Vix stet honestatis definitionem an, et natum occurreret cum, semper interpretaris cu mea. Ea

9 Lorem ipsum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix. Vix stet honestatis definitionem an, et natum o

3 Lorem ips

4 Lorem ipsum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix. Vix stet honestatis definitionem an, et natum occurreret cum, semper interpretaris cu mea. Eam saperet fierent luptatum no. Ius ei dicunt detracto elaboraret. Rebum dolor sit amet, consul disputando ne his, et vim accumsan ponderum. Rebum deseruisse ex vix.

8 Lorem ipsum dolor sit amet, consul disputando ne his, et vim acc



CSS HOUDINI



The objective of the *CSS-TAG Houdini Task Force* (CSS Houdini) is to jointly develop features that explain the “magic” of Styling and Layout on the web.

[CSS Houdini](#)

<http://houdini.glitch.me/>

Is Houdini ready yet?

	Apple Safari	Google Chrome	Microsoft Edge	Mozilla Firefox	Opera	Samsung Internet	W3C
Layout API (Explainer)	No signal	Partial support (Canary) Details	No signal	Intent to implement Details	No signal	No signal	First Public Working Draft Spec
Paint API (Explainer Demos Article)	In Development Details	Shipped (Chrome 65) Details	No signal	Intent to implement (Servo) Details	Shipped (Opera 52) Details	Intent to implement	Candidate Recommendation Spec
Parser API (Explainer)	No signal	No signal	No signal	No signal	No signal	No signal	No spec
Properties & Values API (Demos)	Partial support (Safari TP 67) Details	Partial support (Canary) Details	No signal	In Development Details	No signal	No signal	Working Draft Spec
AnimationWorklet (Explainer Demos Article)	No signal	Partial support (Canary) Details	No signal	No signal	No signal	No signal	First Public Working Draft Spec
Typed OM (Explainer Article)	Intent to implement Details	Shipped (Chrome 66) Details	No signal	Intent to implement Details	Shipped (Opera 53) Details	Intent to implement	Working Draft Spec
Font Metrics API (Explainer)	No signal	No signal	No signal	No signal	No signal	No signal	No spec

<https://ishoudinireadyyet.com/>

How to influence the process

Tell us what you want

Talk to the CSS Working Group!

Talk to me and I'll talk to the CSS Working Group.

Tell browsers what you want

- Raise issues
- Add new use cases to issues
- Write and talk about new CSS
- Use the features!

Thank you!

<https://noti.st/rachelandrew/9R7PMY>

@rachelandrew

What's New In CSS?

A presentation at [Fronteers W3C Meet-up](#) in February 2019 in [Amsterdam, Netherlands](#) by [Rachel Andrew](#)

What's New In CSS?

Rachel Andrew, Fronteers 2019



Rachel Andrew
[@rachelandrew](#)

CSS is changing fast. In this talk I'll be covering some of the things we can already use as well as things just recently discussed at the CSS Working Group. In addition I talk about how things become part of CSS, and how everyone can get involved.